EDM 2014

7th International Conference on Educational Data Mining

PROCEEDINGS OF THE
SEVENTH INTERNATIONAL CONFERENCE ON
EDUCATIONAL DATA MINING

London, United Kingdom, July 4-7, 2014

**John Stamper, Zachary Pardos,
Manolis Mavrikis, Bruce M. McLaren**

7th International Conference on Educational Data Mining (EDM 2014)
Proceedings of the Seventh International Conference on Educational Data Mining
John Stamper, Zachary Pardos, Manolis Mavrikis, and Bruce M. McLaren (Eds.)
London, July 4-7, 2014

ISBN: 978-0-9839525-4-1

PREFACE

The 7th International Conference on Education Data Mining held on July 4th -7th, 2014 at the Institute of Education, London, UK is the leading international forum for high-quality research that mines large data sets in order to answer educational research questions that shed light on the learning process. These data sets may come from the traces that students leave when they interact, either individually or collaboratively, with learning management systems, interactive learning environments, intelligent tutoring systems, educational games or when they participate in a data-rich learning context. The types of data therefore range from raw log files to eye-tracking devices and other sensor data. Being hosted in London, UK the theme of the conference is "Big Data - Big Ben - Education Data Mining for Big Impact in Teaching and Learning".

In our 7th consecutive year of EDM conferences, it is clear that the field is continuing to grow at a rapid pace. With renewed focus on education driven by big data learning analytics has put the EDM field in the center of growing interest. Traditional educational technologies, intelligent tutoring systems, educational games, and learning management systems all continue to generate growing amounts of data that are becoming available for analysis. The new interest in MOOCs and their promise to reach thousands or even hundreds of thousands of students per class requires techniques for feedback and grading that are being researched in the EDM domain. The conference submissions this year also continue to grow. This year we had 142 submissions as full or short papers representing a 30% increase over 2013. The program committee reviewed all submissions and based on the reviews the program chairs accepted 24 full papers and 34 short papers making the acceptance rate 17% (full) and 41% (full and short) respectively. Selecting the papers as such allows for full papers to be presented in a single track format.

Continuing the successful mini-tutorial sessions offered at EDM 2013, we also have a full workshop and tutorial program this year with 4 workshops and 4 tutorials held on the first day of the conference. One page abstract of each workshop is included in these proceedings. The full proceedings can be found on CEUR: http://ceur-ws.org/Vol-1183/

A tremendous amount of work has gone into bringing this conference together and we personally thank all of those who have contributed including the organizing committee, the program committee, additional reviewers, and the invited speakers. Also, we give a big thanks to our record number of sponsors for their generous support. They include Carnegie Learning, Inc. (Gold), MARi (Gold), Pearson (Gold), Kaplan (Silver), Whizz Education (Bronze), Intellimedia (Bronze), Realize It (Bronze), and Reasoning Mind (Bronze).

We hope you enjoy these proceedings and recognize the great research that has made EDM 2014 such a success!

John Stamper
Zachary Pardos
Manolis Mavrikis
Bruce M. McLaren

Organization

CONFERENCE CHAIRS

Manolis Mavrikis	London Knowledge Lab, Institute of Education, University of London
Bruce M. McLaren	Human-Computer Interaction Institute, Carnegie Mellon University

PROGRAM CHAIRS

John Stamper	Human-Computer Interaction Institute, Carnegie Mellon University
Zachary Pardos	School of Information, UC Berkeley

WORKSHOP AND TUTORIAL CHAIRS

Sergio Gutiérrez-Santos	London Knowledge Lab, Birkbeck, University of London
Olga C. Santos	aDeNu Research Group

POSTER CHAIRS

Mingyu Feng	SRI International
Patricia Charlton	London Knowledge Lab, Institute of Education, University of London

INDUSTRY TRACK CHAIRS

George Khachatryan	Reasoning Mind
Kaska Porayska-Pomsta	London Knowledge Lab, Institute of Education, University of London

YOUNG RESEARCHERS TRACK CHAIRS

Gautam Biswas	School of Engineering, Vanderbilt University
Martina Rau	School of Education, University of Wisconsin-Madison

WEB AND SOCIAL MEDIA

Beate Grawemeyer	London Knowledge Lab, Birkbeck, University of London
Ani Aghababyan	Utah State University

PROGRAM COMMITTEE

Omar Alzoubi	Carnegie Mellon University, Qatar
Mirjam Augstein	Upper Austria University of Applied Sciences
Tiffany Barnes	North Carolina State University
Gautam Biswas	Vanderbilt University
Mary Jean Blink	TutorGen, Inc.
Rafael Calvo	University of Sydney
Theodore Carmichael	TutorGen, Inc.
John Champaign	Massachusetts Institute of Technology
Min Chi	North Carolina State University
Christophe Choquet	University of Maine, France

Sidney D'Mello	University of Notre Dame
Michel Desmarais	Ecole Polytechnique de Montreal
Hendrik Drachsler	Open University of the Netherlands
Kristy Elizabeth Boyer	North Carolina State University
Mingyu Feng	SRI International
Davide Fossati	Carnegie Mellon University, Qatar
Armando Fox	UC Berkeley
Dragan Gasevic	Athabasca University
Eva Gibaja	University of Córdoba
Daniela Godoy	ISISTAN Research Institute
Ilya Goldin	Pearson
Joseph Grafsgaard	North Carolina State University
Neil Heffernan	Worcester Polytechnic Institute
Arnon Hershkovitz	Tel Aviv University
Jonathan Huang	Stanford University
Roland Hubscher	Bentley University
Sébastien Iksal	University of Maine, France
Jihie Kim	USC Information Science Institute
Evgeny Knutov	Technische Universiteit Eindhoven
Kenneth Koedinger	Carnegie Mellon University
Irena Koprinska	University of Sydney
Vanda Luengo	Université Joseph Fourier
Lina Markauskaite	The University of Sydney
Noboru Matsuda	Carnegie Mellon University
Manolis Mavrikis	London Knowledge Lab
Riccardo Mazza	University of Lugano / SUPSI
Gordon McCalla	University of Saskatchewan
Bruce Mclaren	Carnegie Mellon University
Agathe Merceron	Beuth University of Applied Sciences Berlin
Rob Miller	Massachusetts Institute of Technology
John Mitchell	Stanford University
Tanja Mitrovic	University of Canterbury, Christchurch
Jack Mostow	Carnegie Mellon University
Sergiy Nesterko	HarvardX
Alexandru Niculescu-Mizil	NEC Labs America / Princeton University
Tristan Nixon	Carnegie Learning Inc.
Roger Nkambou	University of Québec at Montréal
Andrew Olney	University of Memphis
Abelardo Pardo	The University of Sydney
Zachary Pardos	UC Berkeley
Mykola Pechenizkiy	Eindhoven University of Technology
Steve Ritter	Carnegie Learning, Inc.
Cristobal Romero	University of Córdoba
Carolyn Rose	Carnegie Mellon University
Ryan S.J.D. Baker	Columbia University Teachers College
Richard Scheines	Carnegie Mellon University

John Shawe-Taylor	University College London
John Stamper	Carnegie Mellon University
Jun-Ming Su	National Chiao Tung University
Sebastián Ventura	University of Córdoba
Stephan Weibelzahl	Private University of Applied Sciences Göttingen
Fridolin Wild	The Open University
Kalina Yacef	The University of Sydney
Michael Yudelson	Carnegie Learning, Inc.
Osmar Zaïane	University of Alberta

ADDITIONAL REVIEWERS

Seth Adjei
Behzad Beheshti
Shane Bergsma
Nigel Bosch
Chris Brooks
Philip Buffum
Carrie Cai
Veronica Catete
Kai-Min Kevin Chang
John A. Doucette
Mohammad Hassan Falakmasir
Philippe Fournier-Viger
Christopher Fox
April Galyardt
Elena Glassman
José González-Brenes
Beate Grawemeyer
Seiji Isotani
Srecko Joksimovic
Sokratis Karkalas
Tanja Käser
Juho Kim
John Kinnebrew
Vitomir Kovanovic
Sébastien Lallé
Nan Li
Ran Liu
Collin Lynch
Luca Mazzola
Thomas McTavish
Caitlin Mills
Behrooz Mostafavi
Andrea Nickel
Amy Ogan
Terry Peckham
Justis Peters
Irena Rindos
Rinat Rosenberg-Kima
Mohamed Rouane-Hacene
Oliver Scheuer
Johannes Schönböck
Douglas Selent
Mika Seppala
Eliane Stampfer
Craig Thompson
Yutao Wang
Yanbo Xu

GOLD SPONSORS

Carnegie Learning

MARi™

PEARSON

SILVER SPONSORS

BRONZE SPONSORS

Table of Contents

Invited Keynote Talks (Abstracts)

Full Papers

Short Papers

Posters

Young Researcher Papers

Workshops

Invited Keynote Talks (Abstracts)

The field of EDM: where we came from and where we're going

Joseph Beck

Abstract: The Educational Data Mining community has undergone tremendous growth in the past decade. This talk will discuss how we got to where we are, as well as upcoming challenges for the field. The beginning of the EDM workshop series grew out of the AIED and ITS conferences, which greatly influenced both the initial participants and the frameworks used for viewing data mining problems. The development of the EDM conference series served to focus the field, and greatly increase the range of participants. Although much progress has been made in the past 6 years, there remain some large challenges not (yet) well addressed by the EDM community. Two issues include who are the consumers for the advances that we make, and under what conditions can we draw scientific conclusions from data-mining activities.

Short biography: Joseph Beck, assistant professor of Computer Science, has been at WPI since 2007. His research focuses on educational data mining, a new discipline that develops techniques for analyzing large educational data sets to make discoveries that will improve teaching and learning. His work centers on estimating how computer tutors impact learning. He established the first workshop in the field and in 2008 was program co-chair of the first International Conference on Educational Data Mining. He holds a BS in mathematics, computer science, and cognitive science from Carnegie Mellon University, and a PhD in computer science from the University of Massachusetts, Amherst.

Generative Adaptivity for Optimization of the Learning Ecosystem

Zoran Popovic

Abstract: Most of the current work on improving learning outcomes focuses on a small subset of variables of an immensely multi-dimensional space of the learning ecosystem. With ITS, learning games, and other digital content we consider only individual students, other research focuses only on teacher development, or only on curriculum improvement. In this talk I will describe our efforts on how to discover optimal parameters of this system that considers student factors (engagement and mastery), classroom factors (blended learning variations and group learning variations), curriculum factors (multidimensional variation of existing curricula), and teacher factors (in-class tools that mitigate weaknesses, and promote teacher development). I will describe our work on algorithms to discover optimal learning pathways in this high-dimensional space. I will conclude with recent remarkable outcomes of deploying a portion of our platform on algebra challenges conducted on two US states and the country of Norway.

Short biography: Zoran Popovic is a Director of Center for Game Science at University of Washington and founder of Engaged Learning. Trained as a computer scientist his research focus is on creating interactive engaging environments for learning and scientific discovery. His laboratory created Foldit, a biochemistry game that produced three Nature publications in just two years, an award-winning math learning games played by over five million learners worldwide. He is currently focusing on engaging methods that can rapidly develop experts in arbitrary domains with particular focus on revolutionizing K-12 math education. His Algebra Challenges conducted in Washington, Minnesota, and Norway, have shown that more than 93% of children even in elementary school can learn key algebra concepts in 1.5 hours. He has recently founded Engaged Learning to apply his work on generative adaptation to any curricula towards the goal of achieving school mastery by 95% of students. His contributions to the field of interactive computer graphics have been recognized by a number of awards including the NSF CAREER Award, Alfred P. Sloan Fellowship and ACM SIGGRAPH Significant New Researcher Award.

150K+ online students at a time: How to understand what's happening in online learning

Daniel Russell

Abstract: Many MOOCs have had more that 100K students register for their courses, with many completing, but many dropping out. Is this the future of online education? Should we worry about attrition, or is this a new, natural, and expected trend in online learning? More importantly, how can we come to understand the (new) student experience? In the past year we have run several MOOCs with more than 350K registrants (and then another 250K who have taken the MOOC without the synchronous class structure). Learning in MOOCs is rather different than traditional learning experiences, and now we have the tools to start to understand how and why those differences exist. However, analytics often miss important behaviors that are key to understanding the inner life of the online student. I'll discuss the boundaries between EDM and observational methods that reveal the social community of learners that are essential for making MOOCs succeed, and what seems to work (and not work) in MOOCs.

Short biography: Daniel Russell is the Über Tech Lead for Search Quality and User Happiness in Mountain View. He earned his PhD in computer science, specializing in Artificial Intelligence until he realized that magnifying human intelligence was his real passion. Twenty years ago he foreswore AI in favor of HI, and enjoys teaching, learning, running and music, preferably all in one day. His MOOCs have helped students become much more effective online searchers. His online course, PowerSearchingWithGoogle.com has had ~500K students go through the content, meaning that somewhere on earth, a video of him teaching search skills has been on-screen for more than 200 years.

Full Papers

Adaptive Practice of Facts in Domains with Varied Prior Knowledge

Jan Papoušek
Masaryk University Brno
jan.papousek@mail.muni.cz

Radek Pelánek
Masaryk University Brno
pelanek@fi.muni.cz

Vít Stanislav
Masaryk University Brno
slaweet@mail.muni.cz

ABSTRACT

We propose a modular approach to development of a computerized adaptive practice system for learning of facts in areas with widely varying prior knowledge: decomposing the system into estimation of prior knowledge, estimation of current knowledge, and selection of questions. We describe specific realization of the system for geography learning and use data from the developed system for evaluation of different student models for knowledge estimation. We argue that variants of the Elo rating systems and Performance factor analysis are suitable for this kind of educational system, as they provide good accuracy and at the same time are easy to apply in an online system.

1. INTRODUCTION

Computerized adaptive practice [10] aims at providing students with practice in an adaptive way according to their skill, i.e. to provide the students with tasks that are most useful to them. Our aim is to make the development of such a system as automated as possible, particularly to enable the system to learn the relevant aspects of the domain from the data so that there is no need to rely on domain experts. This aspect is especially important for development of systems for small target groups of students, e.g. systems dealing with specialised topics or languages spoken by relatively small number of people (like Czech).

This work is focuses on the development of adaptive systems for learning of facts. In the terminology of the "knowledge learning instruction framework" [11] we focus on constant-constant knowledge components, i.e. knowledge components with a constant application condition and a constant response. We are particularly concerned with learning of facts in areas where students are expected to have nontrivial and highly varying prior knowledge, e.g. geography, biology (fauna, flora), human anatomy, or foreign language vocabulary. To show the usefulness of focusing on estimation of prior knowledge, Figure 1 visualizes the significant differences in prior knowledge of African countries.

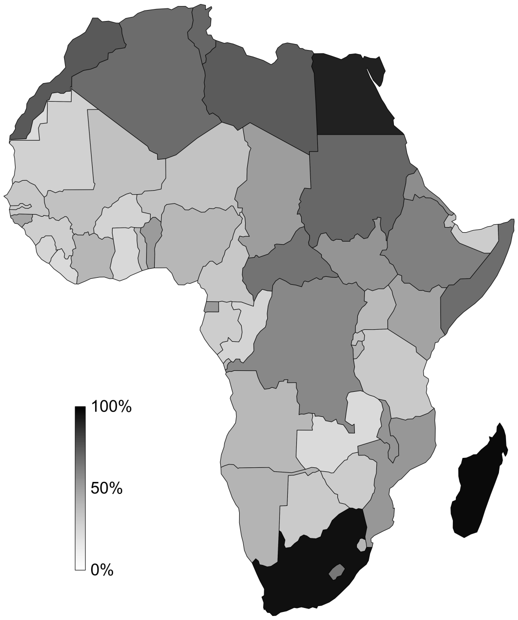


Figure 1: Map of Africa colored by prior knowledge of countries, the shade corresponds to the probability of correct answer for an average user of `slepemapy.cz`.

To achieve effective learning in domains such as geography it is necessary to address several interrelated issues, particularly the estimation of knowledge, the modeling of learning, the memory effects (spacing and forgetting), and the question selection.

The above-mentioned issues have been studied before, but separatedly in different context. Adaptation has been studied most thoroughly in the context of *computerized adaptive testing* (CAT) with the use of the item response theory [3]. In CAT the goal is the testing, i.e. to determine the skill of students. Therefore, the focus of CAT is on precision and statistical guarantees. It usually does not address learning (students' skill is not expected to change during a test) and motivation. In our setting the primary goal is to improve the skill; estimation of the skill is only a secondary goal which helps to achieve the main one. Thus the statistical accuracy of the estimation is not so fundamental as it is in CAT. On the other hand, the issues of learning, forgetting, and motivation are crucial for adaptive practice.

Another related area is the area of *intelligent tutoring systems* [23]. These systems focus mainly on learning of more complex cognitive skills than learning of facts, e.g. mathematics or physics. The modeling of learning is widely studied in this context, particularly using the popular Bayesian knowledge tracing model [2]. A lot of research focuses on the acquisition of skills, less attention is given to the prior knowledge and the forgetting (see e.g. [15, 20]).

The learning of facts is well studied in the research of *memory*, e.g. in the study of spacing and forgetting effects [16] and spaced repetition [9]. These studies are not, however, usually done in a realistic learning environment, but in a laboratory and in areas with little prior knowledge, e.g. learning of arbitrary word lists, nonsense syllables, obscure facts, or Japanese vocabulary [4, 16]. Such approach facilitates interpretation of the experimental results, but the developed models are not easily applicable in educational setting, where prior knowledge can be an important factor. There are also many implementations of the spaced repetition principle using "flashcard software" (well known example is SuperMemo), but these implementations usually use scheduling algorithms with fixed ad-hoc parameters and do not try to learn from collected data (or only in a limited way). The spaced repetition was also studied specifically for geography [26], but only in a simple setting.

In this work we propose both a general structure and a specific realization of a computerized adaptive practice system for learning of facts. We have implemented an instance of such system for learning geography, particularly names of countries (`slepemapy.cz`, the system is so far implemented only in Czech). Data from this system are used for the evaluation (over 2 500 students, 250 000 answers). To make the description more concrete and readable, we sometimes use the terminology of this system, i.e., learning of country names. Nevertheless, the approach is applicable to many similar domains (other geographical objects, anatomy, biology, foreign vocabulary).

The functionality of the system is simple: it provides series of questions about countries ("Where is country X?", "What is the name of this country?") and students answer them using an interactive map. Questions are interleaved with a feedback on the success rate and a visualization of the estimated knowledge of countries. The core of the system lies in estimating students' knowledge and selecting suitable questions.

We decompose the design of such system into three steps and treat each of these steps independently:

1. *Estimation of prior knowledge.* Estimating the probability that a student s knows a country c before the first question about this country. The estimate is based on previous answers of the student s and on answers of other students about the country c.
2. *Estimation of current knowledge.* Estimating the probability that the student s knows a country c based on the estimation of prior knowledge and a sequence of previous answers of student s on question about country c.
3. *Selection of question.* Selection of a suitable question for a student based on the estimation of knowledge and the recent history of answers.

Each of these issues is described and evaluated in a single section. The independent treatment of these steps is a useful simplifications, since it makes the development of the system and student models more tractable. Nevertheless, it is clearly a simplification and we discuss limitations of this approach in the final section.

2. BACKGROUND

In this section we briefly describe some of the relevant models that are used in the realization and evaluation of our approach.

2.1 Bayesian Knowledge Tracing

Bayesian knowledge tracing (BKT) [2, 21] is a well-known model for modeling of learning (changing skill). It is a hidden Markov model where skill is the binary latent variable (either learned or unlearned). The model has 4 parameters[1]: probability that the skill is initially learned, probability of learning a skill in one step, probability of incorrect answer when the skill is learned (slip), and probability of correct answer when the skill is unlearned (guess). The skill estimated is updated using a Bayes rule based on the observed answers. Parameter estimation can be done using the Expectation Maximization algorithm or using the exhaustive search.

2.2 Rasch Model

Basic model in the item response theory is the Rasch model (one parameter logistic model). This model assumes the student's knowledge is constant and expressed by a skill parameter θ, the item's difficulty is expressed by a parameter b, and the probability of a correct answer is given by the logistic function:

$$P(correct|b,\theta) = \frac{1}{1+e^{-(\theta-b)}}$$

The standard way to estimate the parameters from data is to use the joint maximum likelihood estimation [3], which is an iterative procedure. In the case of multiple choice question with n options, the model is modified to use a shifted logistic function:

$$P(correct|b,\theta) = \frac{1}{n} + (1-\frac{1}{n})\frac{1}{1+e^{-(\theta-b)}}$$

2.3 Performance Factor Analysis

Performance factor analysis (PFA) [17] can be seen as an extension of Rasch model with changing skill. The skill, which is a logit of probability of a correct answer, is given by a linear combination of the item's difficulty and the past successes and failures of a student:

$$P(correct) = \frac{1}{1+e^{-m}}$$

$$m = \beta + \gamma s + \delta f$$

[1] BKT can also include forgetting. The described version corresponds to the variant of BKT that is most often used in research papers.

where β is the item difficulty, s and f are counts of previous successes and failures of the student, γ and δ are parameters that determine the change of the skill associated with correct and incorrect answer. Note that originally PFA [17] is formulated in terms of vectors, as it uses multiple knowledge components; for our analysis the one-dimensional version is sufficient.

2.4 Elo System

The Elo rating system [5] was originally devised for chess rating, i.e. estimating players skills based on results of matches. For each player i we have an estimate θ_i of his skill, based on the result R ($0 =$ loss, $1 =$ win) of a match with another player j; the skill estimate is updated as follows:

$$\theta_i := \theta_i + K(R - P(R = 1))$$

where $P(R = 1)$ is the expected probability of winning given by the logistic function with respect to the difference in estimated skills, i.e. $P(R = 1) = 1/(1 + e^{-(\theta_i - \theta_j)})$, and K is a constant specifying sensitivity of the estimate to the last attempt. An intuitive improvement, which is used in most Elo extensions, is to use an "uncertainty function" instead of a constant K. There are several extension to the Elo system in this direction, the most well-known is Glicko [6].

We can use the Elo system in student modeling, if we interpret a student's answer on an item as a "match" between the student and the item. Recently, several researchers have studied this kind of application of the Elo system in the educational data mining [10, 24, 25].

The basic Elo system (reinterpreted in the context of educational problems) also uses the logistic function and one parameter for each student and problem. Thus the Rasch model and the Elo system are in fact very similar models, the main principal difference is that the Rasch model assumes the constancy of parameters, the Elo system assumes a changing skill.

3. ESTIMATION OF PRIOR KNOWLEDGE

At first, we treat the estimation of prior knowledge. Our aim is to estimate the probability that a student s knows a country c based on previous answers of students s to questions about different countries and previous answers of other students to questions about country c – as a simplification (for an easier interpretation of data) we use only the first answer about each country for each student in this step.

3.1 Model

In the following text we use a key assumption that both students and studied facts are homogenous; we assume that we can model students' overall prior knowledge in the domain by a one-dimensional parameter. This assumption is reasonable for geography and students from Czech Republic (which is the case of our application), but would not hold for geography and mixed population or for a mix of facts from geography and chemistry. If the homogenity is not satisfied, we can group the students and facts into homogenous groups (e.g. students by their IP address, facts by an expert or by an automatic technique [1]) and then make predictions for each subgroup independently.

More specifically, we model the prior knowledge by the Rasch model, i.e. we have student parameter θ_s corresponding to the global knowledge of a student s of geography, the item parameter b_c corresponding to the difficulty of a country c, and the probability of a correct first answer is given by the logistic function $P(correct|s, c) = \frac{1}{1+e^{-(\theta_s - b_c)}}$.

As we mentioned above, the standard approach to the parameter estimation for the Rasch model is joint maximum likelihood estimation (JMLE). This is an iterative approach that is slow for large data, particularly it is not suitable for an online application, where we need to adjust estimates of parameters continuously.

Therefore, we also consider the application of the Elo rating system in this setting. Although the assumptions in this context are closer to the assumptions of the Rasch model (the global skill and the difficulty of items are rather constant), the Elo system is much more suitable for an online application and results with simulated data suggest that it leads to similar estimates [19].

3.2 Evaluation

The basic version of the Elo system with the constant update parameter K does not provide a good estimation – if the parameter K is small, the system takes long to learn skills and difficulties, if the parameter K is large, the behavior of the system is unstable (estimates are too dependent on a last few answers). Therefore, instead of the constant K we use an uncertainty function $\frac{a}{1+bn}$, where n is the order of the answer and a, b are parameters. Using a grid search we have determined optimal values $a = 1, b = 0.05$. This exact choice of parameter values is not important, many different choices of a, b provide very similar results.

This variant of the Elo system provides both fast coarse estimates after a few answers and stability in the long run (see Figure 2 A). It also provides nearly identical estimates as the joint maximum likelihood estimation (Figure 2 B, correlation 0.97). JMLE is computationally demanding iterative procedure, the Elo system requires a single pass of the data and can be easily used online. Since the estimates of the two methods are nearly identical, we conclude that the Elo system is preferable in our context.

Distribution of the difficulty parameters (Figure 2 C) reflects the target domain and student population. In our case the difficulty of countries for Czech students is skewed towards very easy items, which are mostly European countries. Difficult countries are mostly African. Skill parameters are distributed approximately normally.

We have tested the assumption of a single global skill by computing the skill for independent subsets of items (countries from different continents) and then checking the correlation between the obtained skill. Figure 2 D shows the results for two such particular "subskills", the correlation coefficient for this case and other similar pairs of subskills is around 0.6. Given that there is some intrinsic noise in the data and that the skills are estimated from limited amount of questions, this is quite high correlation. This suggests that the assumption of a global skill is reasonable.

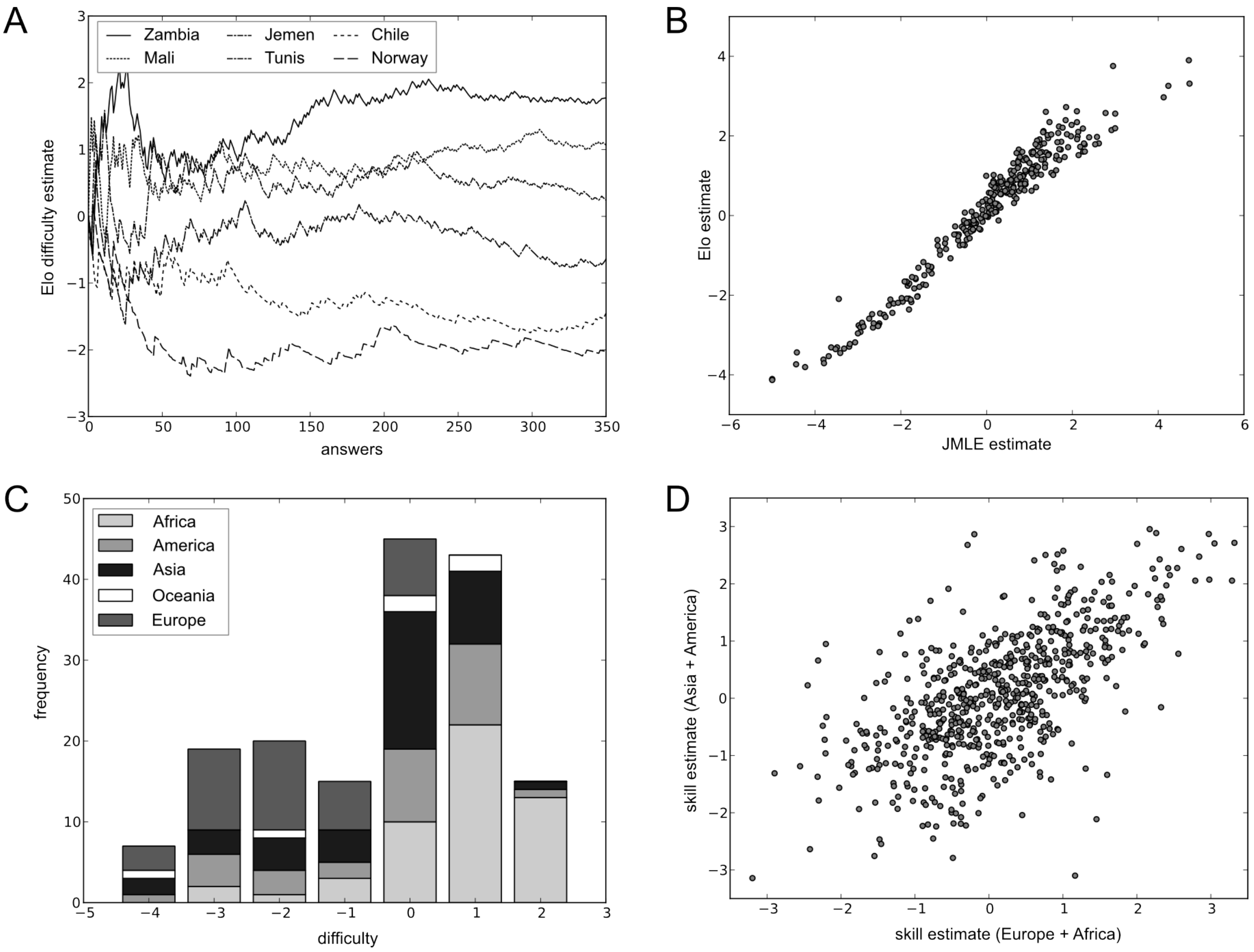


Figure 2: Estimation of prior knowledge: A) Development of estimates of difficulty of selected countries under Elo system, B) Comparison of Elo and JMLE difficulty estimates, C) Histogram of difficulty of countries, D) Correlation of "subskills" computed for different sets of countries.

4. ESTIMATION OF CURRENT KNOWLEDGE

We now turn to the estimation of a student's current knowledge, i.e. knowledge influenced by the repeatedly answering of questions about a country. The input data for this estimation are an estimate of prior knowledge (provided by the above described model) and the history of previous attempts, i.e. the sequence of previous answers (correctness of answers, question types, timing information).

4.1 Models

Several different models can be considered for the estimation of current knowledge. Bayesian knowledge tracing can be used in a straightforward way. In this context the probability of initial knowledge is given by the previous step. The probability of learning, guess, and slip are either given by a context (guess in the case of multiple choice question) or can be easily estimated using an exhaustive search. However, in this context the assumptions of BKT are not very plausible. BKT assumes a discrete transition from the unknown to the known state, which may be reasonable a simplification for procedural skills, but for declarative facts the development of the memory is gradual.

Assumptions of the Performance factor analysis are more relevant for the learning of facts. Instead of the item difficulty parameter β_i, used in the original version of PFA, we can use the estimate of the initial knowledge for a student s and a country c in our setting. This is given by the difference $\theta_s - b_c$.

A disadvantage of PFA is that it does not consider the order of answers (it uses only the summary number of correct and incorrect answers) and it also does not take into account the probability of guessing. Guessing can be important particularly in our setting, where the system uses multiple choice questions with variable number of options. To address these issues we propose to combine PFA with some aspects of the Elo system (in the following text we denote this version as PFAE – PFA Elo/Extended):

- K_{sc} is the estimated knowledge of a student s of a country c.
- The initial value of K_{sc} is provided by the estimation of prior knowledge: $K_{sc} = \theta_s - b_c$.
- The probability of correct answer to a question with n options is given by the shifted logistic function:

$$P(correct|K_{sc}, n) = \frac{1}{n} + (1 - \frac{1}{n})\frac{1}{1 + e^{-K_{sc}}}$$

- After a question with n options was answered, the estimated knowledge is updated as follows:
  - $K_{sc} := K_{sc} + \gamma \cdot (1 - P(correct|K_{sc}, n))$, if the answer was correct,
  - $K_{sc} := K_{sc} + \delta \cdot P(correct|K_{sc}, n)$, if the answer was incorrect.

The estimation can be further improved by taking into account the timing information. If two questions about the same item are asked closely one after another, then it can be expected that the student will answer the second one correctly, because the answer is still in his short term memory. In models based on a logistic function (PFA, PFAE) we can model this effect in the following way: the skill is "locally" increased by $\frac{w}{t}$, where t is the time (in seconds) between attempts and k is a suitable constant (optimal $w = 80$ for our data). It should be possible to further improve the model by a more thorough treatment of forgetting and spacing effects, e.g., by incorporating some aspects of the ACT-R model [16].

Another useful timing information is the response time. As the response time tends to be log-normally distributed [8, 22], we work with the logarithm of time. Intuitively, the higher knowledge of a country leads not only to higher probability of a correct answer, but also to a faster response. Figure 3 shows results of an experiment supporting this intuition – distribution of times of correct answers is shifted to lower values if the next answer on the same country is correct. This suggests that response time could be used to improve the estimation of knowledge. Indeed, even simple modification of the γ parameter in the PFA model (by comparison of the response time to mean response time) leads to a slight improvement in predictions. A more involved application of the response time requires a suitable normalization due to different speeds of students and different sizes of countries – it is much easier to click on China than on Vietnam.

4.2 Evaluation

The described models provide predictions of probability of a correct answer. To evaluate these models we need to choose a metric by which we measure performance of models. In educational data mining researchers often provide evaluation with respect to a chosen metric without providing any rationale for the particular choice. In some cases the choice of metric is not fundamental and different metrics lead to similar results (that is the case for above described experiments with estimating prior knowledge). However, the evaluation of models of the current knowledge is sensitive to the choice of a metric, and thus it is necessary to pay attention to this issue.

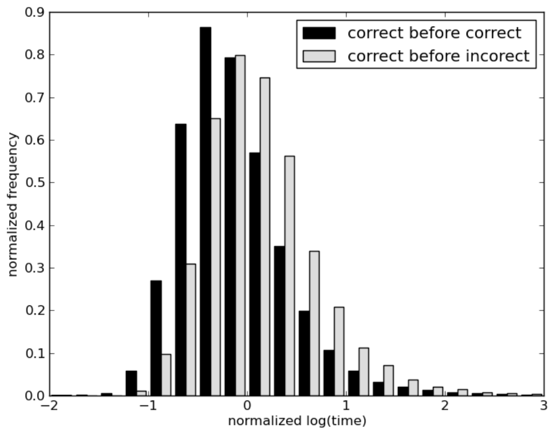


Figure 3: Normalized logarithm of time of correct answers, depending on whether the next answer about the country is answered correctly or incorrectly.

Let us review the most commonly used metrics in educational data mining and their suitability in our context. The mean absolute error (MAE) is not a good metric, since for unbalanced data it prefers models skewed towards the larger class. Consider a simulated student that answers correctly with constant probability 0.7. If we optimize a constant predictor with respect to the mean absolute error, the predicted probability is 1. The root mean square error (RMSE) is a similar measure that does not share this disadvantage and is thus preferable. The log-likelihood (LL) metric behaves similarly to RMSE except for predictions very close to 0 or 1. Since LL is unbounded, a single wrong prediction can degrade the performance of a model. To prevent this behaviour, an ad-hoc bound can be introduced in the computation of LL. Metrics like AIC and BIC are extensions of the log-likelihood penalizing large number of model parameters. All models described above have only very small number of parameters, and thus these metrics are not relevant for the current discussion. Another popular metric is the area under the receiver operating characteristic curve (AUC). This metric considers the prediction only in relative way – note that if all predictions are divided by 2, the AUC metric stays the same. In our application, however, the precision of the absolute prediction is important, since the value is used in computations that determine the choice of questions and number of options in multiple choice questions.

Thus it seems that the most suitable metrics from the commonly used ones is RMSE. Thus we use RMSE as our primary metric, i.e. to optimize values of model parameters. Table 1 provides a comparison of different models also for other metrics. We can see that the results are inconclusive regarding the comparison of BKT and PFA, but the newly proposed extension PFAE beats both the standard PFA and BKT models with respect to all three reported metrics. The results also show that the consideration of timing information further improves the performance of models.

Table 1: Model comparison.

model	RMSE	LL	AUC
BKT	0.262	-42048	0.668
PFA	0.265	-44740	0.669
PFA + time	0.262	-43088	0.695
PFAE	0.262	-41947	0.682
PFAE + time	0.259	-40623	0.714

For the reported evaluation we use models with “global” parameters, i.e., for example in the PFA and its extension we use the same parameters γ, δ for all countries and students. Thus the models have very small number of parameters (at most 4 for the extension with timing information) and can be easily fit by an exhaustive search. Since the number of data points is many orders larger (tens of thousands), overfitting is not an issue. It would be possible to use the “local” parameter values for individual countries and students, such variant would require an improved parameter estimation and a mechanism for dealing with uneven distribution of data among countries and students.

5. QUESTION SELECTION

We will now focus on the issue of the question selection. Based on the past performance of the student we want to select a suitable next question. In the context of our geography application the selection of a question consists of several partial decisions: which country to target, which type of the question to ask (“Where is X?” versus “What is the name of this country?”), and how many options to give a student to choose from.

Compared to the knowledge estimation, the question selection is much harder to evaluate, since we do not have a single, clear, easily measurable goal. The overall goal of the question selection is quite clear – it is the maximization of student learning. But it is not easy to measure the fulfilment of this general goal, since it depends also on the context of the learning. An experiment with pre-test, post-test, and fixed time in the system may provide a setting for an accurate evaluation of the different question selection strategies. Results of such experiment would, however, lack ecological validity, as many of the users of the system use the system on their own and with variable time in the system, so for example the issue of motivation is much more important than in a controlled experiment. A related work [18] presents this kind of controlled experiment for card selection in drill practice, the authors however provide comparison only with respect to a very simple cyclic selection technique and not to an evaluation of different alternatives of the selection algorithm. Another possibility is to use the time spent in educational system as a measure of quality of question selection. Here, however, the optimal choice with respect to this measure may not be optimal for learning, see [12] for a specific instance of an educational online game with this dynamics.

Thus at the moment we do not provide the evaluation of the question selection. We formulate general criteria that the question selection should satisfy and propose a specific approach to achieve these criteria.

5.1 Criteria

The question selection process should satisfy several criteria, which are partly conflicting. The criteria and their weight may depend on the particular application, the target student population, and student goals. We propose the following main criteria.

The selection of question should depend on an estimated *difficulty* of question. From the testing perspective, it is optimal to use questions with expected probability of a correct answer reaching 50%, because such question provide most information about students’ knowledge. However, 50% success rate is rather low and for most students it would decrease motivation. Thus in our setting (adaptive practice) it is better to aim for a higher success rate. At the moment we aim at 75%, similarly to previous work [7].

Another important issue is the *repetition* of questions. This aspect should be governed by the research about spacing effects [4, 16], particularly it is not sensible to repeat the same question too early.

It may be also welcome to have *variability* of question types. Different question types are useful mainly as a tool for tuning the difficulty of questions, but even if this is not necessary, the variability of question types may be meaningful criteria in itself, since it improves user experience, if used correctly.

5.2 Selecting Target Country

We propose to use the linear scoring approach to select a target country (the correct answer of the question). For each relevant attribute, we consider a scoring function that expresses the desirability of a given country with respect to this attribute. These scoring functions are combined using weighted sum, the country with highest total score is selected as a target. We consider the following attributes:

1. the probability the student knows the country,
2. time since the last question about the country,
3. the number of questions already answered by the student about the country.

Figure 4 shows the general shape of scoring functions for these attributes. Further we specify concrete formulas that approximate these shapes using simple mathematical functions.

The first case takes into account the relation between the estimated probability of a correct answer (P_{est}) and the target success rate (P_{target}). Assume that our goal is to ask a question where the student has 75% chance of a correct answer. The distance from the probability for the difficult countries (nearly 0% chance of the correct answer) is higher than for easy ones (almost 100%), so it is necessary to normalize it.

$$S_{prob}(P_{est}, P_{target}) = \begin{cases} \frac{P_{est}}{P_{target}} & \text{if } P_{target} \geq P_{est} \\ \frac{1-P_{est}}{1-P_{target}} & \text{if } P_{target} < P_{est} \end{cases}$$

The second scoring function penalizes countries according to the time elapsed since the last question, because we do not want to repeat countries in a short time interval when

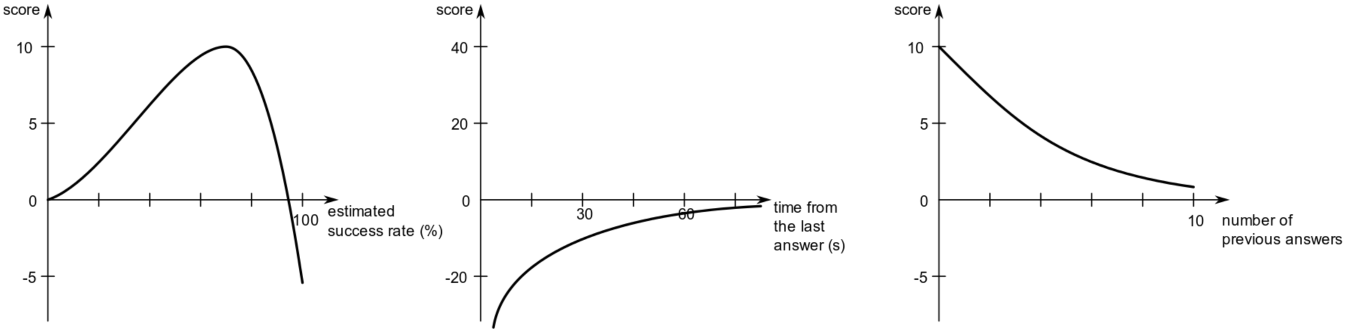


Figure 4: Desired contribution of different criteria to selection of target country.

they are still in short term memory. We use the function $S_{time}(t) = -1/t$, where t is time in seconds. Using just the above mentioned attributes the system would ask questions for only a limited pool of countries. To induce the system to ask questions about new countries we introduce the third scoring function that uses the total number n of questions for the given country answered by the student: $S_{count}(n) = 1/\sqrt{1+n}$. The total score is given as a weighted sum of individual scores, the weights are currently set manually, reflecting experiences with the system: $W_{prob} = 10$, $W_{count} = 10$, $W_{time} = 120$.

5.3 Choosing Options

Once the question's target is selected, the question can be adjusted according to the student's needs by using a multiple choice question with suitable number of options. For a multiple choice question the probability of a correct answer is the combination of the probability of guessing the answer (P_{guess}) and knowing the target country (P_{est})[2]:

$$P_{success} = P_{guess} + (1 - P_{guess}) \cdot P_{est}$$

As our goal is to get $P_{success}$ close to P_{target}, we would like to make P_{guess} close to

$$G = \frac{P_{target} - P_{est}}{1 - P_{est}}$$

For $G \leq 0$, we use open question (no options), otherwise we use n closest to $\frac{1}{G}$ as a number of options. For principal reasons the minimal possible value of n is 2, for practical reasons there is also an upper bound for n (more than 6 options would be confusing). The type of the question – "Where is country X?" or "What is the name of this country?" is currently selected randomly. In case of an open question the first type is always used.

When using multiple choice questions, we also need to choose the distractor options. Unlike other systems for practice dealing with text [13, 14], we work with well structured data, so the problem of option selection is easier. The choice of options can be based on domain information, e.g. geographically close countries or countries with similar names. The easiest way to choose good distractors is, however, to simply base the choice on past answers. We can take countries most commonly mistaken with the target country (in open questions) and select from them randomly. The random choice is weighted by the frequency of mistakes with the given country, for example Kamerun is most often confused with Niger (38%), Nigeria (27%), Central African Republic (10%), Republic of the Congo (9%), Gabon (6%), Ivory Coast (5%), Uganda (3%), and Guinea (2%).

6. DISCUSSION

We described the functionality of the system in three independent parts: the estimation of prior knowledge, the estimation of current knowledge, and the selection of a question. The independent treatment of these steps is, however, a simplification, as there is an interaction between these steps.

In our treatment, only the first answer about a given item is taken as an indication of a prior knowledge, other answers are considered as an indication of changes in knowledge. But for example the second answer, clearly, also contains some information about prior knowledge. A more precise models should be possible by incorporating more integrated approach to the estimation of prior and current knowledge.

The selection of a question was treated as a subsequent step after the estimation of knowledge, but in reality there is a feedback loop: the estimation of knowledge influences the selection of a question and the selection of a question determines the data that are collected and used for the estimation of knowledge. Since the collected data are partially determined by the model used, there may be a bias in the data towards certain questions, and this bias may, in a subtle way, influence the evaluation. For example, if the model overestimates the knowledge of students, the question selection stops asking questions about items too early, which means that the system does not collect data that would contradict the overestimated knowledge. The question selection procedure may be also modified in such a way to collect data most useful for improving the precision of the estimation. The study of these interactions may be more important than differences between different models or estimation procedures, which typically get most attention in current research in student modeling.

Acknowledgements

Authors thank anonymous reviewers for valuable comments, Jiří Řihák and Juraj Nižnan for fruitful discussions, and Tereza Doležalová for language assistance.

[2] This is, of course, simplification since a multiple choice question can also be answered by ruling out distractor options. But if the distractors are well chosen; this simplification is reasonable.

7. REFERENCES

[1] P. Boroš, J. Nižnan, R. Pelánek, and J. Řihák. Automatic detection of concepts from problem solving times. In *Proc. of International Conference on Artificial Intelligence in Education (AIED 2013)*, volume 7926 of *LNCS*, pages 595–598. Springer, 2013.

[2] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[3] R. De Ayala. *The theory and practice of item response theory*. The Guilford Press, 2008.

[4] P. F. Delaney, P. P. Verkoeijen, and A. Spirgel. Spacing and testing effects: A deeply critical, lengthy, and at times discursive review of the literature. *Psychology of learning and motivation*, 53:63–147, 2010.

[5] A. E. Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.

[6] M. E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.

[7] B. R. Jansen, J. Louwerse, M. Straatemeier, S. H. Van der Ven, S. Klinkenberg, and H. L. Van der Maas. The influence of experiencing success in math on math anxiety, perceived math competence, and math performance. *Learning and Individual Differences*, 24:190–197, 2013.

[8] P. Jarušek and R. Pelánek. Analysis of a simple model of problem solving times. In *Proc. of Intelligent Tutoring Systems (ITS)*, volume 7315 of *LNCS*, pages 379–388. Springer, 2012.

[9] J. D. Karpicke and H. L. Roediger. Repeated retrieval during learning is the key to long-term retention. *Journal of Memory and Language*, 57(2):151–162, 2007.

[10] S. Klinkenberg, M. Straatemeier, and H. Van der Maas. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824, 2011.

[11] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[12] D. Lomas, K. Patel, J. L. Forlizzi, and K. R. Koedinger. Optimizing challenge in an educational game using large-scale design experiments. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 89–98. ACM, 2013.

[13] R. Mitkov, L. A. Ha, and N. Karamanis. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2):177–194, 2006.

[14] J. Mostow, B. Tobin, and A. Cuneo. Automated comprehension assessment in a reading tutor. In *ITS 2002 Workshop on Creating Valid Diagnostic Assessments*, pages 52–63, 2002.

[15] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[16] P. I. Pavlik and J. R. Anderson. Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, 29(4):559–586, 2005.

[17] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *Proc. of Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.

[18] P. Pavlik Jr, T. Bolster, S.-M. Wu, K. Koedinger, and B. Macwhinney. Using optimally selected drill practice to train basic facts. In *Intelligent Tutoring Systems*, pages 593–602. Springer, 2008.

[19] R. Pelánek. Time decay functions and elo system in student modeling. In *Proc. of Educational Data Mining (EDM)*, 2014.

[20] Y. Qiu, Y. Qi, H. Lu, Z. A. Pardos, and N. T. Heffernan. Does time matter? modeling the effect of time with bayesian knowledge tracing. In *Proc. of Educational Data Mining (EDM)*, pages 139–148, 2011.

[21] B. van de Sande. Properties of the bayesian knowledge tracing model. *Journal of Educational Data Mining*, 5(2):1, 2013.

[22] W. Van der Linden. A lognormal model for response times on test items. *Journal of Educational and Behavioral Statistics*, 31(2):181, 2006.

[23] K. Vanlehn. The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265, 2006.

[24] K. Wauters, P. Desmet, and W. Van Den Noortgate. Adaptive item-based learning environments based on the item response theory: possibilities and challenges. *Journal of Computer Assisted Learning*, 26(6):549–562, 2010.

[25] K. Wauters, P. Desmet, and W. Van Den Noortgate. Monitoring learners' proficiency: Weight adaptation in the elo rating system. In *EDM*, pages 247–252, 2011.

[26] D. M. Zirkle and A. K. Ellis. Effects of spaced repetition on long-term map knowledge recall. *Journal of Geography*, 109(5):201–206, 2010.

Alternating Recursive Method for Q-matrix Learning

Yuan Sun
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Japan
yuan@nii.ac.jp

Shiwei Ye
School of Electronic and Communication Engineering, University of China Academy of Science, China
shwye@ucas.ac.cn

Shunya Inoue
Tokyo Kasei University
1-18-1 Kaga, Itabashi-ku, Japan
inoues@tokyo-kasei.ac.jp

Yi Sun
School of Computer and Control Engineering, University of China Academy of Science, China
sunyi@ucas.ac.cn

ABSTRACT

The key issue affecting Cognitive Diagnostic Models (CDMs) is how to specify attributes and the Q-matrix. In this paper, we first attempt to use the Boolean Matrix Factorization (BMF) method to express conjunctive models in CDMs. Because BMF is an NP-hard problem [2], we propose a recursive method that updates the attribute matrix (its rank equals to one) in each step. As Boolean algebra is irreversible, it requires time to recursively compute and update the matrix, especially when the number of attributes is large. To speed up computations, we use a Heaviside step function, which allows us to decompose the recursive computing process into normal non-negative matrices and get the results by mapping them back into a Boolean matrix. Two different algorithms are presented: a deterministic heuristic algorithm and a stochastic algorithm. Simulation results from an actual test show that the proposed method can learn the original Q-matrix well from item response data.

Keywords

Q-matrix, Cognitive Diagnostic Models (CDMs), Boolean Matrix Factorization (BMF), Conjunctive Models, Alternating Recursive Algorithm.

1. INTRODUCTION

Cognitive diagnostic assessment (CDA) has attracted a great deal of attention in the psychological and educational measurement fields. It not only reports students' total test scores, but also assesses students' mastery of attributes, which refers to their knowledge, skills or strategies, so that it can provide students or their teachers with diagnostic information on their strengths and weaknesses. A key issue in CDA is to correctly specify the so-called Q-matrix introduced by Tatsuoka [23], which associates the items and attributes of students a diagnostic test intends to assess.

To construct a Q-matrix, experts in the particular domain usually need to specify what attributes are key knowledge or skills for students to acquire. There are two approaches to constructing it. One is to develop test items and specify the Q-matrix for the items particular to cognitive diagnostic purposes. The other way is to apply diagnostic modeling to an existing test and specify the Q-matrix for the test items. Once the Q-matrix has been specified and items have been administered in a test, the items are calibrated to one of the cognitive diagnostic models, in which a known Q-matrix is typically assumed [24, 11, 13, 12, 15, 4]. However, if the Q-matrix is not specified appropriately, it could seriously affect the models' goodness of fit [21]. In that sense, the key issues affecting CDMs are how to define attributes and specify the Q-matrix. On the other hand, there is a big problem with the current manual method of generating the Q-matrix. When the domain or content of the tests is broader, it is an extremely difficult task for experts to specify attributes and the Q-matrix manually. This difficulty and their time-consuming nature could be reasons why CDMs are still not as popular as they should be in educational fields. Automatic and intelligent help to alleviate this difficulty is obviously desirable, and even necessary. During the past decade, the problem of how to map test items into latent skills based on students' test responses has become a hot topic in psychometrics and in educational data mining (see [16, 17, 14, 28, 7, 6, 5, 1, 3] for recent contributions).

In the previous studies [17] and [28], the proposed DINA model-based Q-matrix learning approach use EM algorithm to solve the Q-matrix. They need to involve a matrix T(Q) on a scale of $2^n \times 2^K$, wherein n, K is the number of items and attributes, respectively, which makes it extremely difficult to achieve for even a medium sized Q-matrix. In [7], an alternative least square method was proposed to solve the Q-matrix, where they use a matrix inverse operation that values may appear negative and estimated values for all the parameters are real instead of binary 0 or 1. To finally map the results into binary 0 or 1, there need to assume an appropriate threshold to truncate the values, but the threshold selected can't be automatically given, it can only be artificial and thus subjective. Moreover, the most serious problem for the previous methods so far ([7][17][28]) is that the number of attributes K have to be set in advance. They can't be determined by students' real response dataset R matrix, which is extremely critical for data driven Q-matrix learning approach.

Here we present new algorithms to learn the Q-matrix automatically from the students' item response matrix on the basis of the Boolean Matrix Factorization (BMF) technique and demonstrate how the methods can yield promising results from simulation data. This is the first attempt to apply BMF to Q matrix discovery.

2. CONJUNCTIVE MODELS AND BOOLEAN MATRIX FACTORIZATION

Various statistical models have been built around the Q-matrix [23, 15, 13, 11, 12, 26, 4, 21]. The applicable statistical model, such as conjunctive or compensatory models, will vary depending on whether there are hierarchical relations or interactions among the defined attributes. Conjunctive models assume that students can get "correct" for an item only if they have mastered all attributes required for that item and only a fraction of them results in a success probability equal to that of a student possessing none of the attributes. The DINA model (deterministic inputs, noisy-and-gate; [13]) is one of the simplest and most widely studied conjunctive models. This study makes the conjunctive assumption for dichotomously scored test items in CDMs.

2.1 Item Response Matrix, Q-matrix, and Knowledge States Matrix

Suppose there are K attributes in a particular domain. Then a student's attribute patterns $\boldsymbol{\alpha_i} = (\alpha_{i1,}\ \alpha_{i2...,}\ \alpha_{iK})$, called knowledge states, indicate the student's mastery status in terms of the K attributes. α_{ik}=1 indicates the ith student's mastery of attribute k and α_{ik}=0 indicates non-mastery of the attribute. As stated above, the Q-matrix indicates the required attributes for each item. The entry of the Q-matrix (denoted q_{jk}) equals one if item j requires attribute k; and zero otherwise. The Q-matrix is used to establish a relationship between the students' responses and the attribute. It is assumed that the item responses are determined by the attributes involved in each item and the attributes mastered by each student.

For conjunctive models, based on the students' knowledge states and Q-matrix for items, an ideal item response matrix R can be generated, whose element r_{ij} is typically represented in the following form.

$$r_{i,j}(A,Q) = \xi_{ij} = \prod_{k=1}^{K} \alpha_{i,k}^{q_{j,k}} = \begin{cases} 1 & (\alpha_{i,k} \geq q_{j,k} : k = 1,.., K) \\ 0 & (\alpha_{i,k} < q_{j,k} : \exists k \in \{1,..., K\}) \end{cases} \quad \text{......}(1)$$

Here r_{ij} is the latent response variable of the ith student with the latent knowledge states $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{jK})$ to the jth item, indicating whether the student i has all the attributes required for item j. It represents a deterministic prediction of item response from each student's knowledge state.

An example of eight students' knowledge states and their ideal response patterns to seven items identified by a Q-matrix is illustrated. Given two matrices $\boldsymbol{A}$ and $\boldsymbol{Q}$, the ideal response matrix $\boldsymbol{R}$ will be obtained as follows.

$$\boldsymbol{A}=\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{Q}=\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix},$$

$$\boldsymbol{R}=\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

It is important to note that mapping of knowledge states to ideal response patterns is not a one-to-one correspondence; rather, given a particular set of items and a particular Q-matrix, two or more different knowledge states can result in the same ideal response pattern.

2.2 Boolean Matrix Factorization

A recent technique based on Boolean Matrix Factorization (BMF) has been shown to be extremely effective for getting valuable results in binary data analyses [8, 9, 10, 18, 19, 20, 25, 27].

Definition: If $P = (p_{i,j})^{m\times k} \in \{0,1\}^{m\times k}$ and $Q = (q_{i,j})^{k\times n} \in \{0,1\}^{k\times n}$, the Boolean product of P and Q is defined as

$$P \odot Q = (\bigvee_{s=1}^{k} p_{i,\mathrm{s}} q_{\mathrm{s,j}})^{m\times n} \in \{0,1\}^{m\times n} \cdot$$

In our previous work involving BMF, we verified that an ideal response matrix R can be expressed in terms of the following Boolean relations of the knowledge states matrix $\boldsymbol{A}$ and the $\boldsymbol{Q}$-matrix (see [30] for details).

$$\boldsymbol{R} = \overline{\boldsymbol{A} \odot \boldsymbol{Q}^T} \quad \text{...} \ (2)$$

Here, $\boldsymbol{A}$ and $\boldsymbol{Q}$ are a m-students by K-attributes binary mastery matrix and an n-items by K-attributes binary Q-matrix, for student i=1,..,m; item j=1,…,n; and attribute k=1,…,K. The bar notation in equation (2) represents logical NOT operation (i.e. $\overline{0} = 1, \overline{1} = 0$), and T represents the transpose of the matrix. The notation ⊙ represents the Boolean product. These notations will apply hereafter.

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \vdots \\ \boldsymbol{\alpha}_m \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1K} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mK} \end{pmatrix}$$

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_n \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1K} \\ q_{21} & q_{22} & \cdots & q_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nK} \end{pmatrix}$$

The process of Boolean matrix factorization is to determine the matrices $\boldsymbol{Q}$ and $\boldsymbol{A}$ from $\boldsymbol{R}$, and the goal of a factorization algorithm is to minimize the estimated R with R in equation (2). It is clear that equation (2) becomes much more powerful than the usual equation (1) as the number of latent attributes increases.

3. ALTERNATE RECURSIVE METHOD FOR Q-MATRIX LEARNING

3.1 Approach to Response Matrix through Attribute Latent Space Perturbation

Instead of approximating the item response matrix $\boldsymbol{R}$, for simplicity, we will approach its complementary matrix $\overline{\boldsymbol{R}}$ by using matrix $\boldsymbol{H}$=$\boldsymbol{X}$ ⊙ $\boldsymbol{Y}^T$. From equation (2), it is implicit that $X = \overline{A}$ and $Y = Q$, where $\boldsymbol{A}$ and $\boldsymbol{Q}$ are the initial knowledge state matrix and Q-matrix, respectively. In order to approximate $\overline{\boldsymbol{R}}$, we need to perturb $\boldsymbol{X}$ and $\boldsymbol{Y}$ by adding one-column vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ to the latent attribute space (the operations are denoted as $\boldsymbol{X}_{\text{new}}$=[$\boldsymbol{X}$, $\boldsymbol{x}$], and $\boldsymbol{Y}_{\text{new}}$=[$\boldsymbol{Y}$, $\boldsymbol{y}$]). Thus, $\boldsymbol{H}_{\text{new}}$=$\boldsymbol{X}_{\text{new}}$⊙ $(\boldsymbol{Y}_{\text{new}})^T$, which minimizes the following objective error function:

$$E(\boldsymbol{x},\boldsymbol{y})=(\| \ \overline{\boldsymbol{R}} - \boldsymbol{X}_{\text{new}} \odot (\boldsymbol{Y}_{\text{new}})^T \|_F)^2$$

where $\| \ \|_F$ indicates the Frobenius norm of the matrix. Without loss of generality, we can fix $\boldsymbol{y}$ and choose $\boldsymbol{x}$ to minimize the error function $E(\boldsymbol{x},\boldsymbol{y})$:

$$\left\| \overline{R} - X_{\text{new}} \odot Y_{\text{new}}^T \right\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (\overline{r}_{ij} - h_{ij} \vee x_i y_j)^2$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{n}(\overline{r}_{ij}+h_{ij}\vee x_i y_j-2\overline{r}_{ij}(h_{ij}\vee x_i y_j))$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{n}(\overline{r}_{ij}+h_{ij}+\overline{h}_{ij}x_i y_j-2\overline{r}_{ij}(h_{ij}+\overline{h}_{ij}x_i y_j))$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{n}(\overline{r}_{ij}-\overline{r}_{ij}h_{ij}+h_{ij}-\overline{r}_{ij}h_{ij})+\sum_{i=1}^{m}\sum_{j=1}^{n}(\overline{h}_{ij}x_i y_j-2\overline{r}_{ij}\overline{h}_{ij}x_i y_j)$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{n}(\overline{r}_{ij}(1-h_{ij})+h_{ij}(1-\overline{r}_{ij}))+\sum_{i=1}^{m}x_i(\sum_{j=1}^{n}(1-\overline{r}_{ij})\overline{h}_{ij}y_j-\sum_{j=1}^{n}\overline{r}_{ij}\overline{h}_{ij}y_j)$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{n}(\overline{r}_{ij}\overline{h}_{ij}+r_{ij}h_{ij})+\sum_{i=1}^{m}x_i(\sum_{j=1}^{n}r_{ijij}\overline{h}_{ij}y_j-\sum_{j=1}^{n}\overline{r}_{ij}\overline{h}_{ij}y_j)$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{n}(\overline{r_{ij}}\overline{h}_{ij}+r_{ij}h_{ij})+\sum_{i=1}^{m}x_i\sum_{j=1}^{n}r_{ij}\overline{h}_{ij}y_j-\sum_{i=1}^{m}x_i\sum_{j=1}^{n}\overline{r}_{ij}\overline{h}_{ij}y_j \quad \cdots (3)$$

When $u,v\in\{0,1\}$, the $\max\{u,v\}=u+v-uv=v+u\overline{v}$ is used to obtain equation (3). The three terms in equation (3) consist of the original ***H*** matrix error and two perturbation errors. We express them as $E_{orig}=\sum_{i=1}^{m}\sum_{j=1}^{n}(\overline{r}_{ij}\overline{h}_{ij}+r_{ij}h_{ij})$, $E_P^{+}(i)=\alpha_i=\sum_{j=1}^{n}r_{ij}\overline{h}_{ij}y_j$, and $E_P^{-}(i)=\beta_i=\sum_{j=1}^{n}\overline{r}_{ij}\overline{h}_{ij}y_j$. Obviously, E_{orig} is the already existing error term which is independent of how ***x*** and ***y*** are optimized. Therefore, in what follows, we focus on the α_i and β_i error terms.

Observation 1: For i=1,2,…,m, if $\overline{r}_{ij}=0$ and h_{ij}=0, when x_i=1 the total error of the function E(***x***,***y***) increases by a positive factor $E_P^{+}(i)=\alpha_i=\sum_{j=1}^{n}r_{ij}\overline{h}_{ij}y_j$. Although the original approaching matrix ***H*** doesn't have any error with respect to matrix $\overline{\boldsymbol{R}}$, the new approaching matrix is such that $1=\boldsymbol{H}_{\text{new}}(i,j)=h_{ij}\vee x_i y_j>\overline{r}_{ij}=0$; hence, the total error increases.

Observation 2: For i=1,2,…,m, if $\overline{r}_{ij}=1$ and h_{ij}=0, when x_i=1 the total error of E(***x***,***y***) decreases by a positive factor $E_P^{-}(i)=\beta_i=\sum_{j=1}^{n}\overline{r}_{ij}\overline{h}_{ij}y_j$. Although the original approaching matrix ***H*** has an error with respect to matrix $\overline{\boldsymbol{R}}$, the new approaching matrix is such that $1=\boldsymbol{H}_{\text{new}}(i,j)=h_{ij}\vee x_i y_j=\overline{r}_{ij}=1$; hence, the total error decreases.

The above two observations imply that if $\alpha_i\le\beta_i$, setting x_i=1 will decrease the total error of E(***x***,***y***), and if $\alpha_i>\beta_i$, we obviously have to set x_i=0 directly to ensure that the total error does not increase. To control the error in the case of $\alpha_i\le\beta_i$ and setting x_i=1, we exploit two heuristic methods by using a determinate function or by using a sigmoid function based on adjusting coefficients ρ and τ, which are constant parameters bigger than zero, and satisfies $\alpha_i\le\rho\cdot\beta_i$, 0<ρ≤1. The two heuristic algorithms based on the BMF approach are illustrated in Figure 1.

```
Step 1. Input the response matrix $\overline{\boldsymbol{R}}\in\{0,1\}^{m\times n}$, Initialize
    $\boldsymbol{H}=0$, $\boldsymbol{X}=\Phi$, $\boldsymbol{Y}=\Phi$; $k=1$; for given $K$
Step 2. Compute $\boldsymbol{P}=(r_{ij}\overline{h}_{ij})_{m\times n}$, $\boldsymbol{Q}=(\overline{r}_{ij}\overline{h}_{ij})_{m\times n}$, $y(j)=1$, when
    $j=k$; otherwise, $y(j)=0$; $j=1,\ldots,n$
Step 3. Calculate $\boldsymbol{\alpha}=\boldsymbol{Py}$, $\boldsymbol{\beta}=\boldsymbol{Qy}$, and update x based on one
    of the following two methods:
  1) (update method I) $\boldsymbol{x}=\theta(\rho\cdot\boldsymbol{\beta}-\boldsymbol{\alpha})$, $0<\rho\le1$ is a given
     parameter
  2) (update method II) Prob $(\boldsymbol{x}(i)=1|\alpha_i,\beta_i)=\frac{1}{1+e^{-\tau(\beta_i-\alpha_i)}}$ ,
     where τ is a given  parameter
Step 4.  Compute $\boldsymbol{u}=\boldsymbol{P}^T\boldsymbol{x}$, $\boldsymbol{v}=\boldsymbol{Q}^T\boldsymbol{x}$ and select one of the
    following approaches to update y:
  1)  $\boldsymbol{y}=\theta(\rho\cdot\boldsymbol{v}-\boldsymbol{u})$,  where $0<\rho\le1$ is a given parameter;
  2) Prob $(\boldsymbol{y}(j)=1|u_j,v_j)=\frac{1}{1+e^{-\tau(v_i-u_i)}}$ where τ is a given
     parameter;
Step 5. Repeat Step 3 and Step 4, till x and y converge
    (update method I) or the distributions of x and y become
    stable (update method II)
Step 6. Set $\boldsymbol{H}=\boldsymbol{H}\vee(\boldsymbol{xy}^T)$, $\boldsymbol{X}=[\boldsymbol{X},\boldsymbol{x}]$, and $\boldsymbol{Y}=[\boldsymbol{Y},\boldsymbol{y}]$, $k=k+1$, if
    $k>K$, output H, X, Y and break,
Else go to step2
```

Figure 1. Perturbation Approaching Algorithms (PAA)

For PAA algorithm we have the following convergence theorem 1.

Theorem 1.

1) For PPA deterministic algorithms, suppose H be the original matrix and Hnew be the updated matrix, then $\|\overline{\boldsymbol{R}}-X_{\text{new}}\odot(Y_{\text{new}})^{T}\|_F\le\|\overline{\boldsymbol{R}}-H\|_F$. holds.

2) By the same token, for PPA stochastic algorithm, it converges each step at the final equilibrium distribution $P(x,y)=(1/Z)\exp(-x^{T}(P-Q)y/\tau)$, wherein Z is a normalized constant for probability distribution .

Proof: Define local energy function as $Eloc(x,y)=x^{T}(P-Q)y$.

1) For PPA deterministic updating algorithm, if we see it as Hopfield neural network bidirectional associative computing updating algorithm [29], it holds according to Hopfield convergence theorem of bidirectional associative memory (BAM).

2) For the PPA stochastic algorithm, if we see it as a limitation Boltzmann machine in neural network [29] , according to the Boltzmann convergence theorem, every step of the final x, y equilibrium distribution will be $P(x,y)=(1/Z)\exp(-x^{T}(P-Q)y/\tau)$, wherein Z is a probability distribution of a normalization factor , τ is called Boltzmann machine annealing temperature which should be gradually reduced during the iteration . QED.

As each step of the optimization process are discrete variables {0,1} quadratic optimization problem , they are all NP-hard problem according to the computational complexity theory. Therefore there will always be a local optimum, which is also the reason why sometimes PPA stochastic algorithm is used. While deterministic algorithm can always guarantee that each time one attribute is added, the error of approaching response matrix with the real response matrix R can be reduced accordingly, although it may eventually fall into local minimum of the local error Eloc. The stochastic algorithm uses that random error energy can increase certain probability in an iterative process, so when the annealing temperature decreases slowly enough, the conclusions based on simulated annealing algorithm, it is possible to converge to the probability of a global optimum.

3.2 A Fast Alternating Recursive Algorithm

Due to the monotony of the Boolean matrix product, the algorithm we proposed in 3.1 often relapses into a local optimal solution. We suggested an alternative recursive algorithm to improve the accuracy of approaching matrix $\boldsymbol{H}$. For given $\boldsymbol{H}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_K) \odot (\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_K)^T$, let $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_K)$, $\boldsymbol{Y}=(\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_K)$, $\boldsymbol{X}^i=(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{i-1},\boldsymbol{x}_{i+1},\ldots,\boldsymbol{x}_K)$, $\boldsymbol{Y}^i=(\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_{i-1},\boldsymbol{y}_{i+1},\ldots,\boldsymbol{y}_K)$, $\boldsymbol{H}^i= X\odot (\boldsymbol{Y}^i)^T$. Therefore, we have $\boldsymbol{H}=\boldsymbol{H}^i\vee \boldsymbol{x}_i(\boldsymbol{y}_i)^T$. The index i can be chosen by using random or deterministic methods. Using the results of Perturbation Approaching Algorithms (PAA) as initial values, one can iteratively optimize the perturbation vector $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ for fixed $\boldsymbol{X}^i$ and $\boldsymbol{Y}^i$. In practice, when the dimension of attribute space K is quite large, we need to reuse the previous results for the next iteration on the perturbation vectors $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ in order to reduce the time complexity of the algorithm. However, because of the special characteristics of the Boolean matrix product, we cannot make use of $\boldsymbol{H}^j$ in the previous round, since $\boldsymbol{H}^j\neq\boldsymbol{H}^i\vee \boldsymbol{x}_i(\boldsymbol{y}_i)^T \wedge(\bar{\boldsymbol{x}}_j\bar{\boldsymbol{y}}_j^T)$. Luckily, the algorithm can be sped up by using the general matrix product instead of the Boolean matrix product if we introduce the Heaviside step function as follows.

Define $\theta(x)$ as the Heaviside step function.

$$\theta(x)=\begin{cases}1 & x>0\\ 0 & x\le 0\end{cases} \quad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Given a matrix $B=(b_{i,j})^{m\times n}\in R^{m\times n}$, we define its Heaviside step function as $\theta(B)=(\theta(b_{i,j}))^{m\times n}\in R^{m\times n}$; namely, $\theta(x)$ acts on every element of matrix B.

Property 1. $P=(p_{i,j})^{m\times n}\in R_+^{m\times n}$ and $Q=(q_{i,j})^{m\times n}\in R_+^{m\times n}$ Here, $R_+=\{x, x\ge 0\}$. If $\lambda\ge 1$ exists such that $P\ge\lambda Q$, we have $\theta(P)=\theta(P-Q)$.

Proof: If $q_{i,j}=0$, it immediately follows that

$$\theta(p_{i,j})=\theta(p_{i,j}-q_{i,j})=0.$$

On the other hand, if $q_{i,j}>0$, from the assumptions, we have

$$p_{i,j}-q_{i,j}>(\lambda-1)q_{i,j}$$

Therefore, $\theta(p_{i,j})=\theta(p_{i,j}-q_{i,j})=1$,

Hence, we have $\theta(P)=\theta(P-Q)$. QED.

Property 2. $P=(p_{i,j})^{m\times n}\in R_+^{m\times n}$ and $Q=(q_{i,j})^{m\times n}\in R_+^{m\times n}$, here $R_+=\{x, x\ge 0\}$. Then we have $\theta(PQ)=\theta(P)\odot\theta(Q)$

Proof: Let $G=(g_{i,j})^{n\times k}=PQ$. If $g_{i,j}=0$, then $\theta(g_{i,j})=0$. While if $g_{i,j}>0$, then $\theta(g_{i,j})=1$.

In case of $g_{i,j}=\sum_{t=1}^{k}p_{i,t}q_{t,j}=0$, because P and Q are nonnegative matrices, we have $p_{i,t}q_{t,j}=0$ for all $1\le t\le n$, which leads to

$$\theta(p_{i,t}q_{t,j})=0.$$

From the definition of Heaviside step function θ, i.e. formula (4), we have

$$\theta(p_{i,t})\theta(q_{t,j})=0.$$

Therefore, we have

$$\bigvee_{t=1}^{n}\theta(p_{i,t})\theta(q_{t,j})=\theta(g_{i,j})=0.$$

On the other hand, in case of $g_{i,j}=\sum_{t=1}^{k}p_{i,t}q_{t,j}>0$, because P and Q are nonnegative matrices, there exists $p_{i,t}q_{t,j}>0$ for $1\le t\le n$, which leads to

$$\theta(p_{i,t}q_{t,j})=1.$$

From definition of θ, we have $\theta(p_{i,t})\theta(q_{t,j})=1$.

Therefore, we have

$$\bigvee_{t=1}^{n}\theta(p_{i,t})\theta(q_{t,j})=1$$

Considering the above two cases, we have

$$\theta(PQ)=\theta(P)\odot\theta(Q)$$

hold for $P=(p_{i,j})^{m\times n}\in R_+^{m\times n}$ and $Q=(q_{i,j})^{m\times n}\in R_+^{m\times n}$. QED.

For given matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, we generate $\boldsymbol{G}=\boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}}$, $\boldsymbol{G}_i=\boldsymbol{X}_i(\boldsymbol{Y}_i)^{\mathrm{T}}$ by taking the general matrix product. According to the above properties, $\boldsymbol{H}=\theta(\boldsymbol{G})$ and $\boldsymbol{H}_i=\theta(\boldsymbol{G}_i)$ hold. We update $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ after updating $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ in the previous round. Instead of computing the whole matrix $\boldsymbol{H}_j$, we calculate

$$\boldsymbol{G}^j=\boldsymbol{G}^i+\boldsymbol{x}_i\boldsymbol{y}_i^T-\boldsymbol{x}_j\boldsymbol{y}_j^T \quad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$$\boldsymbol{H}^j=\theta(\boldsymbol{G}^j) \quad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

In equation (5), $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are the updated values and $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ are the original values. Equation (5) and (6) enable us to compute matrix $\boldsymbol{H}^j$ quickly. The alternating approach algorithm is as follows in Figure 2.

For AIA algorithms, we have the following convergence theorem 2.

Theorem 2.

1) For AIA deterministic algorithms, suppose H be the original matrix and $\boldsymbol{H}_{new}$ be the updated matrix, then

$$\|\overline{\boldsymbol{R}}-\mathrm{X}_{\mathrm{new}}\odot(\boldsymbol{Y}_{\mathrm{new}})^{\mathrm{T}}\|_{\mathrm{F}}\le\|\overline{\boldsymbol{R}}-\mathrm{H}\|_{\mathrm{F}}.$$

2) By the same token, for AIA stochastic algorithm, it converges each step at the final equilibrium distribution

$$\mathrm{P(x,y)=(1/Z)exp(}-\|\overline{\boldsymbol{R}}-\mathrm{X}\odot\mathrm{Y}^T\|_F/\tau),$$

wherein Z is a normalized constant for probability distribution.

Proof: Define local energy function as Eloc (x, y) $=\|\overline{\boldsymbol{R}}-\mathrm{X}\odot\mathrm{Y}^{\mathrm{T}}\|_{\mathrm{F}}$.
1) For AIA deterministic updating algorithm, if we see its 3rd step as Hopfield neural network bidirectional associative computing

updating algorithm [29], it holds according to Hopfield convergence theorem of bidirectional associative memory.

2) For the AIA stochastic algorithm, if we see its 3rd step as a simulated annealing algorithm [29], according to the conclusions of simulated annealing, the final equilibrium distribution $P(x,y)=(1/Z)\exp(-\|\overline{\boldsymbol{R}} - X \odot Y^T\|_F / \tau)$ can be obtained, wherein Z is a probability distribution of a normalization constant , τ is called Boltzmann machine annealing temperature, which should be gradually reduced during the iteration. QED.

```
Step 1. Input the response matrix $\overline{\boldsymbol{R}} \in \{0,1\}^{m \times n}$ and
    $\boldsymbol{X} \in \{0,1\}^{m \times k}$, $\boldsymbol{Y} \in \{0,1\}^{n \times k}$
Step 2. Compute $\boldsymbol{G} = \boldsymbol{X}\boldsymbol{Y}^T$.
Step 3. Randomly select (or deterministically select) $l$,
    $1 \le l \le K$, and update the $l$ columns of $\boldsymbol{X}$ and $\boldsymbol{Y}$.
  3.1 Compute $\boldsymbol{G}_l = \boldsymbol{X}(:,l)\boldsymbol{Y}(:,l)^T$, $\boldsymbol{G}^l = \boldsymbol{G} - \boldsymbol{G}_l$, $\boldsymbol{H} = \theta(\boldsymbol{G}^l)$,
      $U = (r_{ij}\overline{h}_{ij})_{m \times n}$, $V = (\overline{r}_{ij}\overline{h}_{ij})_{m \times n}$;
  3.2 Set $\boldsymbol{y} = \boldsymbol{Y}(:,l)$;
  3.3 Calculate $\boldsymbol{\alpha} = \boldsymbol{Uy}$, $\boldsymbol{\beta} = \boldsymbol{Vy}$, and update $\boldsymbol{x}$ based on one of the
      following methods:
      1) (update method I) $\boldsymbol{x} = \theta(\rho \cdot \boldsymbol{\beta} - \boldsymbol{\alpha})$, where $0 < \rho \le 1$ is a given
         parameter;
      2) (update method II) $\mathrm{Prob}(\boldsymbol{x}(i)=1 | \alpha_i, \beta_i) = \frac{1}{1+e^{-\tau(\beta_i - \alpha_i)}}$,
         where $\tau$ is a given parameter;
  3.4 Compute $\gamma = \boldsymbol{U}^T\boldsymbol{x}$, $\delta = \boldsymbol{V}^T\boldsymbol{x}$, and update $\boldsymbol{y}$ based on one of
      the following methods:
      1) $y = \theta(\rho\delta - \gamma)$, where $0 < \rho \le 1$ is a given parameter;
      2) $\Pr ob(y(j)=1) | \gamma_j, \delta_j) = \frac{1}{1+e^{-\tau(\delta_j - \gamma_j)}}$, where $\tau$ is a
         given parameter;
  3.5 Repeat steps 3.3 and 3.4, till $\boldsymbol{x}$ and $\boldsymbol{y}$ converge (update
      method I) or the distributions of x and y become stable
      (update method II).
  3.6 Update $\boldsymbol{X}$: $\boldsymbol{X}(:,l) = \boldsymbol{x}$;
      Update $\boldsymbol{Y}$: $\boldsymbol{Y}(:,l) = \boldsymbol{y}$;
      Update $\boldsymbol{G}$: $\boldsymbol{G} = \boldsymbol{G}^l + \boldsymbol{xy}^T$;
Step 4. Repeat step 3 till the change of values less than a
    given threshold.
```

Figure 2. Alternating Iteration Algorithms (AIA)

According to the conclusions of simulated annealing the AIA stochastic algorithm can obtain global optimal Q-matrix and A matrix when attributes fixed, as long as we assure the decreasing rate of annealing temperature being slow enough.

4. EXPERIMENTAL RESULTS

Q-matrix Reproduction from Real Response Data

To show how the proposed methods work, real responses to an actual test we developed is used. The test is a fraction diagnostic test comprised of 35 items in terms of the conjunctive assumption. Eight attributes have been specified and developed by content experts as essential skill required in solving fraction problem according to the "Japanese government curriculum guidelines for teaching" (for details, see [22]). We administered the test to 144 sixth grade students in an elementary school in Tokyo and the real response data of 144 students to 35 items are used as ***R*** matrix.

We use the proposed algorithms to approach this response matrix ***R*** and reproduce the ***Q***-matrix and knowledge states matrix ***A***. Specifically we use the PAA algorithm illustrated in Figure 1 to optimize latent matrices ***Q*** (denoted as ***X***) and ***A*** (denoted as ***Y*** for complementary ***A***) first. By setting the results as initial values we use the other AIA algorithm illustrated in Figure 2 then to iteratively optimize the perturbation and get the final estimated matrices ***Q***, ***A*** and approaching matrix ***H*** of the complementary matrix ***R***. As shown in the above section, there are two update methods in each algorithm but here we only show the results of update method I.

It is indicated that calculations converge within 15-20 iterations for all ρ parameter ranging from 0 to 1. An example for ρ=0.5 is shown in Figure 3. The vertical axis represents the coverage rate indicating how well the generated approaching matrix has reproduced the original response ***R*** matrix. We can see that the initial value estimated by the PAA algorithm covers a little more than 85% of the original ***R*** and by the AIA algorithm the coverage rate increases rapidly at first 5-6 iterations moving towards a stable value (Figure 3). The convergence plots for all ρ parameter have the same trends, although the coverage rate for each one is a slight different.

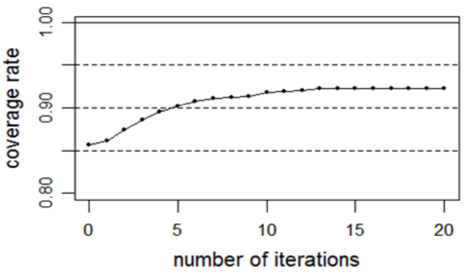


Figure 3. Converging of coverage rate (ρ=0.5)

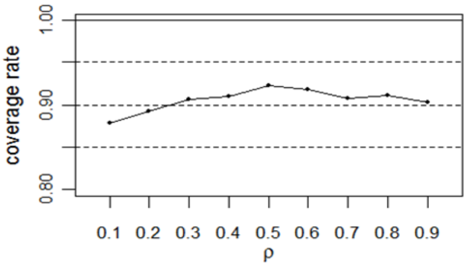


Figure 4. Coverage rates along with different ρ

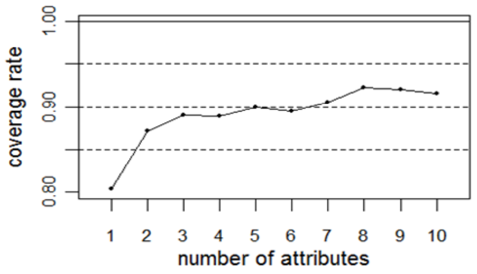


Figure 5. Coverage rates with number of attributes (ρ=0.5)

The coverage rate at 20 iterations for each ρ is given in Figure 4. We can see that despite of different ρ the coverage rates are as high as around 90%, which indicates our algorithms are valid and give good optimization and reproduction from the original response data.

Figure 5 shows how well the original response ***R*** matrix is reproduced by different number of attributes. The coverage rate, starting from 80% by single attribute, has been increasing as the number of attributes increases reaching the peak at eight attributes. It is interesting to notice that "eight" are the number the content experts specified and considered to be appropriate for the particular test in our previous study [22].

5. CONCLUSION AND FUTURE WORK

We used Boolean Matrix Factorization (BMF) to express conjunctive models in CDMs and proposed recursive algorithms for updating the matrix in latent attributes space (its rank equals one) at each step in order to get optimal solutions. We also used a Heaviside step function to decompose the recursive computing process into normal non-negative matrices and get results by mapping them back into a Boolean matrix, which makes our approximation algorithms faster. Two different algorithms were presented: a deterministic heuristic algorithm and a stochastic algorithm. We presented examples demonstrating applications of one of these algorithms based on an actual test dataset. The results indicate that Q-matrix learned can reproduce item response data with more than 90% coverage rate, which suggests that our algorithms are valid.

As the next step we will compare the Q-matrix learned from the data by the proposed methods with the one created by experts in our previous research [22]. We will also introduce statistical parameters, such as "guessing" and "slip", to make our methods more applicable for real data.

6. REFERENCES

[1] Barnes, T. 2010. Novel derivation and application of skill matrices: The Q-matrix method. *Handbook on Educational Data Mining*, 159-172.

[2] Belohlavek, R., Vychodi, V. 2010. Discovery of optimal factors in binary data via a novel method of matrix decomposition, *Journal of Computer and System Sciences* 76:3-20

[3] Carpineto, C., Romano, G. 2004. *Concept Data Analysis. Theory and Applications*. Wiley.

[4] Chiu, C.-Y., Douglas, J. A., and Li, X. 2009. Cluster analysis for cognitive diagnosis: Theory and applications. *Psychometrika*, 74(4), 633-665.

[5] De La Torre, J. 2008. An empirically based method of q-matrix validation for the DINA model: Development and applications. *Journal of educational measurement*, 45(4), 343-362.

[6] DeCarlo, L.T. 2011. On the analysis of fraction subtraction data: the DINA model, classification, latent class sizes, and the Q-matrix. Applied Psychological Measurement 35, 8-26.

[7] Desmarais, M.C. 2011. Mapping question items to skills with non-negative matrix factorization. *ACM KDD-Explorations*, 13(2), 30-36.

[8] Desmarais, M.C., Beheshti, B., Naceur, R. Item to skills mapping: Deriving a conjunctive Q-matrix from data. *Proceeding of ITS'12 Proceedings of the 11th international conference on Intelligent Tutoring Systems*, 454-463.

[9] Desmarais, M.C.; Naceur, R. 2013. A Matrix Factorization Method for Mapping Items to Skills and for Enhancing Expert-Based Q-matrices. *Artificial Intelligence in Education, Lecture Notes in Computer Science,* 7926, 441-450.

[10] Desmarais, M.C. 2011. Mapping Question Items to Skills with Non-negative Matrix Factorization. *ACM SIGKDD Explorations Newsletter archive*, 13(2), 30-36.

[11] DiBello, L. V., Roussos, L. A., and Stout, W. 2007. Review of cognitively diagnostic assessment and a summary of psychometric models. *Handbook of statistics*. 26(31), 1-52.

[12] Hartz, S. M. 2002. *A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality*. PhD thesis, University of Illinois at Urbana-Champaign.

[13] Junker, B. W. and Sijtsma, K. 2001. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(1), 258-273.

[14] Koedinger, K.R., McLaughlin, E.A., Stamper, J.C. 2012. Automated student model improvement. In *Proceedings of the 5th International Conference on Educational Data Mining*.

[15] Leighton, J. P., Gierl, M. J., and Hunka, S. M. 2004. The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach. *Journal of Educational Measurement*, 41(3), 205-237.

[16] Li, N., Cohen, W.W., Matsuda, N., Koedinger, K.R. 2011. A machine learning approach for automatic student model discovery. In *Proceedings of the 4th International Conference on Educational Data Mining*. 31-40.

[17] Liu, J., Xu, G., Ying, Z. 2012. Data-driven learning of q-matrix. *Applied Psychological Measurement*. 36(7), 548-564.

[18] Miettinen, P., Mielik?inen, T., Gionis, A., Das, G., Mannila, H. 2006. The discrete basis problem, in: Proc. PKDD 2006, in: Lecture Notes in Artificial Intelligence, 4213, 335-346.

[19] Miettinen, P., Mielikainen, T., Gionis, A., Das, G., Mannila, H. 2008. The Discrete Basis Problem. *IEEE Transactions on Knowledge and Data Engineering*, 20(10), 1348-1362.

[20] Neruda, R., Snasel,V., Platos,J., Kromer,P., Husek, D. 2008. Implementing Boolean Matrix Factorization. *ICANN 2008, Part I, LNCS* 5163, 543-552.

[21] Rupp, A. A. and Templin, J. 2008. The effects of Q-matrix misspecification on parameter estimates and classification accuracy in the DINA model. *Educational and Psychological Measurement*, 68(6):78-96.

[22] Takahashi, T., Sun, Y., Kakinuma, S. 2011. Development of attributes of cognitive diagnostic test in fraction calculations, in *Proceeding of 2011 conference of the Japan Association for Research on Testing*, 220-221.

[23] Tatsuoka, K. K. 1983. Rule space: an approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4), 345-354.

[24] Tatsuoka, K. K. 1985. A probabilistic model for diagnosing misconceptions in the pattern classification approach. *Journal of Educational Statistics*, 10(1), 55-73.

[25] Vaidya, J. 2012. Boolean Matrix Decomposition Problem: Theory, Variations and Applications to Data Engineering. *IEEE 28th International Conference on Data Engineering*

[26] von Davier, M. 2005. A general diagnostic model applied to language testing data. *ETS Research Report RR-05-16*. Princeton: Educational Testing Service

[27] Wille, R. 1982. Restructuring lattice theory: An approach based on hierarchies of concepts, in: I. Rival (Ed.), *Ordered Sets*, Reidel, Dordrecht/Boston, 445-470.

[28] Xiang, R. 2013. *Nonlinear Penalized Estimation of True Q-Matrix in Cognitive Diagnostic Models*. PhD thesis, Columbia University.

[29] Haykin, S. 2009. Neural networks and learning machines, 3rd ed., Prentice Hall, NJ.

[30] Ye, S., Sun, Y., Inoue, S., Sun, Y. 2014. Matrix extending methods for Q-matrix learning (work paper in progress).

Application of Time Decay Functions and the Elo System in Student Modeling

Radek Pelánek
Masaryk University Brno
pelanek@fi.muni.cz

ABSTRACT

One of the key aspects of educational data mining is estimation of student skills. This estimation is complicated by the fact that students skills change during the use of an educational system. In this work we study two flexible approaches to skill estimation: time decay functions and the Elo rating system. Results of experiments in several different settings show that these simple approaches provide good and consistent performance. We argue that since these approaches have several pragmatical advantages (flexibility, speed, ease of application) they should be considered in educational data mining at least as a baseline approach.

1. INTRODUCTION

One of the goals of educational data mining is to estimate skill (knowledge) of students. The problem of skill estimation is the following: we have sequential data about student performance (e.g., answers to exercises, timing) and we want to estimate a latent student skill. The quality of the skill estimate can be evaluated by its ability to predict future performance. Once we have a reliable skill estimate, it can be used in many ways: for guiding adaptive behaviour in intelligent tutoring systems, for computerized adaptive practice, or for providing feedback to students (e.g., in skillometers, open learner models).

In skill estimation, there are two main approaches to dealing with the sequentiality of the data. One approach is simply to ignore the ordering of the data, i.e., to make a simplifying assumption that students do not learn and the skill is a constant. This approach is usually used with "coarse-grained" skills (like "fractions" or even "arithmetic"), where the rate of skill change is slow and thus the assumption of constancy is reasonable. A typical example of this approach is item response theory [3], which is used mainly for adaptive testing. In this case the assumption is justified since we do not expect students to learn during test. But even some models used in adaptive learning systems do not consider the order of data and treat all data points in same way (e.g., performance factor analysis [19] or a model of problem solving times [10]).

The second main approach is to make a fixed assumption about learning and "hard code" it into the model. A typical example of this approach is Bayesian knowledge tracing (BKT) [2, 22], which models knowledge as a binary variable (known/unknown) with a given probability of switching from unknown to known. Another approach of this type are models based on learning curves [16], which typically assume a logarithmic increase in skill with respect to number of attempts (e.g., a model of problem solving times with learning [9]). These types of models are used mainly with "fine-grained" skills (e.g., a specific operations with fractions). For their application it is necessary that the skills are correctly identified, so that the model assumptions hold [2].

In this work we study educational application of two interrelated techniques – time decay functions and the Elo system. These techniques are between the two above described approaches. They do take the sequentiality of the data into account, but do not make fixed assumptions about learning. Both techniques are rather flexible and thus are applicable to wide range of skill granularity.

The first technique is based on time decay functions. Since students skills and knowledge changes over time, the older data are less relevant for the estimation than the recent data. Thus it makes sense to use some kind of data discounting – in analysis of sequential data this can be done using weighting by a time decay function [12, 6]. Only little research in student modeling has so far studied data discounting or some similar temporal dynamics, e.g., using less data in BKT [17], data aging [29], or effect of real time (not just ordering) in BKT [20].

The second technique is the Elo system [4], which was originally devised for chess rating (estimating players skills based on results of matches), but has recently been used also for student modeling [13, 27]. In context of skill estimation we interpret an attempt of a student to answer an item as a "match" between the student and the item. This approach updates a skill estimate based on the result of a last match in such a way that implicitly leads to a discounting of past attempts.

The goal of this work is to explore applicability of time decay functions and Elo system in educational data mining.

More specifically to study the following questions: What is a good time decay function in the context of educational data mining? How sensitive are results with respect to parameters of time decay function and Elo rating? How do these approaches compare to other student modeling techniques? To answer these questions we apply the techniques different contexts and we use for evaluation several different datasets. The obtained results are quite stable and favourable for these approaches, and thus we also discuss their possible application in intelligent tutoring systems.

2. MODELS FOR SKILL ESTIMATION

We study the skill estimation in two context: modeling of correctness of student answers (the only measure of performance is correctness of the answer, possibly also the number of hints used) and modeling of problem solving times (the only measure of performance is a time to solve a problem).

2.1 Overview of Relevant Models

In item response theory the main model is the 3 parameter logistic model, which assumes a constant student skill θ and three item parameters: b is the basic difficulty of the item, a is the discrimination factor, and c is the pseudo-guessing parameter. The model assumes that the probability of a correct answer is given by a (scaled) logistic function:

$$P_{a,b,c,\theta} = c + (1-c)\frac{e^{a(\theta-b)}}{1+e^{a(\theta-b)}}$$

A specific case of this model is a 1 parameter model, which is obtained by setting $c = 0, a = 1$; this model is also called the Rasch model.

A model of problem solving times [10] uses parameters with analogous meaning and assumes a log-normal distribution of problem solving times:

$$f_{a,b,c,\theta}(\ln t) = \mathcal{N}(a\theta + b, c)(\ln t) = \frac{1}{\sqrt{2\pi}c} e^{-\frac{(\ln t-(a\theta+b))^2}{2c^2}}$$

Bayesian knowledge tracing [2, 22] models a changing skill. It is a hidden markov model where skill is the binary latent variable (either learned or unlearned). The model has 4 parameters[1]: probability that the skill is initially learned, probability of learning a skill in one step, probability of incorrect answer when the skill is learned (slip), and probability of correct answer when the skill is unlearned (guess). The skill estimated is updated using a Bayes rule based on the observed answers.

2.2 Time Decay Functions

Time decay function are used in the study of concept drift [6, 12, 21]. Concept drift is relevant example for modeling the change of user preferences in recommender systems, where the inclusion of temporal dynamics into models can improve their performance [15]. A different area that uses temporal discounting is economics and study of decision making [5], where temporal discounting and time decay functions are studied mainly with respect to decisions about future. All these areas can provide useful inspiration for student modeling (e.g., the choice of the time decay function), but are not directly applicable.

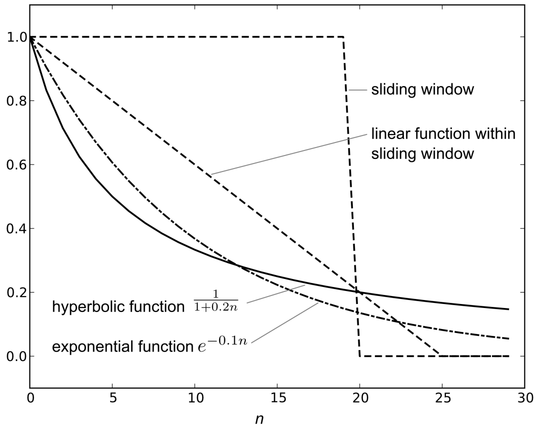


Figure 1: Examples of time decay functions.

A time decay function assigns a weight to a data point (student performance) that happened in the past. As a measure of "time" we use a number of attempts (denoted n). Other possibilities are to use a "real time" (seconds from the attempt) or "semi-real time", which counts the number of attempts but takes into account big pauses (e.g., larger step for a day switch). Figure 1 shows several natural candidates for time decay functions, which we have evaluated in our experiments.

Let us apply time decay functions to student modeling. In the case of modeling correctness of answers, we have data of the following type: student s gave to an item i an answer with correctness c_{si} (usually a binary variable, in case of a "partial credit model" [25] it can also have a continuous value between 0 and 1). The skill of a student s is estimated as a weighted average of c_{sk} with weights given by the time decay function, i.e., $\theta_s = \sum f(k)c_{si_k} / \sum f(k)$, where i_k is the item solved by the student k steps into the past. This skill estimate is in the range $[0, 1]$ and can be directly used to predict future performance.

We also study modeling of problem solving times. In accordance with previous research [10, 23], we work with the logarithm of time, since raw times are usually log-normally distributed. Now we assume data of the type: student s solves a problem p in a logarithm of time t_{sp}. We denote θ_{sp} a "local skill estimate" on a particular problem: $\theta_{sp} = m_p - t_{sp}$, where m_p is a mean time to solve the problem p. A current skill of a student s is estimated as a weighted average of these local estimates with weights given by the time decay function: $\theta_s = \sum f(k)\theta_{sp_k} / \sum f(k)$, where p_k is the problem solved in k steps into the past. The skill estimate can be used to predict performance on an unsolved problem p as follows: $\hat{t_{sp}} = m_p - \theta_s$. Note that with a constant weight function this approach is equivalent to the baseline personalized predictor used in [9, 10].

[1] BKT can also include forgetting. The described version corresponds to the variant of BKT that is most often used in research papers.

Compared to more complex students models (BKT, model of problem solving times) the outlined approaches to estimating student skill are quite simple. The advantage of this simplicity (apart of simplicity of implementation and application) is that they make minimal assumptions about the behaviour of students, e.g., this approach can naturally accommodate forgetting (as opposed to BKT, where the inclusion of forgetting means an additional parameter) and also such effects as a change of working environment (e.g., switching from a computer with mouse to notebook with touchpad can increase problem solving times for interactive problems).

2.3 The Elo System

The basic principle of the Elo system is the following. For each player i we have an estimate θ_i of his skill, based on the result R ($0 = $ loss, $1 = $ win) of a match with another player j the skill estimate is update as follows:

$$\theta_i := \theta_i + K(R - P(R = 1))$$

where $P(R = 1)$ is the expected probability of winning given by the logistic function with respect to the difference in estimated skills, i.e., $P(R = 1) = 1/(1 + e^{-(\theta_i - \theta_j)})$, and K is a constant specifying sensitivity of the estimate to the last attempt.

There exists several extension to the Elo system, the most well-known are Glicko [7], which explicitly models uncertainty in skill estimates, and Trueskill [8], which can be used also for team competitions. The Elo system has also been used previously in modeling of correctness of student answers by interpreting student solution attempt as a match between a student and an item [13, 27].

In the case of problem solving times we can apply the method as follows: for each student we have an skill estimate θ_s, for each problem we have a difficulty estimate d_p. When the student s solves the problem p in the logarithm of time t_{sp} we update these estimates as follows:

$$\theta_s := \theta_s + K(E(t|s,p) - t_{sp})$$

$$d_p := \theta_p + K(t_{sp} - E(t|s,p))$$

where $E(t|s,p)$ is an expected solving time for a student s and problem p, which is given as $E(t|s,p) = d_p - \theta_s$.

The value of the constant K determines the behaviour of the system – if K is small, the estimation converges too slowly, if K is large, the estimation is unstable (it gives too large weight to last few attempts). An intuitive improvement, which is used in most Elo extensions, is to use an "uncertainty function" instead of a constant K. Previous work on using the Elo system for student modeling [13, 27] used ad hoc uncertainty functions selected for particular application.

An important difference of application of the Elo systems in its typical domains (chess and other competitions) and in student modeling, is the asymmetry in student modeling between students and problems. Particularly we typically have much more students than problems and consequently more data about particular problems than students. Thus it makes sense to use different uncertainty functions for students and problems.

2.4 Relation between Time Decay and the Elo System

Both described approaches are closely related – they can both capture changing skill and do not make any specific assumptions about the nature of the change, they just give more weight to recent attempts. The close relation between these two approaches is apparent particularly in modeling of problem solving times. Using the previously described notation of a local performance $\theta_{sp} = d_p - t_{sp}$, the update rule of the Elo system can be rewritten as follows:

$$\begin{aligned} \theta_s := \quad & \theta_s + K(E(t|s,p) - t_{sp}) = \theta_s + K(d_p - \theta_s - t_{sp}) = \\ & \theta_s + K(\theta_{sp} - \theta_s) = (1 - K)\theta_s + K\theta_{sp} \end{aligned}$$

Now if we consider a sequence of n solved problems and assume an initial skill estimate 0, the final skill estimate is given by:

$$\theta_s = K \sum_{i=1}^{n} (1 - K)^{n-i} \theta_{sp_i}$$

The resulting expression is very similar to the estimation with exponential decay function, the main difference is the use of m_p (mean problem solving time) versus d_p (difficulty parameter estimated by the Elo system), but since problems are usually solved by large numbers of students and difficulty parameter is quite easy to estimate [9], this difference is not practically important.

The relation with time decay function is not so straightforward for applications of the Elo system to correctness data, which uses the logistic function, and for extension of the Elo system with uncertainty function. Nevertheless, some sort of temporal dynamics is inherently included in all variants of the Elo system. Extension usually correspond to the use of a steep decay function during first few student attempts and flatter decay function later, when the skill estimate is more stable.

3. EVALUATION

We present evaluation of time decay functions and the Elo system on several different datasets. Rather than performing one exhaustive experiment with one dataset, we performed several basic experiments in different settings (different types of datasets, simulated data).

3.1 Simulated Data

Using simulated data we explore how well can the Elo system and estimation using time decay functions approximate previously studied models (mentioned in Section 2.1). We use the following type of experiment: we generate data using one of the standard models and then try to fit the data using one of the studied approaches (the Elo system, time decay functions).

The first experiment concerns comparison of the Rasch model (one parameter logistic model) and the Elo system. These two approaches are very similar, since both assume one student parameter (skill), one item parameter (difficulty), and the same functional form of the probability of correct answer (logistic function with respect to the difference between skill and difficulty). The differences between these approaches are in the assumption about constancy of parameters and in parameter estimation methods. The Rasch model assumes

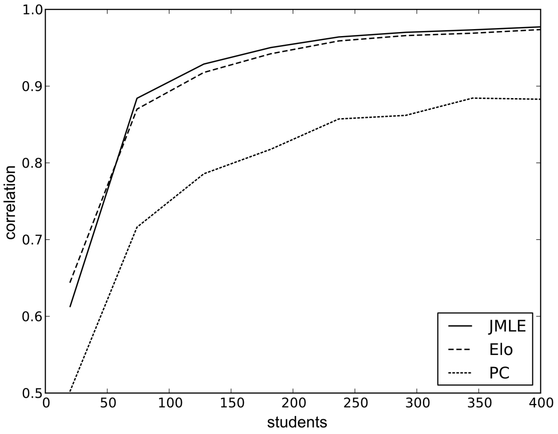


Figure 2: Correlation between generated and estimated difficulty parameters for different number of students. JMLE = Joint maximum likelihood estimation, Elo = Elo system, PC = proportion correct.

that the parameters are constant, specifically that the skill is constant (i.e., no learning). The standard method for estimating parameters of the Rasch model is the iterative procedure joint maximum likelihood estimation (JMLE) [3]. The Elo system does not make any specific assumptions about the constancy or change of the skill or difficulty and tracks these parameter in more heuristic fashion.

We performed the following experiment. The simulated data are generated using the Rasch model with skills and difficulties generated from standard normal distribution. The data are then fitted using JMLE and the Elo system and we compare the fitted values of parameters with the generated values. As a metric of fit we use the correlation coefficient. The Elo system with constant K leads to significantly worse results than JMLE, but if we use a suitable uncertainty function, the two estimation procedures give very similar results (correlation mostly above 0.99). A suitable uncertainty function is for example the hyperbolic function $\frac{a}{1+bx}$, with parameters $a = 4, b = 0.5$. Suitable parameters can be easily found by grid search, the performance of the system is quite stable and the precise choice of parameter values is not fundamental to the presented results.

In the case that we have complete data about answers or data are missing at random, even a simple “proportion correct” statistics gives good prediction of item difficulty. However, in real systems data are not missing at random, particularly in adaptive systems more difficult items are solved only by students with above average skill.

Figure 2 shows the results for such scenario. Data are generated using the Rasch models, portion of the data is missing, the availability of answers is correlated with student skill and item difficulty. The data are generated for 100 items and different number of students, the results are averaged over 50 runs. In this scenario, results for “proportion correct” are significantly worse than for the other two methods, detailed analysis shows that the estimates are wrong particularly in the middle of the difficulty range. Results for JMLE and Elo are nearly identical, the graph demonstrates that the difference in the amount of available data is more important than the difference between the estimation procedure used. If we have enough data, the JMLE is slightly better than Elo, but for small amount of data Elo is even better than JMLE. Note that this scenario is optimistic for the JMLE, since the simulated data adhere to the constancy of skill assumption, whereas any real data will contain at least some variability. We have performed similar experiments in the case of problem solving times. The results are similar, again we get similar performance and a suitable uncertainty function is the hyperbolic function.

Another experiment concerns comparison of the Bayesian knowledge tracing and skill estimation using time decay functions. Similarly to the previous experiment, we simulated data from a BKT model with fixed parameters. Then we use the BKT model and the time decay approach to make predictions and compare them using the AUC metric (results for RMSE metric are similar). For the predictions we use the BKT model with the optimal parameters, i.e., those used to generate the data. This is again overly optimistic case for BKT, as we assume that the data fully correspond to the assumptions of the model and that we know the correct parameter values. To make the comparison fairer, the estimation using time decay has at least the information about the initial probability of learned skill. Even in this setting, time decay approach gets close to BKT. For BKT parameters 0.5, 0.14, 0.09, 0.14 (taken from [20] as average BKT parameter values from the ASSISTments system), the AUC values are 0.822, 0.815. The time decay function used is the exponential function $e^{-0.3n}$; similarly to the previous experiment the choice of optimal value of the parameter can be done easily using an exhaustive search.

3.2 Real Data

At first we describe experiments with models of problem solving times. For this evaluation we use data from the Problem Solving Tutor [11], which is an open web portal with logic puzzles and problems from mathematics and computer science. For comparing different models we use root mean square error (RMSE) metric.

The results show that time decay functions can bring improvement of predictions. Figure 3 shows results for the exponential decay function. As the graph shows, the optimal parameter k for the exponential function e^{-kn} is around 0.1. Hyperbolic function $1/(1 + kn)$ achieves similar results as the exponential function, with optimal values of the parameter k in the interval 0.2 to 1.2. The sliding window and linear function within sliding window achieve significantly worse results.

Different problem types behave similarly with respect to which time decay functions and which parameter values bring the best improvement. They, however, differ in the amount of improvement. For some problems the improvement is only minor – these are for example Tilt maze and Region division, which are rather simple puzzles where we do not expect significant learning or other temporal effects affecting performance. Hence it is not very useful to discount data about past attempts. On the other end are problems like

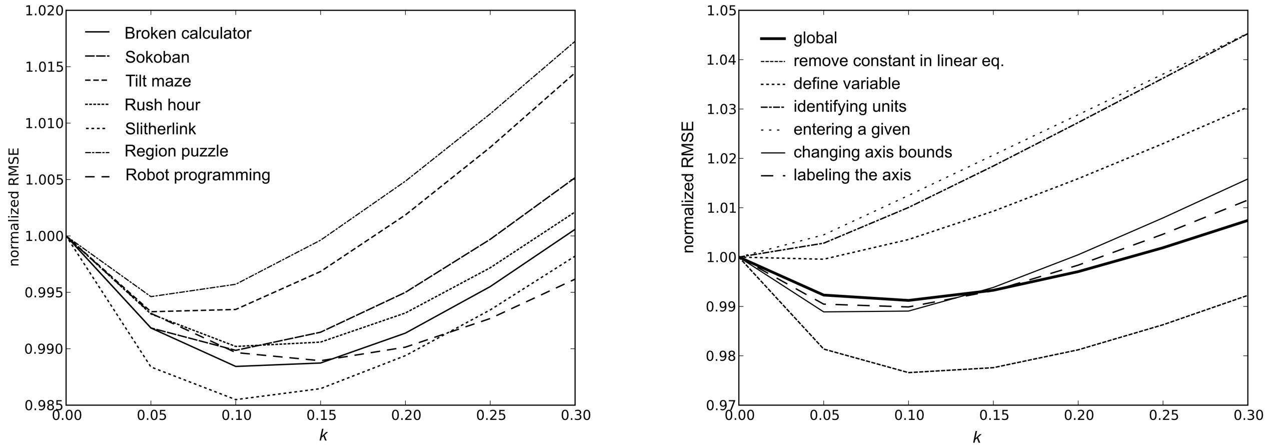


Figure 3: Results for exponential time decay function e^{-kn} for varying k; the graph shows normalized RMSE (with respect to constant time decay function). Left: Data from Problem Solving Tutor (problem solving times), Right: Algebra data set (correctness data).

Slitherlink (more advanced logic puzzle) or Broken Calculator (practice of calculations), where the improvement is larger and the best results are obtained by steeper decay functions. For these problems learning is more significant and thus it is sensible to take into account particularly last few attempts.

In previous work [9] we have proposed a model of problem solving times that makes a fixed assumption about learning, particularly the assumption of logarithmic improvement with respect to the number of attempts (in agreement with the research on learning curves). For the used dataset, this model does not bring any systematic improvement in predictions, whereas time decay functions do improve predictions (see [14] for more detailed analysis of this comparison). Thus it seems that for the used dataset there are temporal effects in the performance of students that do not easily conform to the assumptions of learning curves – the dataset contains nonstandard educational problems and logic puzzles and some of the problems require "insight", not just application of some fixed set of principles.

So far we have used the time decay functions with respect to the number of attempts. Another option is to use the time decay function with respect to real time or to take at least some aspects of the real time into account, e.g., to consider large pause between attempts (similarly to the approach used in [20]). We have performed experiments with this extension, but the results stay very similar or bring only small improvement (using linear combination of number of attempts and the logarithm of passed time [14]).

In Figure 3 we have evaluated the parameter of time decay function with respect to the problem type. We can do similar analysis with respect to students. If the student's performance is improving fast, then the optimal time decay function for him is steep, i.e., there is some relation between learning and optimal choice of the time decay function. However, this relation is not straightforward, as steep decay function can also mean high autocorrelation without learning, e.g., when the student accesses the educational system from different environments (mouse vs touchpad) or at different conditions (morning vs night).

The results for the Elo system over this dataset are similar and we summarise them only briefly. Even the basic Elo system achieves similar predictions as the model of problem solving times from [10]. The extension of the Elo system that uses the uncertainty function with parameters determined from experiments with simulated data can achieve improvement over the previously published model by 1 to 3 percent in RMSE [24].

For experiments with student models that predict correctness of answers we used an Algebra I dataset from KDD Cup 2010 (binary correctness) and ASSISTment dataset with partial credit data [25] (correctness is a number between 0 and 1 depending on the number of hints used). In both of these datasets each item has a knowledge component assigned and we compute skills for these specified knowledge components.

Although this is different setting and completely different datasets from the previous experiments, the results are very similar (Figure 3). For the choice of a time decay function we have analogical results: exponential and hyperbolic functions work best, sliding window (in both versions) is significantly worse. The choice of optimal parameters for time decay functions is again similar (usually around 0.1 for exponential function) and again we observe differences between different skills (knowledge components). For generic skills (like "Identifying units", "Entering a given" in Algebra), time decay does not bring an improvement. For specific skills (like removing constant in linear equation), the optimal time decay function is steep and improves performance, i.e., for these skills there is significant learning and hence it pays to give large weight to recent attempts.

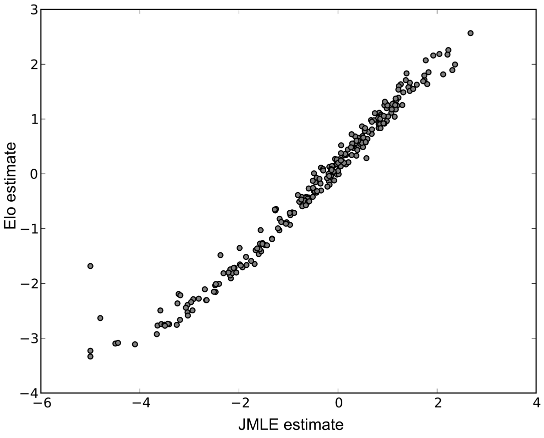


Figure 4: Comparison of JMLE and Elo estimates of difficulty of geography items (country names).

For the Elo system in the context of correctness of answers, we have applied and evaluated the system in an educational application for learning geography (names of countries) – `slepemapy.cz`. We use the Elo system to estimate the prior geography knowledge of students and difficulty of countries. Similarly to the above reported experiments with simulated data, the Elo model (with uncertainty function) achieves very similar results as the joint maximum likelihood estimation for the Rasch model (see Figure 4). The Elo system is much faster and more suitable for online application than the iterative JMLE procedure. To estimate the probability of correct answer after a sequence of attempts at a given country we use a model that combines aspects of performance factor analysis [19] and the Elo system. This combined model achieves better results than both standard performance factor analysis and Bayesian knowledge tracing. More details about this application and evaluation are given in [18].

4. DISCUSSION

We have performed experiments in different settings and with different datasets. Basic results are quite consistent. The Elo system and estimation using time decay functions are simple and flexible approaches, which can match more specific models (Rasch model, Bayesian knowledge tracing) even if the data are generated exactly according to the assumptions of the more specific model. For the choice of time decay function, it seems that for student modeling it is most useful to use either exponential or hyperbolic function (our experiments do not show systematic significant difference between these two). Sliding window and linear function within sliding window lead to worse results.

The choice of specific parameters is also quite consistent, e.g., for the exponential time decay function e^{-kt} the best k is usually around 0.1. The differences between problems of the optimal value of the parameter k (i.e., of the shape of time decay function) are related to the speed of learning for a particular problem type (knowledge component). This relation is however not straightforward, because the time decay approach captures not just learning, but also other temporal effects (e.g., autocorrelation of result due to the use of the system from different working environments). For the uncertainty function of the Elo system a good candidate is a hyperbolic function $\frac{a}{1+bx}$, the specific parameters a, b differ according to the exact application, but the values can be easily found using a grid search and the performance of the system is only mildly sensitive to the exact values.

The advantages of both studied techniques are their flexibility, small number of parameters, and easiness of application. Flexibility is due to the weak assumptions about student behavior and allows for application in wide variety of contexts – this was demonstrated by wide range of data used in our evaluation (e.g., logic puzzles, math problems, knowledge of country names). Small number of parameters reduces the chance of overfitting and leads to stable results. Both techniques are very easy to implement and have low computational demands – predictions are easy to compute, the Elo system and exponential time decay function can be even used in online fashion without storing data about individual student attempts.

Both studied techniques are quite general. Since they do not make any specific assumptions, they should not be expected to bring an optimal performance results for a particular situation. But as we show, they can be easily applied in wide range of situations and provide reasonable performance. Moreover, small improvements in performance (which can be brought by more specific models) are often not practically important for applications of skill estimates. Even if more specific models are available, these simple approaches can be used to get quick insight into the data and should be used in evaluations to judge the merit of more complex models. The basic ideas of the Elo system and time decay functions can also be incorporated into other models, e.g., time decay functions could be quite naturally incorporated into performance factor analysis [19].

Some of the natural features of these approaches can also be useful for intelligent tutoring and adaptive practice. Consider two students with the following history of answers to a particular knowledge component: student A: 1, 1, 1, 1. student B: 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1. Immediately after these sequences, it is not useful to give any of these two students more problems about this knowledge component, as there is a high probability of a correct answer. But there is clearly a difference between these students – whereas student A probably has solid knowledge and there is little use in returning to the knowledge component in the future, for student B a review in the future would be certainly useful. If we summarise the skill by a single number as is typically done by BKT, it is hard to capture this difference. Using time decay functions, it is easy to cover this situation – we can estimate a "current skill" using a steep time decay function and a "long term skill" with a flat time decay function.

Recently, there has been several works that studied the Elo system in the context of student modeling and adaptive practice [1, 13, 26, 27, 28]. However the impact so far has been rather marginal, particularly compared with Bayesian knowledge tracing. As the discussion above suggest, the approach deserves more attention.

Acknowledgement

The author thanks Matěj Klusáček and Libor Vaněk for performing some of the experiments and for discussions about the topic of the paper.

5. REFERENCES

[1] M. Antal. On the use of elo rating for adaptive assessment. *Studia Universitatis Babes-Bolyai, Informatica*, 58(1), 2013.

[2] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[3] R. De Ayala. *The theory and practice of item response theory*. The Guilford Press, 2008.

[4] A. E. Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.

[5] S. Frederick, G. Loewenstein, and T. O'donoghue. Time discounting and time preference: A critical review. *Journal of economic literature*, 40(2):351–401, 2002.

[6] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):1–37, 2014.

[7] M. E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.

[8] R. Herbrich, T. Minka, and T. Graepel. Trueskill: A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576, 2006.

[9] P. Jarušek, M. Klusáček, and R. Pelánek. Modeling students' learning and variability of performance in problem solving. In *Educational Data Mining (EDM)*, pages 256–259. International Educational Data Mining Society, 2013.

[10] P. Jarušek and R. Pelánek. Analysis of a simple model of problem solving times. In *Proc. of Intelligent Tutoring Systems (ITS)*, volume 7315 of *LNCS*, pages 379–388. Springer, 2012.

[11] P. Jarušek and R. Pelánek. A web-based problem solving tool for introductory computer science. In *Proc. of Innovation and technology in computer science education*, pages 371–371. ACM, 2012.

[12] R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281–300, 2004.

[13] S. Klinkenberg, M. Straatemeier, and H. Van der Maas. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824, 2011.

[14] M. Klusáček. Modeling learning in problem solving. Master's thesis, Masaryk University Brno, 2013.

[15] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[16] B. Martin, A. Mitrovic, K. R. Koedinger, and S. Mathan. Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction*, 21(3):249–283, 2011.

[17] B. B. Nooraei, Z. A. Pardos, N. T. Heffernan, and R. S. J. de Baker. Less is more: Improving the speed and prediction power of knowledge tracing by using less data. In *Educational Data Mining (EDM)*, pages 101–110, 2011.

[18] J. Papoušek, R. Pelánek, and V. Stanislav. System for learning facts in domains with prior knowledge. In *Educational Data Mining (EDM)*, 2014. To appear.

[19] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *Artificial Intelligence in Education (AIED)*, pages 531–538. IOS Press, 2009.

[20] Y. Qiu, Y. Qi, H. Lu, Z. A. Pardos, and N. T. Heffernan. Does time matter? modeling the effect of time with bayesian knowledge tracing. In *Educational Data Mining (EDM)*, pages 139–148, 2011.

[21] A. Tsymbal. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 2004.

[22] B. van de Sande. Properties of the bayesian knowledge tracing model. *Journal of Educational Data Mining*, 5(2):1, 2013.

[23] W. Van Der Linden. Conceptual issues in response-time modeling. *Journal of Educational Measurement*, 46(3):247–272, 2009.

[24] L. Vaněk. Elo systém a modelování času řešení. Master's thesis, Masaryk University Brno, 2014. To appear.

[25] Y. Wang and N. Heffernan. Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes. In *Artificial Intelligence in Education (AIED)*, volume 7926 of *LNCS*. Springer, 2013.

[26] K. Wauters, P. Desmet, and W. Van Den Noortgate. Adaptive item-based learning environments based on the item response theory: possibilities and challenges. *Journal of Computer Assisted Learning*, 26(6):549–562, 2010.

[27] K. Wauters, P. Desmet, and W. Van Den Noortgate. Monitoring learners' proficiency: Weight adaptation in the elo rating system. In *Educational Data Mining (EDM)*, pages 247–252, 2011.

[28] K. Wauters, P. Desmet, and W. Van Den Noortgate. Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, 58(4):1183–1193, 2012.

[29] G. I. Webb and M. Kuzmycz. Evaluation of data aging: A technique for discounting old data during student modeling. In *Intelligent Tutoring Systems (ITS)*, pages 384–393. Springer, 1998.

Causal Discovery with Models: Behavior, Affect, and Learning in Cognitive Tutor Algebra

Stephen E. Fancsali
Carnegie Learning, Inc.
437 Grant Street, Suite 918
Pittsburgh, PA 15219 USA
1.888.751.7094 x219
sfancsali@carnegielearning.com

ABSTRACT

Non-cognitive and behavioral phenomena, including gaming the system, off-task behavior, and affect, have proven to be important for understanding student learning outcomes. The nature of these phenomena requires investigations into their causal structure. For example, given that gaming the system has been associated with poorer learning outcomes, would reducing such behavior improve outcomes? Answering this question requires an understanding of whether gaming the system is a cause of poor outcomes, rather than, for example, only sharing a common cause with factors influencing learning. Because controlled experiments to settle such causal questions are often costly or impractical, we employ algorithmic search for the structure of graphical causal models from non-experimental data. Using sensor-free, data-driven detectors of behavior and affect, this work extends Baker and Yacef's notion of "discovery with models" to incorporate causal discovery and reasoning, resulting in an approach we call "causal discovery with models." We explore a case study of this approach using data from Carnegie Learning's Cognitive Tutor for Algebra and raise questions for future research.

Keywords

Discovery with models, causal discovery, graphical causal models, probabilistic graphical models, gaming the system, affect, off-task behavior, sensor-free detectors, intelligent tutoring systems, Cognitive Tutor, measurement.

1. INTRODUCTION

Recently, researchers in educational data mining, learning analytics, and the learning sciences have used the moniker "discovery with models" to describe analyses in which "a model of a phenomenon is developed through any process that can be validated in some fashion…, and this model is then used as a component in another analysis, such as prediction or relationship mining" [10]. Examples of discovery with models range over a variety of constructs that capture student context and interaction with educational software and courseware [22] like help seeking strategies [2] and patterns of use of online resources (e.g., [23]).

We focus on models that function as sensor-free "detectors" that use data from student interactions with an intelligent tutoring systems (ITS) to predict whether actions are likely instances of particular forms of behavior or arise from a student being in a particular affective state. Such detectors (and corresponding constructs of interest) have been the topic of a great deal of literature in educational data mining and the learning sciences; predicted constructs include "gaming the system" [6,8], off-task behavior [3], affective states (e.g., boredom and engaged concentration) [9], and carelessness [39], among others. These detectors are generally validated by comparing their data-driven predictions (e.g., whether a student is likely to be gaming the system or bored at a particular [interval of] time) to classifications provided by trained observers in a classroom or computer laboratory environment (cf. [30]).

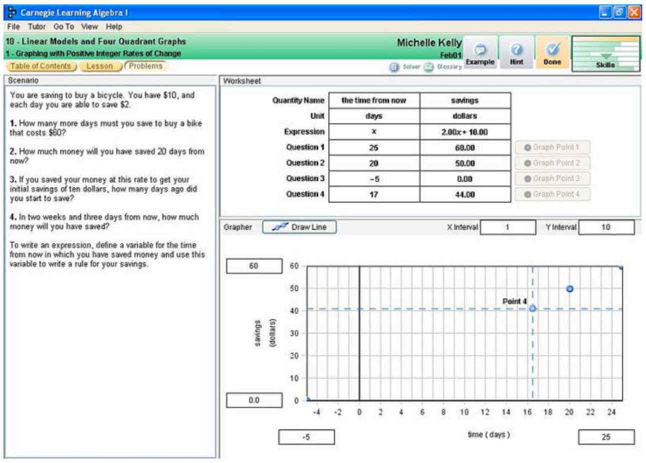
While discovery with models approaches have been used to associate learning outcomes with various constructs, we suggest that many constructs of recent interest are especially important because of underlying causal questions: What causes behavior like gaming the system? What makes students bored, careless, or frustrated? What are causal links (if any) among such modeled constructs, and are they causally linked to outcomes like learning? Gaming the system, for example, and learning are found to be negatively associated in several studies (e.g., [14,31]), but the mere association of gaming behavior and learning does not imply that if we reduced gaming we would increase learning. Perhaps both are caused by some other factor (like motivation), making a focus on gaming behavior itself ineffective in increasing learning. Ideally, researchers would settle causal questions using randomized experiments (e.g., A/B software tests, randomized controlled trials), but often such experiments, if possible, are expensive, difficult, or unethical. Given non-experimental ITS log data, and its wide availability from sources like the Pittsburgh Science of Learning Center's DataShop [26], we turn to algorithmic methods to discover causal structure from observational data.

After describing Carnegie Learning's Cognitive Tutor® (CT) Algebra [36] ITS and several constructs for which sensor-free detectors have been developed, we briefly explicate the framework of data-driven search for the structure of graphical causal models. We then apply this framework in a discovery with models approach to find causal explanations that integrate aspects of behavior, affect, and learning in ITSs. Finally, we describe several important and interesting problems, especially but not limited to measurement problems, at the intersection of causal discovery from non-experimental data and discovery with models in educational data mining (i.e., "causal discovery with models").

2. PRELIMINARIES

2.1 Cognitive Tutor (CT) Algebra

CT Algebra is an ITS with hundreds of thousands of middle school and high school users both in the United States and internationally. Increasingly, CT Algebra is also deployed in higher education settings. The CT adaptively presents mathematics content to students by tracking their mastery of fine-grained knowledge components (KCs) or skills, into which mathematics content has been atomized, as they work through (parts of) problems (cf. the screenshot of Figure 1). At each problem-solving step, students can request context-sensitive hints and receive immediate feedback about correctness that is sometimes accompanied by just-in-time, context-sensitive feedback that is more detailed.



You are saving to buy a bicycle. You have $10, and each day you are able to save $2.

1. How many more days must you save to buy a bike that costs $60?

2. How much money will you have saved 20 days from now?

3. If you saved your money at this rate to get your initial savings of ten dollars, how many days ago did you start to save?

4. In two weeks and three days from now, how much money will you have saved?

To write an expression, define a variable for the time from now in which you have saved money and use this variable to write a rule for your savings.

Quantity Name	the time from now	savings
Unit	days	dollars
Expression	x	2.00x + 10.00
Question 1	25	60.00
Question 2	20	50.00
Question 3	-5	0.00
Question 4	17	44.00

Figure 1. Screenshot of problem solving in CT Algebra

The CT deploys content that is divided into curricular units comprised of (roughly topical) sections. When the CT judges a student, using a framework called Bayesian Knowledge Tracing (BKT) [15], to have reached mastery of all the KCs in a particular section, she is graduated to the following section (or unit if she has completed all the sections in a given unit).

2.2 Data-Driven Detectors of Behavior and Affect

Recent work has developed a variety of data-driven, sensor-free "detectors" to infer or measure different features of student interactions with educational courseware, especially ITSs. In this work, we focus on using detectors to infer aspects of learners' gaming the system, off-task behavior, and affective states while interacting with the CT Algebra ITS.

2.2.1 Gaming the System & Off-Task Behavior

A great deal of recent work has been directed at using data-driven, predictive models to measure or infer disengaged behavior, including gaming the system and off-task behavior, and linking such behavior to learning outcomes using discovery with models techniques. Gaming the system [5-7] is characterized as behavior that allows for progression through curricular material without genuine learning by taking advantage of ITS affordances available to the learner. In general, such behavior can be broadly characterized by learners' abuse of hints [1], including cycling through hints until the last hint (i.e., a "bottom out" hint) is reached that provides the answer to a problem-solving step, and by rapid and/or systematic guessing [7]. While some gaming behavior has been called "non-harmful" because it is not associated with decreased learning, there is a great deal of evidence for "harmful gaming" that is associated with decreased learning [6,7]. The "harmful" modifier can (at least tacitly) be read causally, even if generally used to describe merely correlational results, so one research question involves determining whether we can provide evidence from non-experimental data for the claim that gaming the system is a *cause* of decreased learning. Off-task behavior refers to behavior that is disengaged and/or unrelated to learning or the learning environment [3].

A variety of data-driven detectors of gaming the system and off-task behavior have been developed in recent years (e.g., [11,24,43]). We deploy detectors of gaming the system [8] and off-task behavior [3] developed for CT Algebra, the statistical basis of which are Latent Response Models [29]. These detectors employ features "distilled" from fine-grained CT process data that capture the types of behavior we described above (e.g., for gaming the system: quick actions after making at least one error on a problem-solving step [7]; for off-task behavior: extremely fast or extremely slow actions [3]). Detectors generate a prediction for each learner action in the CT as to whether it is likely an instance of gaming the system or off-task behavior. These predictions can then be "rolled-up" to the level of consecutive actions on a particular problem solving-step (i.e., roughly consecutive actions involving the same KC). If any one action within a problem-solving step is determined to be gamed or off-task, then we call that step gamed or off-task, following other applications of these detectors (e.g., [14]).

2.2.2 Affective States

While evidence suggests that learner affect can influence learning (e.g., [16,33]), measurement and assessment of affect, whether via surveys, physical sensors, or direct observation, can be obtrusive, time-consuming, and suffers from a lack of scalability to larger numbers of learners over longer periods of time. In an effort to overcome these obstacles, recent work [9] has taken a data-driven, sensor-free approach to infer learner affect from ITS process data, much like that adopted to infer gaming and off-task behavior.

As with gaming and off-task detectors, in the development of affect detectors, a wide variety of features are distilled from CT process data, but rather than learning a Latent Response Model, machine learning classifiers are applied to features of "clips" of problem-solving (durations of learner actions up to twenty seconds in length) to classify them as likely corresponding to learners being in a state of boredom, confusion, engaged concentration, or frustration. Boredom in a particular clip can be detected, in part, through features like the maximum number of previous incorrect actions and hint requests for any skill in the clip. Confusion in a clip can be detected, for example, using the percentage of actions that take longer than five seconds after two incorrect answers. The duration of the fastest action in a clip is one feature upon which the detector of engaged concentration relies. Finally, frustration can be detected, in part, by the percentage of past actions on skills in a clip that were incorrect [9].

These classifiers can then be applied to new data to generate predictions about problem-solving clips and their correspondence to learner affective states. We now review several successful instances of discovery with models approaches using detectors to predict external, student-level learning outcomes. These results demonstrate correlations between learning outcomes and gaming the system, off-task behavior, and affective states, but we seek further insight into whether non-experimental data alone can provide evidence for causal claims about the impact of these phenomena on learning.

2.3 Prior Work: Using Models of Behavior & Affect to Predict Learning Outcomes

Several recent projects have used data, aggregated over fine-grained predictions, from these detectors as input to statistical, predictive analyses of student-level, substantive learning outcomes (i.e., adopting a discovery with models approach). One study used aggregate counts of gaming and off-task problem-solving steps to predict post-test scores for several units of CT content [14]. These researchers built linear regression models for each CT unit they considered, summarizing their results by reporting that gaming the system was weakly associated, and off-task behavior strongly associated, with poorer learning in the aggregate.

Later work has successfully used detectors of gaming the system, off-task behavior, and affect on data from the ASSISTments system [21], to predict Massachusetts Comprehensive Assessment System (MCAS) standardized test scores [31] and college enrollment (in a different population and study) some time after using the software [38]. The former study reports pairwise correlations between variables aggregated from detector predictions and raw MCAS scores, treating different types of ASSISTments' problems separately. For our purposes, ASSISTments' "scaffold" problems, presented after a student has made a mistake or asked for help, are most relevant to understanding student behavior in the CT, as their structure is the norm for CT problems.

Boredom, confusion, engaged concentration, and frustration on scaffold problems are all positively and significantly correlated with MCAS scores across two academic years of data. They report mixed results (one year positive, one negative) about the correlation of off-task behavior with MCAS scores and that gaming the system is significantly, negatively correlated with MCAS scores. A logistic regression model of college enrollment based on detectors also found significant, positive associations between enrollment and both boredom and confusion [38].

Having summarized several correlational studies that exemplify discovery with models using data-driven detectors, we now introduce the framework of graphical causal models and algorithmic search procedures to learn causal structure from non-experimental data. Our aim will then be to use detector predictions as input to these procedures to go beyond analysis of correlations.

3. GRAPHICAL CAUSAL MODELS & CAUSAL DISCOVERY

To learn causal relationships among the phenomena of gaming the system, off-task behavior, affective states, and learning, we adopt the formalism of graphical causal models, specifically causally interpreted directed acyclic graphs (DAGs), to represent the qualitative causal structure among variables of interest [32,42]. Such models have been used to better understand causal relationships among various phenomena in ITSs (e.g., [20,34,35]) and in educational technology more generally (e.g., [17,40]).

Under the causal interpretation of DAGs, nodes represent random variables and directed edges represent direct causal relationships, relative to the set of variables in the model. For linear causal relations and multivariate normal joint probability distributions, DAGs imply (conditional) independence constraints on observed distributions or covariance matrices. Whether particular constraints obtain for observed data can be ascertained by statistical tests for whether appropriate partial correlations vanish.

However, it is often the case that more than one DAG is consistent with the same set of (conditional) independence constraints; that is, causal structure is underdetermined by non-experimental data, and members of the set of all of DAGs consistent with the same constraints (i.e., members of an equivalence class of DAGs) are indistinguishable from observation alone. Consider, for example, a simple case of three observed variables X, Y, and Z, no pair of which shares any unmeasured common cause. Suppose the pair-wise correlation of each pair of variables is non-zero, and that by a statistical test we determine that the sample partial correlation $\rho_{X,Y.Z}$ vanishes (i.e., $\rho_{X,Y.Z} = 0$). Three DAGs are consistent with this conditional independence relationship (i.e., are members of the equivalence class consistent with this constraint):

- $X \rightarrow Z \rightarrow Y$
- $X \leftarrow Z \rightarrow Y$
- $X \leftarrow Z \leftarrow Y$

Beyond data-driven constraints, background and domain knowledge are also important. If we knew, for example, that Z were prior in time to X and Y, then only one graph (the graph in which Z is a common cause of X and Y: $X \leftarrow Z \rightarrow Y$) is consistent with both the conditional independence constraint and background knowledge.

Researchers have developed asymptotically reliable algorithms[1] [42] to infer the equivalence class of causal graphs that are consistent with observed independencies, conditional independencies, and background knowledge, under different assumptions. We focus on two such algorithms. The PC algorithm [42] learns a graphical object called a pattern that represents the equivalence class of DAGs consistent with observed (conditional) independencies and background knowledge, assuming there are no unmeasured (i.e., latent) common causes of measured variables. Qualitative causal structure of a DAG member of the class represented by a pattern (each member of which will fit the data equally well) can be used to specify a linear structural equation model. Estimating parameters of such a model allows for path analysis and the consideration of quantitative causal effects (as we will see in §5.3.1).

Since the assumption of no latent common causes is implausible for most real-world scientific settings, we also consider search using the FCI algorithm [42], which allows for the possibility of latent common causes. While FCI is similar to PC in many ways, the graphical object it learns from data, called a Partial Ancestral Graph (PAG), also represents an equivalence class of causal graphs but is more expressive to allow for possible latent common causes. Edges in PAG causal models are interpreted as follows:

- X o—o Y: (1) X is an ancestor (i.e., cause) of Y; (2) Y is a cause of X; (3) X and Y share a latent common cause; (4) either (1) & (3) or (2) & (3).
- X o→ Y: Either X is a cause of Y; X and Y share a latent common cause; or both.
- $X \leftrightarrow Y$: X and Y share a latent common cause in every member of the equivalence class represented by this PAG.
- $X \rightarrow Y$: X is an ancestor/cause of Y in every member of the equivalence class represented by this PAG.

A graph containing this last type of edge represents a case where we can make causal inferences despite the assumption that there may be latent common causes of measured variables. We now summarize our data as well as how we construct variables, from

[1] implemented and made freely-available by the Tetrad Project (http://www.phil.cmu.edu/projects/tetrad/)

predictions of detector models, to use as input to causal structure search algorithms.

4. DATA

4.1 Overview

We consider CT Algebra log data over a sample of 102 learners who completed an algebra course in a higher education context. Specifically, we focus on a module of CT Algebra units presented at the end of a particular course that included the following units of instruction:

- Systems of Linear Equations
- Systems of Linear Equations Modeling
- Linear Inequalities
- Graphing Linear Inequalities
- Systems of Linear Inequalities

In addition to pre-test scores for this module of instruction and final exam scores for the *entire* algebra course for each of the 102 learners, we constructed aforementioned data-driven detectors of gaming the system, off-task behavior, and various affective states, including boredom, confusion, engaged concentration, and frustration, from fine-grained log files containing roughly 337,000 student actions. We learned BKT parameters, required as input to these detectors, for the 32 KCs in our data using a brute-force method [4].

4.2 Variable Construction

Since it is implausible that any particular interaction (e.g., gaming the system on a particular problem solving step in CT) is attributable as a cause of aggregate student learning, we seek aggregate patterns of interaction (i.e., variables aggregated at the level of students) over which to learn causal models and provide causal explanations. That is, our present project is to provide causal models that could explain relationships among *student-level* behavior, affect, and learning, but the results of the detector models we seek to use as a component of (i.e., as input to) causal search algorithms are fine-grained (i.e., predictions about behavior and affect during many problem solving steps or clips of interaction per student).

Previous work [18,19] provided a preliminary exploration of this data set using algorithmic causal search over variables defined as student-level *counts* of gamed or off-task problem-solving steps (as assessed by appropriate detectors), following other work that modeled aggregate variables constructed from predictions of these detectors [14]. The current project extends this exploration by integrating detectors of learner affect and constructs variables differently, roughly following more recent, aforementioned research using detectors to predict learners' MCAS scores [31].

We define variables for the *proportion* of problem-solving steps, per learner, that are judged to be instances of gaming and off-task behavior and the proportion of problem-solving clips (i.e., longer durations of problem-solving activity) at which learners are judged to be in particular affective states. Constructing variables in this way allows for each variable to represent the proportion of the student's CT interaction in which they behaved in a particular way or were inferred to be in a particular affective state, eliminating the complication of counting steps versus clips for the two different types of detectors deployed. Despite modest differences in variable construction, high-level results we now present are consistent with these previous modeling efforts using variables defined or constructed as counts.

5. RESULTS

We begin describing our results by summarizing learner behavior and affect. Then we present pair-wise correlations of modeled behavior and affect variables with learning before presenting structural, causal models that explain patterns of conditional independence among these measures.

5.1 Relative Frequencies of Behaviors and Affective States

As a check on the applicability of the detectors, we consider the relative frequency with which particular behavioral and affective predictions are made by the detectors we used. Our findings roughly align with previous applications to data from (and observations of) CT Algebra and other tutors (cf. [9]; [S.M. Gowda, personal communication]). Over all 102 students and all usage, 41% of steps are determined to be instances of gaming the system while 4.4% of steps are deemed off-task. The detectors of affect infer that 5.87% of all problem-solving clips correspond to learner boredom; 3.52% of clips correspond to learner confusion; 67.5% of clips are inferred to be instances of engaged concentration, and .8% of clips correspond to frustration.

The relatively large percentage of gaming the system may be partially attributable to the fact that the detector, as in previous studies of the aggregate impact of gaming (e.g., [14]), does not distinguish between what has been called "harmful" and "non-harmful" gaming. Moreover, some behavior inferred to be gaming might be helpful for learning, as, for example, when students seek "bottom out" hints as worked examples [41].

5.2 Correlations with Learning

Next, we consider pairwise correlations of each modeled behavior and affective state variable and our learning outcome, the algebra course final exam score. We report Pearson correlation coefficients in Table 1 (noting significance of each according to the two-tailed t-test for such coefficients).

It is perhaps unsurprising that *Gaming the System*, *Off-Task Behavior*, and *Frustration* are negatively correlated with learning and *Engaged Concentration* is positively correlated with learning. That *Boredom* and *Confusion* are both positively correlated with learning, while perhaps surprising, is consistent with predictive results reported in both the MCAS and college enrollment studies we briefly summarized in §2.3 that used ASSISTments data. In considering causal models of these constructs in the following section, we provide possible explanations for the directions of these correlations and associations.

Table 1. Pairwise correlations of learning (i.e., final exam score) and variables representing "detected" behavior and affective states (p < .01; ***p < .001)**

Variable / Construct	Pearson Correlation
Boredom	.18
Confusion	.31**
Engaged Concentration	.55***
Frustration	-.27**
Gaming the System	-.63***
Off-Task Behavior	-.09

5.3 Causal Model Discovery

To learn structural, causal models to help explain these pairwise correlations, we apply aforementioned search algorithms to learn qualitative causal structure. So that we might provide a robust analysis, we consider three different sets of assumptions, including the temporal ordering of variables as background knowledge, that constrain the search for causal structure and the types of inferences that can be made. The issue of temporal ordering is especially important given a lack of agreement about whether behavior precedes affect, vice versa, or they co-occur (cf. [9]). We begin with the strongest assumptions, relax those assumptions, and then briefly consider robustness of causal inferences across these assumptions.

5.3.1 *No Unmeasured Common Causes & Affect Precedes Behavior*

We begin with two relatively strong assumptions. First, we assume that there are no unmeasured common causes of measured variables. This assumption is unlikely to hold in most real world settings. Second, we assume that learner affect causally precedes behavior. We constrain the PC algorithm's search space by providing background knowledge that *Module Pre-Test* precedes *Confusion*, *Engaged Concentration*, and *Boredom*, and that these three affective states precede *Gaming the System* and *Off-Task Behavior*. Finally, the course *Final Exam* is the last variable in this ordering.[2]

Using the qualitative causal structure inferred with the PC algorithm,[3] we specify and estimate parameters of the linear structural equation model (graphically represented) in Figure 2. In such a model, each variable is a linear function of its parents (i.e., direct causes) and an independent, normally distributed error term (omitted from Figure 2). This linear model fits the observed data well, as assessed by a chi-square test comparing the implied covariance matrix to the observed covariance matrix ($\chi^2(13) = 14.64$; $p = .33$) [13]. While we will find that the inferred causal relationship between *Gaming the System* and *Final Exam* (i.e., learning) is robust across *all* sets of assumptions we will consider, given the assumptions we have made so far, other edges in the graph should be interpreted cautiously.

Keeping this caveat in mind, considering the structure and parameters of the model in Figure 2, we see that *Module Pre-Test* is directly linked only to *Engaged Concentration*; learners with greater pre-test scores tend to concentrate more. Learners with a higher proportion of *Engaged Concentration* tend to game the system less and go off-task less, and we have already seen that gaming is strongly, negatively correlated with *Final Exam*, our learning outcome.

Off-Task Behavior has no direct effect on *Final Exam* (or on *Gaming the System*), which we will see is also a robust finding. Interestingly, since *Confusion* is positively correlated with *Engaged Concentration* and negatively correlated with *Gaming the System*, one explanation of the positive, pair-wise correlation of *Confusion* and *Final Exam* is that increased *Confusion* virtuously leads both directly and indirectly (via leading students to better concentration) to less *Gaming the System*. This provides one possible causal explanation consistent with recent literature showing that confusion can be beneficial for learning (e.g., [27,28]).

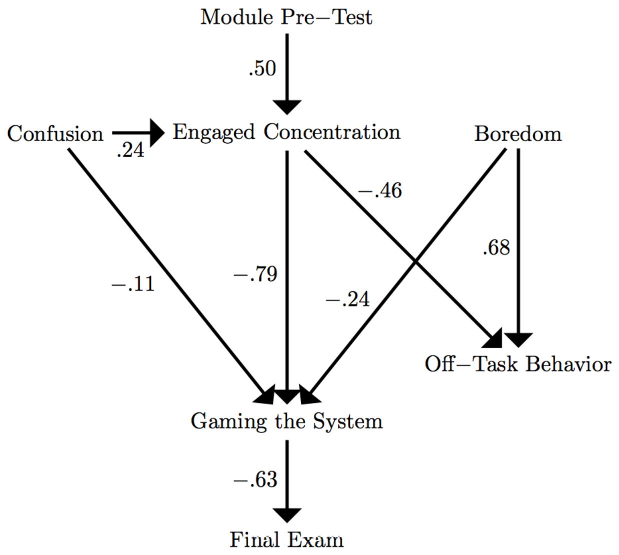


Figure 2. Estimated linear structural equation model

That increased *Boredom* may contribute to less *Gaming the System* and better learning is consistent with an aforementioned finding [31], but the proposed explanation in that work posits an unmeasured common cause (here also unmeasured) of boredom in "scaffold" questions and better learning. Carelessness, for example, rather than a lack of skill mastery, may drive students to answer incorrectly on originally presented questions, forcing learners into ASSISTments' scaffold questions; consequently learners become bored. Apropos, we consider relaxing the assumption that there are no unmeasured common causes and of the ordering of affect and behavior, allowing that neither precedes the other, but rather that they may co-occur.[4]

5.3.2 *Relaxing Assumptions*

The result of relaxing these two assumptions and applying the FCI algorithm to our data is the PAG causal model of Figure 3. First, despite relaxing both of our relatively strong assumptions, we still make the positive inference that *Gaming the System* is a cause of decreased learning (i.e., is generally "harmful" in the aggregate). Next, we see that inferring affective causes of *Gaming the System* is more complicated. Both *Engaged Concentration* and *Boredom* are found to share at least one unmeasured common cause with *Gaming the System*; the same is true of their relationships with *Off-Task Behavior*. However, despite relaxing the ordering of affect preceding behavior, we still find that *Confusion* is possibly a cause of *Gaming the System*, though the two may also share an unmeasured common cause. Finally, *Module Pre-Test* and *Engaged Concentration* are also possibly confounded, but this is perhaps unsurprising because, at best, such a pre-test is a noisy measure of prior ability.

[2] We omit *Frustration* from our analysis because it is relatively rare, and we were unsuccessful in inferring linear models that fit observed data well when we included it in the search. Future work should determine whether frustration in ITS environments is so rare. Assuming we are inferring a valid affective feature, finding appropriate means for analysis is also an important topic for future work.

[3] While in general we learn an equivalence class of causal graphs with the PC algorithm, in this case, our assumption of temporal ordering leads us to the unique DAG structure illustrated by Figure 2.

[4] While perhaps less likely given the relatively short span of the single CT Algebra module we consider, cyclic relationships over time between behavior and affect might be fruitfully treated in an acyclic setting by constructing appropriate, time-indexed (e.g., section-by-section or unit-by-unit) aggregate variables. This is a topic for future research.

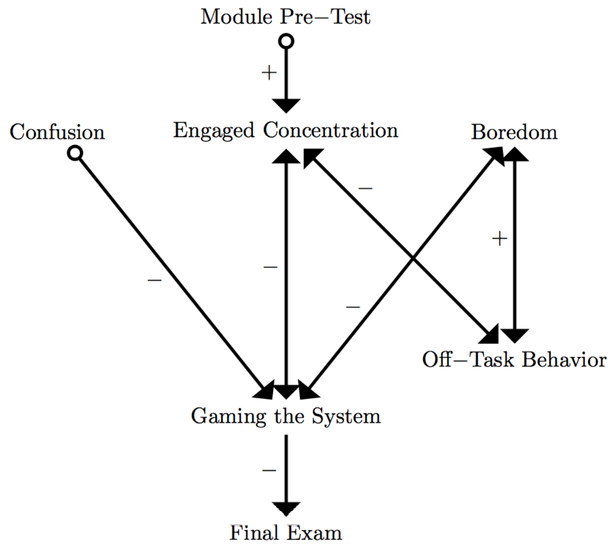


Figure 3. PAG causal model learned with weaker assumptions

5.3.3 Robustness

That affect and behavior co-occur is the weakest assumption and possibly the most reasonable, but we also inferred PAGs assuming that behavior precedes affect and vice versa. Most notably, *Gaming the System* is inferred as a non-confounded cause of *Final Exam* in both cases. However, when behavior is assumed to precede affect, positive causal inferences are also made that *Gaming the System* is a non-confounded cause of all three affect variables. While *Gaming the System* might trivially lead to less *Confusion* (i.e., attempts to avoid genuinely learning should not increase confusion), *Gaming the System* could both impinge upon concentration and decrease *Boredom* in more substantive ways. In the case in which affect precedes behavior (as we assume in the model of Figure 2), we find that *Engaged Concentration* causes decreases in both *Off-Task Behavior* and *Gaming the System*, but relationships between *Confusion*, *Boredom*, and *Gaming the System* may be confounded.

Simply because more positive causal inferences can be made given a particular temporal ordering does not provide evidence that we have arrived at the "correct" ordering. Questions like this should be settled by some combination of theoretical considerations, experimental results, and data analysis. The most important conclusion we reach from examining different sets of assumptions is that the positive inference about *Gaming the System* (i.e., that it is harmful to learning, in the aggregate) is robust across all of them. The negative finding that *Off-Task Behavior* is not a cause of learning is also robust. Lacking robustness for other inferences, we find the model of Figure 3 with weaker (i.e., modest) background knowledge, allowing that behavior and affect co-occur, most plausible and return to it in our discussion.

6. DISCUSSION

This work makes at least two important contributions. First, we have illustrated an approach, combining discovery with models and methods for causal discovery from non-experimental data, that we call causal discovery with models. Second, we have demonstrated that this approach can be used to provide evidence about relationships among important constructs of interest that are not always practical targets of randomized experiments.[5] Specifically, we have provided evidence that gaming the system is, in fact, harmful (i.e., both negatively correlated and likely causally related) to aggregate learning for a relatively novel sample of learners in a higher education context.

Notably, we do not have ground truth labels for our data (e.g., field observations of whether students are off-task or bored), so in this sense this work helps to generalize the idea that these detectors can be applied to data from new student populations and yield interesting connections to learning. Nevertheless, further research is necessary to determine what unmeasured common causes may confound relationships between affect and behavior, either because of measurement problems or because of phenomena we have not included in these models (e.g., carelessness, motivation, etc.).

A perhaps underappreciated problem for a variety of discovery with models approaches concerns the process by which the output or results of particular models (here, fine-grained detector predictions concerning behavior and affect) are used to construct variables that are used as input in other analyses (here, causal graph search). How do we best use the output or results of a particular model as a component (possibly components) of other analyses? Previous work [18,19] explored this problem as one of semi-automated search for constructed variables (including several different levels of aggregation and aggregation functions as suggested by [14]), but future work should explore better aggregate variable construction and feature engineering (including considerations of the interpretability of resulting features) as well as measurement models for these constructs.

To wit, despite sophisticated feature engineering used by detectors to make predictions and classifications, variables included in these models provide a relative paucity of information about the underlying phenomena of interest. More sophisticated measurement models might be used to either explicitly model latent phenomena or to develop improved, measured proxies (e.g., scales) to represent these constructs. As operationalized, *Boredom* and *Off-Task Behavior* may, for example, be confounded by boredom itself, as the constructed variable *Boredom* is only one noisy measure of the underlying phenomenon.

We should also include more phenomena in these models, including both latent phenomena like motivation and learner goals [12,20] and relatively simple process measures from CT log data. Prior work with this data set [18,19], for example, found that the count of actions that trigger context-sensitive, just-in-time feedback in the CT, while possibly less tenable as a target for future interventions, is highly correlated both with the final exam score and gaming the system. These prior results also suggest that this measure, or more likely the phenomena for which it stands in as a proxy,[6] is an intermediate link in a causal chain from *Gaming the System* to learning. Such measures, including other relatively simple measures of learner efficiency and assistance required by learners (e.g., the "assistance score" that sums hints requested and errors made [26]) that have been found to predict standardized test

[5] This is not to say that phenomena like gaming the system have not or cannot be targets of interventions (cf. [5]). However, the design and implementation of many experiments is likely to be non-trivial.

[6] possibly shallow learning or a learner's tendency to simply enter values that appear in a problem

scores for middle school CT users [37], do not require sophisticated feature engineering to achieve predictive access to learning outcomes and have been found to be highly correlated with gaming the system [25]. These and other measures should be further evaluated and explored. Their suitability to preserve (or induce) appropriate conditional independence relationships, necessary for modeling causal relationships in the framework we have described (given assumptions we have considered), should also be evaluated.

7. ACKNOWLEDGMENTS

Ryan S.J.d. Baker, Sujith Gowda, and Charles Shoopak provided code for (and vital assistance with) building predictive, "detector" models for gaming the system, off-task behavior, and student affect. I also thank Susan Berman, Ryan Carlson, David Danks, Ambarish Joshi, Tristan Nixon, Steven Ritter, Richard Scheines, and Michael Yudelson for their insightful comments and suggestions as this work has developed.

8. REFERENCES

[1] Aleven, V., Koedinger, K.R. 2000. Limitations of student control: do students know when they need help? In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems* (Montreal, Canada, 2000). 292-303.

[2] Aleven, V., Mclaren, B., Roll, I., & Koedinger, K. 2006. Toward meta-cognitive tutoring: a model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education* 16 (2006), 101-128.

[3] Baker, R.S.J.d. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proceedings of ACM CHI 2007: Computer-Human Interaction* (San Jose, CA, April 28 – May 3, 2007). ACM, New York, 1059-1068.

[4] Baker, R.S.J.d., Corbett, A.T., Gowda, S.M., Wagner, A.Z., MacLaren, B.M., Kauffman, L.R., Mitchell, A.P., Giguere, S. 2010. Contextual slip and prediction of student performance after use of an intelligent tutor. In *Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization* (Big Island, HI, 2010). Springer, New York, 52-63.

[5] Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Evenson, E., Roll, I., Wagner, A.Z., Naim, M., Raspat, J., Baker, D.J., Beck, J. 2006. Adapting to when students game an intelligent tutoring system. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (Jhongli, Taiwan, 2006). 392-401.

[6] Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. 2004. Off-task behavior in the Cognitive Tutor classroom: when students "game the system." In *Proceedings of ACM CHI 2004: Computer-Human Interaction* (Vienna, Austria, 2004). 383-390.

[7] Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R. 2008. Developing a generalizable detector of when students game the system. *User Model. User-Adap.* 18 (2008), 287-314.

[8] Baker, R.S.J.d., de Carvalho, A. M. J. A. 2008. Labeling student behavior faster and more precisely with text replays. In *Proceedings of the 1st International Conference on Educational Data Mining* (Montreal, 2008). 38-47.

[9] Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G.W., Ocumpaugh, J., Rossi, L. 2012. Towards sensor-free affect detection in Cognitive Tutor Algebra. In *Proceedings of the 5th International Conference on Educational Data Mining*, (Chania, Greece, 2012). 126-133.

[10] Baker, R.S.J.d., Yacef, K. 2009. The state of educational data mining in 2009: a review and future visions. *Journal of Educational Data Mining* 1 (2009), 3-17.

[11] Beal, C.R., Qu, L., Lee, H. 2006. Classifying learner engagement through integration of multiple data sources. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence* (Boston, MA, 2006). 151-156.

[12] Bernacki, M.L., Nokes-Malach, T.J., Aleven, V. 2013. Fine-grained assessment of motivation over long periods of learning with an intelligent tutoring system: methodology, advantages, and preliminary results. In *International Handbook of Metacognition and Learning Technologies*, R. Azevedo & V. Aleven, Eds. Springer, New York, 629-644.

[13] Bollen, K. 1989. *Structural Equations with Latent Variables*. John Wiley & Sons.

[14] Cocea, M., Hershkovitz, A., Baker, R.S.J.d. 2009. The impact of off-task and gaming behavior on learning: immediate or aggregate? In *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (Brighton, UK, 2009). 507-514.

[15] Corbett, A.T., Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adap.* 4 (1995), 253-278.

[16] Craig, S.D., Graesser, A.C., Sullins, J., Gholson, B. 2004. Affect and learning: an exploratory look into the role of affect in learning. *Journal of Educational Media* 29 (2004), 241-250.

[17] Fancsali, S.E. 2011. Variable construction for predictive and causal modeling of online education data. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge* (Banff, Canada, 2011). ACM, New York, 54-63.

[18] Fancsali, S.E. 2012. Variable construction and causal discovery for Cognitive Tutor log data: initial results. In *Proceedings of the 5th International Conference on Educational Data Mining* (Chania, Greece, 2012). 238-239.

[19] Fancsali, S.E. 2013. *Constructing Variables that Support Causal Inference*. Doctoral Thesis. Carnegie Mellon University.

[20] Fancsali, S.E., Bernacki, M.L., Nokes-Malach, T.J., Yudelson, M., Ritter, S. To appear. Goal orientation, self-efficacy, and "online measures" in intelligent tutoring systems. *Proceedings of the 36th Annual Conference of the Cognitive Science Society*. Cognitive Science Society, Austin, TX.

[21] Feng, M., Heffernan, N.T., Koedinger, K.R. 2009. Addressing the assessment challenge in an intelligent tutoring system that tutors as it assesses. *User Model. User-Adap.* 19 (2009), 243-266.

[22] Hershkovitz, A., Baker, R.S.J.d., Gobert, J., Wixon, M., Sao Pedro, M. 2013. Discovery with models: a case study on carelessness in computer-based science inquiry. *Am. Behav. Sci.* 57 (2013), 1479-1498.

[23] Jeong, H., Biswas, G. 2008. Mining student behavior models in learning-by-teaching environments. *Proceedings of the 1st*

International Conference on Educational Data Mining (Montreal, Canada). 127-136.

[24] Johns, J., Woolf, B. 2006. A dynamic mixture model to detect student motivation and proficiency. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence* (Boston, MA, 2006). 163-168.

[25] Joshi, A., Fancsali, S.E., Ritter, S., Nixon, T., Berman, S. To appear. Generalizing and extending a predictive model for standardized test scores based on Cognitive Tutor interactions. *Proceedings of the 7th International Conference on Educational Data Mining.*

[26] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. 2011. A data repository for the EDM community: the PSLC DataShop. In *Handbook of Educational Data Mining*, C. Romero, S. Ventura, M. Pechenizkiy, & R.S.J.d. Baker, Eds. CRC, Boca Raton, FL.

[27] Lee, D.M., Rodrigo, M.M., Baker, R.S.J.d., Sugay, J. Coronel, A. 2011. Exploring the relationship between novice programmer confusion and achievement. In *Proceedings of the 4th Bi-Annual International Conference on Affective Computing and Intelligent Interaction* (Memphis, TN, 2011). Springer, New York, 175-184.

[28] Lehman, B., D'Mello, S.K., Graesser, A.C. 2012. Confusion and complex learning during interactions with computer learning environments. *Internet High. Educ.* 15 (2012), 184-194.

[29] Maris, E. 1995. Psychometric latent response models. *Psychometrika* 60 (1995), 523-547.

[30] Ocumpaugh, J., Baker, R.S.J.d., Rodrigo, M.M.T. 2012. *Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. Training Manual Version 1.0*. Technical Report. New York, EdLab. [http://www.columbia.edu/~rsb2162/bromp.html]

[31] Pardos, Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda, S.M., and Gowda, S.M. 2013. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Proceedings of the Third International Learning Analytics and Knowledge Conference* (Leuven, Belgium, 2013). ACM, New York, NY, 117-124. DOI= http://doi.acm.org/10.1145/2460296.2460320

[32] Pearl, J. 2009. *Causality: Models, Reasoning, and Inference.* 2nd Edition. Cambridge UP, New York.

[33] Pekrun, R., Goetz, T., Titz, W., Perry, R.P. 2002. Academic emotions in students' self-regulated learning and achievement: a program of quantitative and qualitative research. *Educ. Psychol.* 37 (2002), 91-106.

[34] Rau, M., Scheines, R. 2012. Search for variables and models to investigate mediators of learning from multiple representations. In *Proceedings of the 5th International Conference on Educational Data Mining* (Chania, Greece, 2012). 110-117.

[35] Rau, M., Scheines, R., Aleven, V., Rummel, N. 2013. Does representational understanding enhance fluency – or vice versa? Searching for mediation models. In *Proceedings of the 6th International Conference on Educational Data Mining* (Memphis, TN, 2013). 161-168.

[36] Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.T. 2007. Cognitive Tutor: applied research in mathematics education. *Psychon. B. Rev.* 14 (2007), 249-255.

[37] Ritter, S., Joshi, A., Fancsali, S.E., Nixon, T. 2013. Predicting standardized test scores from Cognitive Tutor interactions. In *Proceedings of the 6th International Conference on Educational Data Mining* (Memphis, TN, 2013). 169-176.

[38] San Pedro, M.O.C.Z., Baker, R.S.J.d., Bowers, A.J., Heffernan, N.T. 2013. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *Proceedings of the 6th International Conference on Educational Data Mining* (Memphis, TN, 2013). 177-184.

[39] San Pedro, M.O.C.Z., Baker, R. S. J. d., Rodrigo, M. 2011. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In *Proceedings of 15th International Conference on Artificial Intelligence in Education* (Auckland, New Zealand). 304-311.

[40] Scheines, R., Leinhardt G., Smith, J., Cho, K. 2005. Replacing lecture with web-based course materials. *J. Educ. Comput. Res.* 32 (2005), 1-26.

[41] Shih, B., Koedinger, K. R., Scheines, R. 2008. A response time model for bottom-out hints as worked examples. In *Proceedings of the 1st International Conference on Educational Data Mining* (Montreal, Canada, 2008). 117-126.

[42] Spirtes, P., Glymour, C., Scheines, R. 2000. *Causation, Prediction, and Search*. 2nd Edition. MIT, Cambridge, MA.

[43] Walonoski, J.A., Heffernan, N.T. 2006. Detection and analysis of off-task gaming behavior in intelligent tutoring systems. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (Jhongli, Taiwan, 2006). 382-391.

Choice-based Assessment: Can Choices Made in Digital Games Predict 6th-Grade Students' Math Test Scores?

Min Chi
Department of Computer Science
North Carolina State University
Raleigh, North Carolina 27695
mchi@ncsu.edu

Daniel L. Schwartz
AAA Lab
Stanford University
Stanford, California, 94305 USA
danls@stanford.edu

Kristen Pilner Blair
AAA Lab
Stanford University
Stanford, California, 94305 USA
kpilner@stanford.edu

Doris B. Chin
AAA Lab
Stanford University
Stanford, California, 94305 USA
dbchin@stanford.edu

ABSTRACT

In this paper, we mined students' sequential behaviors from an instructional game for color mixing called Lightlet. Students pkaying the game have two broad strategies. They can either test candidate color combinations in an experiment room without risking an incorrect answer. Or they can choose colors from a faux shopping Catalog containing several different mixing charts. While the results shown in the Experiment Room are always correct, only a few of the charts in the Catalog are correct. Thus, if students use the catalog students must apply critical thinking skills to determine what charts to trust. Our primary goal in this work was to identify the crucial choice pattern(s) in students' game play that would contribute to their learning or subsequent performance. Data was collected from 6th graders. The results showed that children who chose to explore the Catalog of different charts during the game performed better in school. More specifically, the types of behavior choices students committed during the game play predicted about 43% of the variation in their subsequent math grades. This project shows that by assessing students' choices during learning, we can discover a great deal about their learning process and can identify and assess choices that are critical for learning but are often missed by most tests.

Keywords

Educational Assessments, Educational Games, Choice-based Assessment, Mining Behavior Data

1. INTRODUCTION

Educational assessment sits at the epicenter of learning research. Any quantitative study of an intervention, experience, or program to improve learning depends on the quality of the outcome measures. An ideal educational assessment would both reflect and reinforce the educational goals that society deems valuable. One fundamental goal of education is to prepare students to act independently in the world—which is to make good choices. It follows that an ideal assessment would measure how well we are preparing students to do so.

Most existing educational assessments are knowledge-based in that they focus on the amount of knowledge and skills students have accrued. Many such assessments use a format that Bransford and Schwartz (1999) labeled, Sequestered Problem Solving (SPS). In the typical SPS assessment, students are sequestered from learning opportunities and outside resources that might contaminate the validity of the assessment. Bransford and Schwartz argue that these retrospective measures are appropriate if the goal of instruction is training for highly stable performance conditions, but they are not optimally diagnostic when the goal is to prepare students to continue adapting and learning.

As an alternative, Bransford and Schwartz followed the theories of Vygotsky (1934) and Fueurstein (1979) to propose Preparation for Future Learning (PFL) assessments. In a PFL assessment, there are opportunities for learning during the assessment process, and the question is whether students are prepared to take advantage of these opportunities. These types of assessments are appropriate when the assumption is that students will need to continue learning, and the question is whether prior instruction and experiences have prepared them to do so. Multiple studies have shown the value of including PFL measures for assessing the quality of classroom instruction (Schwartz & Bransford, 1998; Schwartz & Martin, 2004; Chin, et al, 2010).

In the present work, we brought PFL assessments into an interactive context, where it is possible to directly measure processes associated with PFL. In our approach, choices, rather than knowledge, was the interpretative frame within which learning assessments are organized. In the following, we refer our approach as choice-based assessment.

Until recently, most researchers treated choice as a form of learning intervention. Iyengar and Lepper, (1999), for example, argued that giving students choices can increase their motivation and learning. Choice is important for learning if only because students need to experience choices in the protected atmosphere of education so they can learn how to handle them before becoming independent. Our approach is different. We ask why choices should be viewed as the outcome of learning and not solely an instructional ingredient to improve it. We contend that choice should be the interpretative framework for understanding and assessing learning outcomes. With new developments in technology, it should be possible to advance this goal which was beyond the reach of prior assessments.

One particularly promising way to integrate choice into educational assessment involves creating process measures that can capture student behaviors dynamically. Digital technologies make it possible to teach and assess student learning in new ways. Simply put, many new technologies are about choice. When browsing webpages, each click can be considered a choice about learning. When deciding what online sources to trust and which friends to consult, people are making learning choices. When using scientific simulations, people choose which sequence of settings that will yield the most telling results. Thus there is a good match between digital technologies and choice-based assessments.

Digital technologies make choice-based assessments possible, because interactive assessments can evaluate students in the context of choosing whether, what, how, and when to learn. By logging what students choose to do in an interactive environment, it is possible to gather functional process measures that can be expensive and difficult to gather by other means (e.g., Aleven et al., 2003; Baker, Corbett, & Koedinger, 2004; Hogyeong et al, 2008; Stevens & Thadani, 2007). Cognitive Tutors (Koedinger & Anderson, 1997), for example, track student progress and actions across multiple hours of use. Videogames include metrics of success and process (Gee, 2003).

However, the examples, which we describe below, all depend on large-scale environments that require many hours of interaction before any useful information can be gathered. To serve a broad range of goals, assessments need to be more nimble. We show that this is possible by demonstrating a digital choice-based assessment designed to assess critical thinking. This assessment is drawn from our work on Choicelets. It may not be what you expect in a test.

There are advantages to making smaller and more nimble environments for assessment. First, nimble assessments do not depend on students completing many hours of a complex game or instructional sequence before it is possible to make any useful assessments.

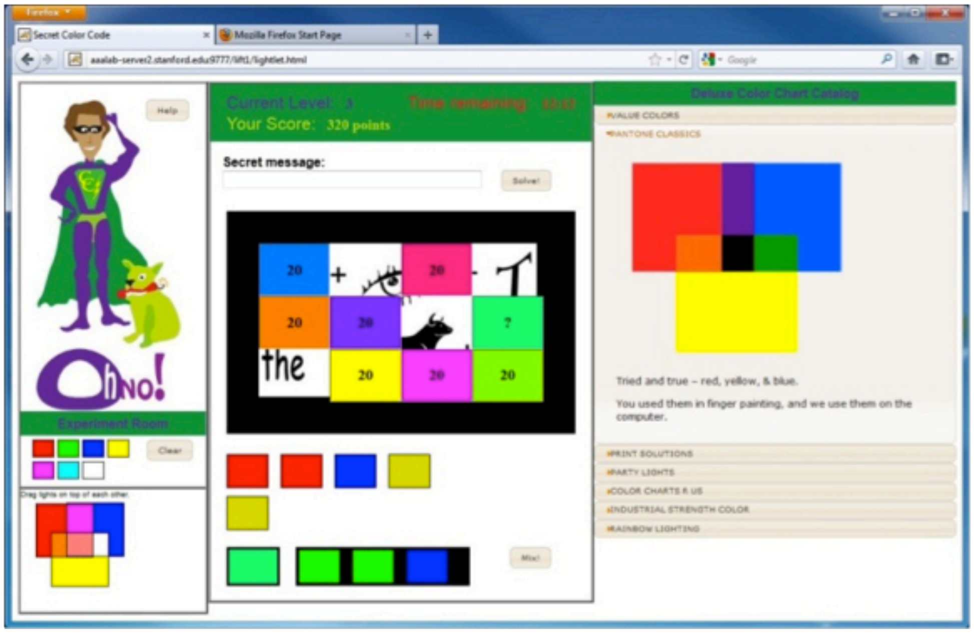
Second, smaller assessments can target specific choices design. This is quite different from searching for diagnostic patterns amid the millions of possible choice combinations in larger open environments.

Third, there is value to having an assessment that can be used to make general comparisons. In video games, cognitive tutors, and many embedded assessment tools, the assessments are locked into a specific model of instruction and delivery system. Thus they cannot be used to compare the effectiveness of different instructional models and learning experiences.

2. Choicelet

Choicelets take the form of short, and hopefully, engaging games that students want to complete. To complete the game, each Choicelet requires some learning, and we keep a log of students' choices during the process. Different Choicelets are designed to assess specific constellations of choices relevant to learning. In the present work we will focus on Lightlet a game designed to assess students' critical thinking skills. Most assessments of critical thinking evaluate deductive reasoning; for example, the ability to recognize when assumptions do not lead to conclusions. We chose instead to reclaim the broader meaning of critical thinking as the process of rationally deciding what to believe (Norris, 1985). Therefore, in Lightlet, we assess the decision to engage in critical thinking for the purpose of learning.

Figure 1 shows the main interface of the Lightlet. To play Lightlet, students mix the primary colors of light to move through a series of puzzle levels. The main component is a game board with colored light tiles, shown in the center of the screen. The first step in the game is to pick a colored tile from the game board. This is the *target color*. There is no constraint on the ordering of tiles to play and students can play them in any order. The students then select from the colors shown below the game board. The colors include the three primary colors (red, green, blue) and one non-primary color. With a click of the mix button, the selected colors mix. If the mixed colors produce the target color, then the tile disappears from the game board and reveals a portion of a rebus (picture that makes a phrase); otherwise, the tile remains. There is no limit on the number of times that a student can try for each tile. Once students remove all of the tiles from the board or reveal enough of the rebus to guess it correctly, then they can move to the next level of the game.



Tried and true – red, yellow, & blue.

You used them in finger painting, and we use them on the computer.

Figure 1. Lightlet GUI showing the Experiment Room (lower-left), Game Board (center) and Catalog (right).

Most students know about the primary colors for mixing paint or *subtractive color*. They are: red, yellow and blue. Mixing light or *additive color* however, depends upon a different set: red, green, and blue (RGB). Red + green makes yellow, and red + green + red makes orange. Thus, a major part of the game involves learning about additive color.

Lightlet includes a pair of resources to help students learn both of which are shown in Figure 1. On the lower left-hand side of the screen is the Experiment Room, where students can try out different color combinations without risking a wrong answer in the game board. In the Experiment Room, students can use seven base colors including the three primary colors. There is no upper limit on the number of times that each base color can be used. Students can clear the experiment room at any time with the 'clear' button.

On the right, there is a faux shopping Catalog in which different companies sell charts for mixing colors. There are seven charts available. Two of them are for additive color (i.e. light) and are correct. The remainder are for subtractive color mixing (i.e. paint) and are thus incorrect. The chart shown in Figure 1 is incorrect and is designed to play into students' prior beliefs about mixing paint. The descriptive text for the chart says: "Tried and true, red, yellow, and blue. You used them in finger painting! Use them now." The Experiment Room correctly shows that yellow and blue make white while the Catalog entry shows that they make green. Students must use critical thinking to decide which charts to believe, if they choose to use them at all. We track each of the students' choices during the game play in the log files.

There are three levels in Lightlet. The introduction (level 1), simply involves mixing red and blue lights to make three colors: red, blue, and magenta. This task is easy as it conforms to students' existing assumptions about mixing paint. Only the experiment room is available in this level.

Level 2 provides for full game play. In this level the color mixing challenges become harder in that students can use all three primary colors (e.g. red + green = yellow); we open both resources (the Catalog and Experiment Room) so that players can use them to figure out how to mix light. It is in this level where we collect the process data of most interest. Students may be induced to use *trial and error* in the Experiment Room or using the chart Catalog. If the students who used the chart Catalog did better, then there would be a warrant that this is a better choice pattern.

In the more advanced level (level 3) students are provided with new, even harder challenges as students can use each primary color twice (e.g., making orange, which requires red+red+green). And both the Catalog and Experiment Room are available for all players. Figure 1 shows this level of the game, which includes color combinations that depend on mixing three lights together.

3. Two Dominant Choices on Lightlet

There were two dominant patterns of choices in Lightlet: One pattern, The catalog-related choice pattern occurred when students chose to figure out which of the color charts are correct.

The other pattern, the Experiment Room-related, happens when students use the Experiment Room to solve the problems. These students would mix colors in the Experiment room to determine which colors to mix for the gameplay. Once they find the answer in the Experiment Room, they then choose the corresponding color on the game board and mix colors correctly.

In brief, on Lightlet, students have to learn the rules of additive color, they have an Experiment Room in the lower left corner, and they have a set of Catalog charts that show different color mixing results, some of which are subtractive charts and some of which are additive. Our question is: Do students choose to engage in critical thinking by deciding what charts to believe or they choose to try-and error in Experiment Room?

We hypothesize that students who applied the Experiment Room choice pattern had learned to solve problems one at a time rather than trying to find a general explanation. In math class, one can imagine students working to get the right answer for each separate math problem without attempting to find the deeper explanation that handles all possible related problems. The students who spent their time trying to decide which color chart to believe, on the other hand, were trying to find the *general* framework that could handle any colors in the game.

4. Teaching Students General Explanation

Prior research has shown the advantage of asking students to generate a general explanation that can handle all cases in a given task. It is much like finding a good theory can explain the results of multiple experimental conditions. For example, in a series of studies, students were provided with sets of contrasting cases designed to help them induce the structure of density (Schwartz, et al. 2011). When students were asked to invent a single procedure for generating a "crowdedness index" for the cases, they learned the ratio structure of density and spontaneously transferred to new problems. They outperformed control students, who were told about density at the outset and then applied the formula to the exact same contrasting cases.

In the current study, we first conducted pre-Lightlet training on generating general explanation. During the training, students were asked to find the similarities and differences between a series of contrasting cases in two physics tasks for two consecutive Friday classes (50 min each), one task per class. Half of the students, the experimental group, were explicitly encouraged to produce a *comprehensive, general* explanation of the similarities and differences. In other words, the experimental group was tasked with finding an underlying general structure or framework that explains all contrasting cases while the control group were tasked with finding the similarities and differences between the cases and they were not explicitly tasked to generate any general explanations or framework.

The two groups were treated identically when using Lightlet. Because the task of determining which set of charts to trust would help students find a *general* framework that could handle any colors in the game, we hypothesize that the experimental students in the pre-Lightlet training would be more likely to do so, especially since they were not only explicitly taught to do so but also greatly benefited from generating a general explanation for both physics tasks. So our first hypothesis is that: *the experimental students will be more likely to use Catalog charts than the control students.*

5. Validating Choice-based Assessments

In the normative world of education, we care about "better" and "worse," so it is crucial to justify whether or not a particular performance is "better." Knowledge-based assessments rely on objective "right" and "wrong" answers as their criteria for better and worse performance. Few would argue with the claim that "five" is a worse answer than "four" to the question, "What is two plus two?" But with choices, people may reasonably challenge the judgment that one choice is better than another. Who is to say that persisting is better than not? If we were to analyze the log file of a student using Lightlet, for example, the choice of where to let the cursor rest while thinking is less relevant than the choice of whether to open up one of the color charts. A data-driven answer would help alleviate some of the problems associated with the social construction of what constitutes a useful choice, at least with respect to learning. People would then be able to debate whether the learning value of a choice is sufficiently high to favor it in assessment.

Here we present some approaches to validating whether some choices are better or worse than others. As always, our criteria of better and worse are relative to learning. We begin with a correlational approach: whether certain choices are connected to standard knowledge-based measures. This is important since most educators still think knowledge-based assessments are the ground truth for evaluating learning. In the following, we used the students' final school math test scores as a *standard* knowledge-based assessment. More specifically, we will investigate whether the choices students made when interacting with Lightlet would predict their final school math test scores.

To further validate the choice-based assessment, we will directly compare the choice-based assessment with *game-embedded* knowledge-based assessments such as how well did a student play Lightlet. Thus our general question is: which assessment is predictive to students' school performance, the choice-based assessment, the *embedded* knowledge-based assessment, or neither. Given that Lightlet has little to do with solving math problems as they appear on the children's mathematics tests, we hypothesize that that *game-embedded* knowledge-based may not be very predictive.

As described above: we hypothesized that students involved in Catalog-related choice patterns are interested in finding a *general* framework that could handle any colors in the game. Since the Catalogs only became available on the full game play level (level 2), we expect general-explanation students would begin exploring the Catalog choices on level 2, immediately or shortly after each catalog becomes available. Therefore, our second hypothesis is that: *when considering level 2 alone the choice-based assessment will be a better predictor of students' math performance than the game-embedded knowledge-based assessment.*

On the other hand, considering level 2 alone may not be sufficient to grasp all the students' choice patterns. On average, students spent only around 5 minutes on level 2 vs. 20 minutes on the whole game. So the effectiveness of choice-based assessment may become even more predictive if we considering the entire game play. Therefore, our third hypothesis is that *when considering the whole game play, the choice-based assessment will still be a better predictor for students' math performance than the game-embedded knowledge-based assessments.* However, it is not clear whether *the former will be more effective than the choice-based assessment when using level 2 alone.* In other words, whether the longer the choice-based assessment is, the more effective it will be?

6. Data Collection

6.1 Participants and Design:

Two 6th-grade classes participated in the study. Both classes were from high-SES schools in California and had the same math teacher. Due to logistical constraints, intact classes were randomly assigned to the two conditions during pre-Lightlet training. The assignments were: Experimental (n = 19) and Control (n = 21). In both conditions, students variously worked individually or in groups, consistent with regular classroom practice. Then all students played Lightlet for 15-30 minutes. All tests in the study were taken individually.

6.2 Procedure:

The study occurred on three Friday classes (50 min each) with two consecutive Fridays for the pre-Lightlet training and one for interactions with Lightlet.

During the pre-Lightlet training, all students were given a set of contrasting cases on "cannon rides" shooting straight out at 0° angle at different speeds and from different heights in the first week and cases on "cannon rides" shooting out at an angle at different speeds and reaches different maximum heights in the second week. The treatment difference occurred in the instructions that students received. Control students were prompted to explain the similarities and differences among the cases while the experimental ones were told to invent a single general framework that would explain all cases. This phase lasted 15 minutes. Students answered a brief test item, used the simulation to test their ideas, and then took the posttest.

Interactions with Lightlet happened six weeks after pre-Lightlet training. All students played Lightlet for fifteen to thirty minutes. The students' final school math test was taken at the end of the semester, about one month after interacting the Lightlet.

6.3 Data Features

In order to identify students' choices and track their game-embedded performance during interactions with Lightlet, we defined a set of Catalog-related, Experiment Room related, and performance-related features based on a combination of theory and prior work on modeling learning environments (Chi, VanLehn, Litman, & Jordan, 2011) and on student modeling (Chi, Koedinger, Gordon, Jordan, & VanLehn, 2011). We do not yet know precisely what choice or performance actions are associated with learning outcomes in advance. Thus, we defined four categories of features.

The first two categories correspond to the two types of the choice patterns in section 3. More specifically:

The first category includes 18 features related with the Catalog usage. It includes two types of features: duration and occurrence. The former covers 11 time-related features such as the total duration that a correct or incorrect Catalog is open; while the latter includes seven features such as the total number of times that a student opened an incorrect Catalog, the total number of Catalogs that the student opened and so on.

The second category includes 13 features related with usage of the Experiment Room. It consists of one feature covering how much time a student spent in the Experiment Room and 12 features related to their behaviors within the room behaviors. These are simple features such as the number of times the students used the Experiment Room and more complicated features such as the number of times that a student successfully makes a target color in the experiment room after picking it on the game board.

The third category includes 6 features focused on general information about the game play. These include simple features such as the total time spent in the game, gender and the condition information during the Pre-Lightlet training. Some more complicated features in this category assess how students choose the next puzzle to play. On Lightlet, students were given 6, 9 and 12 puzzles on level 1, 2 and 3 respectively. For each level, students can select which tile to play in any order. We thus defined three features to detect how the students choose the target. We found that, rather than using the Experiment Room, students sometimes engage in trial and error on the game board. One example feature is TryErrorPick, which is defined as the number of times a student picked a color from the game board that they had previously made a wrong attempt on. For example, if a student trying to mix "orange" by mixing red + green, the student would get yellow and the orange puzzle remains unsolved on the game board; if the student then chose yellow again this would be detected.

The last category includes 13 features related with game-embedded performance-related features. They include features such as the total number of tiles that a student succeeded, the percentage of tiles that a student succeeded (how well a student clear the game board), how efficient a student was when clearing the tiles (the number of tiles students cleared from game board divided by the total time) and so on.

7. Results

In the following, we will present our results in the order of our three hypotheses:

> **Hypothesis 1:** *the experimental students will be more likely to use Catalog charts than the control students*.
>
> **Hypothesis 2:** *when considering level 2 alone the choice-based assessment will be a better predictor for students' math test school than the game-embedded knowledge-based assessment.*
>
> **Hypothesis 3:** *when considering the whole game play, the choice-based assessment still be a better predictor than the game-embedded knowledge-based assessments.*

7.1 Experiemental vs. Control

Our overall results show that the experimental and control groups were comparable at the outset of Pre-Lightlet training when the treatment differences began: there were no significant differences between treatment groups on two tests given by the teacher before the study, or any of our assessments before or at pretest on the first week of Pre-Lightlet training. As expected, after the different treatments, the two groups began to separate. We found that explicitly asking students to generate a general explanation led the experimental group to outperform the control condition on several midtest and posttest items after the different treatments took place. More specifically, 100% and 77.8% of experimental group produced a general explanation for the two physics tasks respectively while only 10% and 33.3% of the control group did so. This difference was statistically-significant for the first task and marginally-significant for the second.

The experimental group significantly outperformed the control group. We argue that this is because the former were asked explicitly to generate a general explanation for all the cases. On Lightlet, to generate a general framework for all the color games would requires students to engage in Catalog activities. We would expect that the experimental group would be more likely to do so. However, our results showed that the experimental students were no more likely to engage in Catalog activities than the control students. We compared the two groups across all Catalog -related features described in previous section on both level 2 and across the whole game. No significant difference was found on any of Catalog-related choices students made.

Furthermore, we found no significant difference between the two conditions on any Lightlet Experiment Room-related behaviors, game-board performance related, or the school math final test scores.

Overall, it seems that after explicit instruction on generating general explanations, the experimental students did not spontaneously make a choice that would lead to finding a general solution for all the colors on Lightlet. There are many possible explanations for this finding. One possible explanation is that the instruction on generating a general explanation was explicit during the Pre-Lightlet training but when interacting with Lightlet, the experimental students were not explicitly asked to do so. Additionally, the experimental students were given a set of comparing and contrasting worked cases in the original general explanation instruction; but on Lightlet, they were not given any.

To summarize, our results showed that it is still an open question how to teach students to make good choices. On the other hand, while it may not be easy to teach students to make good choices, is it still feasible to use choices as an effective assessment? In the following, we investigated whether the choices students made on Lightlet would predict their learning performance in school. We first investigated whether using level 2 data alone would be predictive.

7.2 Choice- vs. Knowledge-based Assessment Using Level 2 Only

We first investigated whether the individual features from the four categories would predict the standard knowledge-based assessment: students' final math test scores. Note that, all the features were calculated based on the level 2's log files alone. Among the four categories, Out of 18 Catalog-related features, 13 features are significantly predicted students' final math test scores. For all 13 features, the more Catalog activities, the higher the students' final math tests scores. Among them, the most predictive feature is: *ResourceReviewDuration* the total duration that a Catalog is open in level 2. *ResourceReviewDuration* significantly predicted students' final math tests scores: $\beta = 0.010$, $t(38) = 2.64$, $p = 0.01$. It alone also explained a significant proportion of variance in students' final math tests scores, $R^2 = .15$, $F(1, 38) = 6.95$, $p < .001$.

Only one out of 13 Experiment-Room related features are significantly predicted students' final math test scores. The feature is GoalOrientedExperimentSuccessTry: the number of times students pick a color from the game board and then try to make the targeted color successfully in the Experiment Room in level 2. GoalOrientedExperimentSuccessTry significantly predicted students' final math tests scores: $\beta = -1.93$, $t(38) = -2.69$, $p = 0.01$. It alone explained a significant proportion of variance in students' final math tests scores, $R^2 = .16$, $F(1, 38) = 7.22$, $p = .01$. So the more a student try and error successfully in the experiment room, the lower his/her final math score is.

Finally, none of the features in the remaining two categories, the general information and the game-embedded performance related features significantly predict students' final math test scores. For example, PercCorrectGamePlay: the total number of times tile a student succeeded divided by the total number of tiles the student tried to play in level 2, did not significantly predict student's final math scores: $R^2 = .05$, $F(1, 38) = 1.85$, $p = .18$.

Therefore, when considering level 2 alone the choice-based assessment is a better predictor for students' math test school than the game-embedded knowledge-based assessment when using the single feature.

We then applied brute-force search to select the best three features from all the four categories that would best predict students' final math test scores. Our final model include three features and they are:

> WrongResourceReviewDuration (Catalog-related): The total duration of a student opening a wrong Catalog
>
> GoalOrientedExperimentSuccessTry (Experiment Room related): The number of times students pick a color from the game board and then try to make the targeted color successfully in the Experiment Room.
>
> TryErrorPick (General): The number of times students pick a color from the game board that is the same color as the previous wrong color.

The results of the regression indicated the three predictors explained a significant proportion of variance in students' final math tests scores: R^2 = .43, F(3, 36) = 8.94, p = .0001. Table 1 shows that all three features significantly predict students' final math tests scores.

Table 1: Coefficient of Three Level 2 Feature Model

Feature Name	β	Sig
WrongResourceReviewDuration	0.018	$t(38) = 3.51$ $p = 0.001$
GoalOrientedExperimentSuccessTry	-2.62	$t(38) = -4.14$ $p = 0.0002$
TryErrorPick	-3.28	$t(38) = -2.69$ p <0.05

Finally, note that none of the game-play performance related features were included in the final three-feature model. Therefore, it again suggested that the choice-based assessment is a better predictor for students' math test school than the game-embedded knowledge-based assessment.

7.3 Choice- vs. Knowledge-based Assessment Across Levels

Similar as previous section, we first investigated whether each individual features we extracted from whole log files would predict students' final math test scores.

While on level 2, 13 out of 18 Catalog-related features are significantly predicted students' final math test scores, only 5 Catalog-related features are significantly predictors when using across levels. Among them, the most predictive feature is the same as using level 2: ResourceReviewDuration (the total duration that a Catalog is open). It significantly predicted students' final math tests scores: $\beta = 0.004$, t(38) = 2.34, p = 0.02. It alone also explained a significant proportion of variance in students' final math tests scores, R^2 = .13, F(1, 38) = 5.46, p= 0.02. So when using the whole game play logs, the best predictive Catalog-based feature is still the same as using the level 2 alone: ResourceReviewDuration. However, when considering the whole game play, the ResourceReviewDuration is less predictive than using the level 2 data alone.

Similarly, out of 13 Experiment-Room related features, the only feature that significantly predicted students' final math tests scores is again: GoalOrientedExperimentSuccessTry (the number of times students pick a color from the game board and then try to make the targeted color successfully in the Experiment Room). It significantly predicted students' final math tests scores: $\beta = -0.45$, t(38) = -2.45, p = 0.02. It alone explained a significant proportion of variance in students' final math tests scores: R^2 = .14, F(1, 38) = 5.995, p = .02. Again, when considering the whole game play, the GoalOrientedExperimentSuccessTry is less predictive than using the level 2 data alone: R^2 = 0.136, p < 0.02 vs. R^2 = 0.16, p = 0.01 respectively.

For the remaining three types of features, as when using the level 2 log alone, none of the features significantly predict students' final math test scores. In other words, again none of the embedded knowledge-based assessment on students' game play performance significantly predict students final math test scores. For example, PercCorrectGamePlay, the total number of times tile a student succeeded divided by the total number of tiles the student tried to play across the whole game, again did *not* significantly predict student's final math scores: R^2 = .02, F(1, 38) = 0.88, p = .35.

As with using level 2 alone, the choice-based assessment is a better predictor for students' math test school than the game-embedded knowledge-based assessment when using the single feature.

The brute-force search selects the best three features from all the four categories that would best predict students' final math test scores. Two features, WrongResourceReviewDuration (Catalog) and GoalOrientedExperimentSuccessTry (Experiment Room), are also shown in best three-feature model using Level 2 log alone; the other feature is:

> ExactDurationNoActivity (General): The total duration that a student is not involving any game playing activities such as reading Catalog, nor using Experiment Room, nor playing a game.

The results of the regression indicated the three predictors explained a significant proportion of variance in students' final math tests scores: R^2 = .34, F(3, 36) = 6.27, p < .002. All three features significantly predict students' final math tests scores (Table 2)

Table 2: Coefficient of Three Across Level Feature Model

	coeff	Sig
WrongResourceReviewDuration	0.008	$t(38) = 3.05$ $p = 0.004$
GoalOrientedExperimentSuccessTry	-0.55	$t(38) = 3.20$ $p = 0.003$
ExactDurationNoActivity	-0.005	$t(38) = 2.25$ $p = 0.03$

In addition to the fact that both WrongResourceReviewDuration and GoalOrientedExperimentSuccessTry showed up in the best predicted models when considering level 2 alone and when considering the whole game play, in both models the former is positively correlated with students' school math performance while the latter is negative correlated. Additionally, note that the best model when considering level 2 alone beat the best model across level: R^2 =0.43 vs. R^2 =0.34.

When using the same three best features used in the level 2's best model to predict students' final math tests scores, the model is still significantly predict the student's school performance: R^2 = .25, F(3, 36) = 4.01, p = .01. We found that WrongResourceReviewDuration significantly predicted students' final math test scores ($\beta = 0.006$, p = 0.02), as did GoalOrientedExperimentSuccessTry ($\beta = -0.44$, p = 0.02), but not TryErrorPick, (p = 0.95)

Again, note that none of the game-play performance related features were included in the final best three-feature model. This again suggests that when considering the whole game play, the choice-based assessment still be a better predictor than the game-embedded knowledge-based assessments.

Furthermore, our results also suggested that the choice-based assessment when using level 2 alone is more predictive than the choice-based assessment using the whole game play. So it suggested that it is important to note that for certain skills, the choice-based assessment should be nimble. 5 minutes on Lightlet is more efficient to detect effective learners than asking student to spend 20 minutes on it.

Finally, we have shown that choice-based assessment is more predictive than game-embedded knowledge-based assessment. One more interesting question to answer is: did the choice student made during the game play help their performance in the game? There is insufficient space in this paper to go into detail. But our overall finding is that the choice student made during the game play indeed significantly predicts their performance in the game.

For this analysis, we treat the third level of Lightlet as a posttest for level 2. If students make good learning choices on level 2 and learn about additive color, then they should do well on level 3. It turns out that the same choice pattern in level 2 that predicted learning in students' math class also predicted performance on level 3 (how efficient they clear the game board on level 3): $R^2 = .32$, $F(3, 36) = 5.61$, $p = .003$. We found that only one feature WrongResourceReviewDuration significantly predicted student's level 3's performance ($\beta = -0.19$, $p = 0.0004$) while the other two were not significantly predictive: TryErrorPick, ($p = 0.10$) GoalOrientedExperimentSuccessTry ($p = 0.15$).

To summarize, students who chose to explore the Catalog were more likely to do better in school. In fact, the amount of time students committed to figuring out the Catalog entries shortly after the Catalog became available predicted about 43% of the variation in the students' grades in their mathematics classes (the only classes for which we had records). In other words, all students tried to level-up in our game, but those who chose not to engage in critical thinking while doing so were also doing worse in school.

8. Conclusions

For many, assessments are a lighthouse in the fog of education—a clear guide by which to make safe decisions. But in reality, assessments create the fog. Current assessments perpetuate beliefs that the proper outcomes of learning are static facts and routine skills—stuff that is easy to score as right or wrong. Interest, curiosity, identification, self-efficacy, belonging, and all the other goals of informal learning cannot even sit at the assessment table, because these outcomes are too far removed from current beliefs about what is really important.

Assessments seem to be built on the presupposition that people will never need to learn anything new after the test, because current assessments miss so many aspects of what it means to be prepared for future learning. These frozen-moment assessments have influenced what people think counts as useful learning, which then shows up in curricula, standards, instructional technologies, and people's pursuits.

Teachers may tell students about the importance of persistence, critical thinking, interest development, and a host of other keys to a successful life. But tests provide the empirical evidence that students use to decide what is truly valued. If an assessment focuses on the retrieval and procedural application of narrow skills and facts, this is what students will think counts as useful learning. By changing assessments to concentrate on choices, we should be able to improve beliefs about what constitutes useful learning.

If the fog were lifted, we would see that most of the stakeholders in education care first and foremost about people's abilities to make good choices. Making good choices depends on what people know, but it also depends on much more, including interest, persistence, and a host of twenty-first-century soft skills that are critical to learning. Where we can anticipate a stable future—decoding letters into words is likely to be a stable demand for the next fifty years—then knowledge- and skill-based assessments make sense. In relation to those aspects of the future that are less stable, though, people will need to choose whether, what, when, and how to learn. Hence, it is important to focus on choices that influence learning, and assessments should measure those choices. Choice is the critical outcome of learning, not knowledge. Knowledge is an enabler; choice is the outcome.

Assessing choices during learning has a number of attractive properties. Foremost, choice-based assessments are process oriented. They examine learning choices in action rather than only the end products. This process focus makes it possible to connect the learning behaviors during the assessment to processes that occur in a learning environment. Second, the assessments reveal what students are prepared to learn, so they are prospective as opposed to retrospective. Third, choice resonates with the rest of the social sciences that examine the movements of people, money, and ideas. Fourth, choices do not lend themselves to simplistic reifications whereby things like people's knowledge or personality traits are misinterpreted as independent of context and immune to change. Fifth, choices can measure a much greater range of learning outcomes than fact retrieval and procedural application. Sixth, learning choices are a good candidate for inclusion in standards, which currently define what knowledge students should have but stay strangely silent about the processes of learning themselves.

Recent advancements in technology create a special opportunity for moving toward a new paradigm of assessment. There are risks, however. People may only use technology to make us faster and more entrenched in doing the wrong thing. When used well, technology makes it possible to create and validate choice-based assessments by using the rapid generation of interactive environments, crowdsourcing, automated logging, and educational data mining. Thus, it is possible for choice to become the core of assessment (and not in the degraded sense of multiple-choice tests). In this paper, we provided an anchoring example of a computerized, choice-based assessment, Lightlet.

Tracking the process of learning is different from simply detecting whether a student knows an answer or not, which is the output of most tests. Choice-based assessments can provide a much richer corpus of information from which to draw actionable information about learners. We can locate the source of the problem rather than just the consequence. The students who were doing the worst in math class, for instance, were those who used the Experiment Room to solve each problem through trial and error. These students, instead of trying to develop an overall understanding of additive color, were simply attempting to get the right answer for each problem in turn. In the best case, identifying this pattern of choices can help a teacher address the underlying learning issue, which is that the students are trying to solve each problem in turn rather than discovering the general principle that governs the solutions to all problems.

In an initial study using a similar environment with sixth grade children, the results were quite clear. Children who chose to look at the Catalog of charts during the game were doing better in school. In fact, the different choice patterns students committed during the game play predicted about 43 percent of the variation in the students' grades in their mathematics classes. While all the students seemed happy to play the game, those who chose not to engage in critical thinking were also the students who performed worse in mathematics. The 43 percent level of prediction is high, especially considering that Lightlet has little to do with solving math problems as they appear on the children's mathematics tests. The assessment captured something crucial about how these children go about learning that is affecting their success in mathematics—and will likely do so in the future.

Overall our results offer two take-home messages. First, by assessing students' choices during game playing, we can discover a great deal about the processes they do or do not use to learn. Second, we can assess choices that are critical to learning, but that are missed by most tests.

To summarize, with more choices and interactivity comes more information about the learner. Performance assessments, such as portfolio and project-based assessments, have tried to capitalize on the increased information found in choice-rich environments (e.g., Resnick and Resnick 1994). Richard Shavelson, Gail Baxter, and Jerome Pine (1991), for example, describe a kit-based performance assessment for science. Students conduct physical experiments to determine which brand of paper towel absorbs more water. The assessment provides information about the students' abilities (or inclinations) to use experimental logic and take careful measurements. Unfortunately, the authors also point out that performance assessments can be prohibitively expensive to deploy and score at scale. Technology can help overcome the difficulties associated with increased information. Computers can deliver assessments where students make choices about how to learn, and the computers can automatically log all user behaviors that might be of interest to a teacher, assessor, or researcher, ranging from chat logs to virtual interpersonal distance to direction of gaze. It is an ethnographer's thick description for free. Computers provide new efficiencies that make tractable what was once impracticable. And with new empirical capabilities, new theories are sure to follow.

People generally in system performance measure for predictive analysis. Our research, however, shows that behaviors features can be far more informative

9. REFERENCES

[1] Aleven V., Koedinger, K. R., & Popescu, O. (2003). A Tutorial Dialog System to Support Self-Explanation: Evaluation and Open Questions. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), Proceedings of the 11th International Conference on Artificial Intelligence in Education, AI-ED 2003 (pp. 39-46). Amsterdam, The Netherlands: IOS Press. Finalist for Conference Best Paper Award, AIED 2003

[2] Baker, R.S., Corbett, A.T., Koedinger, K.R. (2004) Detecting Student Misuse of Intelligent Tutoring Systems. Proceedings of the 7th International Conference on Intelligent Tutoring Systems, 531-540.

[3] Bransford, J. D., and D. L. Schwartz. 1999. "Rethinking Transfer: A Simple Proposal with Multiple Implications." Review of Research in Education 24:61–100.

[4] Chi, M., Koedinger, K. R., Gordon, G., Jordan, P. W., & VanLehn, K. (2011). Instructional Factors Analysis: A Cognitive Model For Multiple Instructional Interventions. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, & J. C. Stamper (Eds.), Proceedings of the 4th International Conference on Educational Data Mining, Eindhoven, The Netherlands, July 6-8, 2011 (pp. 61–70).

[5] Chi, M., VanLehn, K., Litman, D. J., & Jordan, P. W. (2011a). Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. User Model. User-Adapt. Interact., 21(1-2), 137–180.

[6] Doris B. Chin, Ilsa M. Dohmen, Britte H. Cheng, Marily A. Oppezzo, Catherine C. Chase, and Daniel L. Schwartz. (2010) Preparing students for future learning with Teachable Agents. Educational Technology Research & Development.

[7] Feuerstein, R. 1979. The Dynamic Assessment of Retarded Performers: The Learning Potential Assessment Device, Theory, Instruments, and Techniques. Baltimore, MD: University Park Press.

[8] Gee, J. P. 2003. What Video Games Have to Teach Us about Learning and Literacy. New York: Palgrave.

[9] Hogyeong Jeong, Gautam Biswas (2008) Mining Student Behavior Models in Learning-by-Teaching Environments, 127-136. In The 1st International Conference on Educational Data Mining.

[10] Iyengar, S. S., and M. R. Lepper. 1999. "Rethinking the Value of Choice: A Cultural Perspective on Intrinsic Motivation." Journal of Personality and Social Psychology 76:349–366.

[11] Schwartz, D. L., and J. D. Bransford. 1998. "A Time for Telling." Cognition and Instruction 16:475–522.

[12] Schwartz, D. L., and T. Martin. 2004. "Inventing to Prepare for Learning: The Hidden Efficiency of Original Student Production in Statistics Instruction." Cognition and Instruction 22:129–184.

[13] Practicing versus inventing with contrasting cases: The effects of telling first on learning and transfer. Daniel L. Schwartz, Catherine C. Chase, Marily A. Oppezzo, & Doris B. Chin. I (2011). *Journal of Education Psychology.*

[14] Stevens, R. H., & Thadani, V. (2007). Quantifying students' scientific problem solving efficiency and effectiveness. Technology, Instruction, Cognition and Learning, 5(4), 325-337.

[15] Vygotsky, L. S. (1934) 1987. The Collected Works of L. S. Vygotsky, ed. R. Rieber and A. Carton. New York: Plenum.

Comparing Expert and Metric-Based Assessments of Association Rule Interestingness

Diego A. Luna Bazaldua
+1 (917) 543 7674
dal2159@tc.columbia.edu

Ryan S. Baker
+1 (412) 983-3619
baker2@exchange.tc.columbia.edu

Maria Ofelia Z. San Pedro
+1 (508) 330-0410
mzs2106@tc.columbia.edu

Department of Human Development, Teachers College, Columbia University
525 W. 120th Street, New York, NY 10027

ABSTRACT

In association rule mining, interestingness refers to metrics that are applied to select association rules, beyond support and confidence. For example, Merceron & Yacef (2008) recommend that researchers use a combination of lift and cosine to select association rules, after first filtering out rules with low support and confidence. However, the empirical basis for considering these specific metrics to be evidence of interestingness is rather weak. In this study, we examine these metrics by distilling association rules from real educational data relevant to established research questions in the areas of affect and disengagement. We then ask three domain experts to rate the interestingness of the resultant rules. We finally analyze the data to determine which metric(s) best agree with expert judgments of interestingness. We find that Merceron & Yacef (2008) were right. Lift and cosine are good indicators of interestingness. In addition, the Phi Coefficient, Convinction, and Jaccard also turn out to be good indicators of interestingness.

Keywords

Association Rules, Interestingness, Cosine, Phi Coefficient, Human Rating

1. INTRODUCTION

In recent years, Association Rule Mining has become a central method in the field of Educational Data Mining. It plays a prominent role in reviews of the field, including reviews by Romero & Ventura (2007, 2010), Baker & Yacef (2009), Scheuer & McLaren (2012), and Baker & Siemens (in press), referred to this method as a core type of relationship mining. In Association Rule Mining, algorithms search for patterns where a set of values of variables (the “if-clause”) predict another variable’s value (the “then-clause”). (It is also possible for a then-clause to have multiple variables, but less common).

In these reviews, it was noted that Association Rule Mining has several potential applications. It is excellent for generating hypotheses to study further, and for finding unexpected connections within data.

Association Rule Mining has been applied to several applied research problems within the educational data mining community and related research communities. Some notable examples include: Freyberger and colleagues have used association rules to analyze interactions between students and intelligent tutoring systems, in order to find models that predict student's success (Freyberger, Heffernan & Ruiz, 2004); Lu (2004) used association rules to match suitable learning materials based on each student learning needs; Garcia, Romero, Ventura & De Castro (2009) have used association rules to make recommendations to instructors for how to improve the effectiveness of a web adaptive course; in a similar example, association rules have been implemented to provide information to teachers about students’ behavior in intelligent tutoring systems (Ben-Naim, Bain & Marcus, 2009).

A subset of Association Rule Mining, Sequential Pattern Mining, has also seen extensive use in the educational data mining community, as well as being highlighted in reviews of the field (e.g. Romero & Ventura, 2007; Baker & Yacef, 2009; Scheuer & McLaren, 2012; Baker & Siemens, in press). Sequential Pattern Mining consists of finding association rules where the contents of the then-clause occur temporally after the contents of the if-clause (Agrawal & Srikant, 1995). In the case of educational data mining, Kinnebrew, Loretz, & Biswas (2012) have used Sequential Pattern Mining to analyze how students engage in the different activities within an intelligent tutoring system over time, in particular studying the different sequences seen in high-performing and low-performing students. In another example, Perera et al. (2009) used Sequential Pattern Mining to analyze how groups of students use online tools, studying the work patterns of successful and unsuccessful groups, in order to provide feedback to the groups about their work strategies. One more example in education comes from the research done by Romero, Ventura, Delgado & De Bra (2007), who integrated Sequential Pattern Mining techniques in an algorithm within an educational system in order to provide personalized recommendations to students about possible links they should explore.

Association rules are typically initially selected on the basis of rules’ confidence and support (Agrawal & Srikant, 1995). The support of a rule corresponds to the percentage of data points that contain both the if-clause and then-clause. The confidence of the rule is expressed as the percentage of data points that contain both the if-clause and also includes the then-clause, divided by the number of data points that contain the if-clause (Garcia, Romero, Ventura & Calders, 2007).

However, the combination of support and confidence is insufficient to select good association rules. By definition, support and confidence find variable values that are frequently seen together. As such, these metrics often end up selecting combinations of variable values that are trivially associated, such as finding that students who take advanced biology probably took introductory biology, or finding that students who fail a course's exams fail the course as well.

What is desirable is to instead find association rules that are novel, that are surprising, that are unexpected. Frequently, after rules are filtered by looking for all rules with a minimum support and confidence, the next step is to use an alternate metric that can give some indicator of novelty; that can determine if an association rule is *interesting*.

To this end, researchers have tried to decide which metrics best capture an association rule's interestingness, both in general (Tan, Kumar & Srivastava, 2004), and in the specific case of educational data mining (Meceron & Yacef, 2008). Merceron and Yacef (2008) recommend Lift/Added Value (Lift and Added Value are mathematically equivalent) and Cosine as excellent interestingness measures for educational data because their meaning is easily understood even to people not expert in data mining (e.g., teachers, school administrators, and so on); in addition, Cosine does not depend on the data set size. In particular, they recommend that researchers consider an association rule to be interesting if it has a high value for either of these measures.

Moreover, there are additional metrics identified that have the potential to measure interestingness. Tan et al. (2004) review the potential candidates for an interestingness measure, finding over twenty in the published literature. Their list includes lift and cosine, but also includes the Phi coefficient, Goodman-Kruskal's, the Odds ratio, Yule's Q, Yule's Y, Cohen's Kappa, Mutual information, the J-Measure, the Gini Index, Laplace, Conviction, Piatetsky-Shapiro, Certainty Factor, Added Value, Collective strength, Jaccard, and Klosgen. Such variety of possible interestingness measures has made it complicated to identify which one is the most appropriate.

Further complicating the matter of choosing an appropriate interestingness measure (or measures) is the fact that the research on interestingness measures has thus far been mathematical or intuitive: interestingness measures have been selected based on their mathematical properties, and in some cases based on the intuitive perceptions of expert data miners.

In this paper, we consider an alternate strategy for selecting interestingness measures: using data mining to determine which interestingness measure is best, based on expert judgments of interestingness. In other words, instead of selecting a metric formally or intuitively, we can actually collect data on which association rules are seen as being the most interesting by domain experts, the population that could best take advantage of new hypotheses and unexpected findings in a domain. We then analyze this data to determine which metrics, or combination of metrics, best matches the domain experts' perception of specific rules' interestingness.

In the following sections, we take real data from online learning. We then distill association rules for that data relevant to established research questions in the field. We then ask three domain experts to rate the interestingness of the resultant rules. We finally analyze the data to determine which metric(s) best agree with expert judgments of interestingness. In doing so, we will explicitly compare our findings to claims in Merceron & Yacef (2008) as to which metrics best represent interestingness.

2. Method

2.1 Data

In order to study domain experts' assessments of which association rules are interesting, we generated association rules from real student data, relevant to established research questions in the field. We use domain experts, under the hypothesis that what experts consider interesting may be different than what novices consider interesting (and we believe that finding rules that are interesting for an expert is a more valuable use of association rule mining, though opinions could differ). We use genuine data to create these rules rather than simulated data, due to the concern that the metrics that predict the interestingness of genuine data may not be the same as the metrics that predict interestingness in simulated data. This would be a particular concern if the simulated data were to produce association rules that were actually false; and using generic operators would eliminate the potential to leverage domain expertise.

To this end, we used models that assess student affect and disengaged behaviors within a widely-used online learning environment, to examine association rules about the relationships between student´s affect and disengaged behaviors. The study of student disengagement and affect has been a research topic of considerable interest to researchers in EDM and related fields. Sabourin, Rowe, Mott, & Lester (2011) have analyzed the relation between engaged and disengaged behaviors with positive and negative affective states in students while interacting with a learning system, finding that different patterns of affect correlate to engaged and disengaged behaviors. Hershkovitz, Baker, Gobert, & Nakama (2012) have found evidence that boredom mediates between the student´s tendency to avoid novelty and off-task behavior. Baker, D'Mello, Rodrigo & Graesser (2010) find that gaming the system is often preceded and followed by boredom. Chauncey & Azevedo (2010) show a relationship between induced affect and cognitive engagement/meta-cognition, leading to differences in performance.

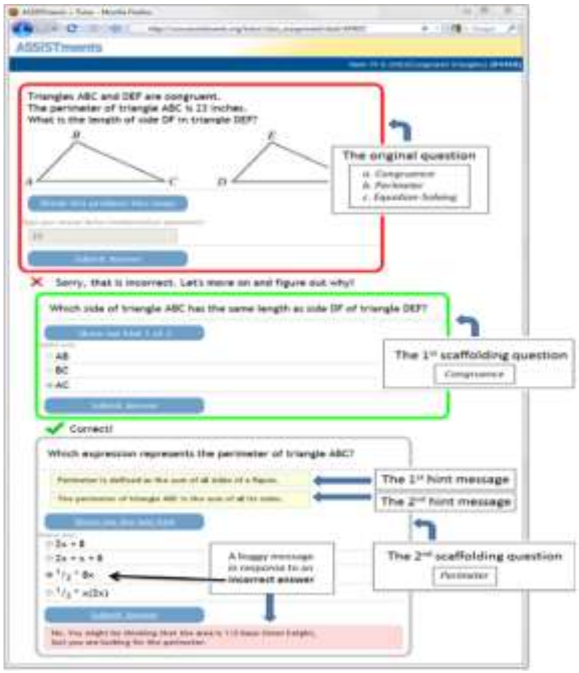
These rules were generated from data from the ASSISTments system (Razzaq, Heffernan, Feng & Pardos, 2007). ASSISTments is an educational web-based system that provides students with intelligent tutor-based online problem solving activities, while providing teachers with dynamic formative assessment of the students' mathematical abilities. The system has been found to be effective at enhancing student learning. (Razzaq et al., 2007), and is used by over 50,000 students a year. Figure 1 shows a screen shot of the ASSISTment system.

Data was obtained from the logs of 724 middle school students from the Northeastern United States, who answered different problems that measure 70 different mathematics skills. Within this data set, there were a total of 107,382 problems solved by students within the ASSISTment software. Student actions in this data set were classified in terms of affective states and disengaged behaviors from machine-learned affect and behavior detectors. The detectors inferred if the student:

- was detected as being bored or not,
- was detected as being concentrated or not,
- was detected as being frustrated or not,
- was detected as being confused or not,
- was detected as being on task or off task,
- was detected as being gaming the system or not,

The following additional features were also included in the data set:

- the student providing a correct answer
- the student providing an incorrect answer
- the student asking for a hint.


ASSISTments

Triangles ABC and DEF are congruent.
The perimeter of triangle ABC is 23 inches.
What is the length of side DF in triangle DEF?

Sorry, that is incorrect. Let's move on and figure out why!

Which side of triangle ABC has the same length as side DF of triangle DEF?

- AB
- BC
- AC

Correct!

Which expression represents the perimeter of triangle ABC?

Figure 1. Example of an ASSISTment item

The detection of these binary categories of affective states/behaviors was done using the detectors presented in Pardos et al (2013). These detectors were developed by distilling features of the students´ interactions with the software, and synchronizing those features with field observations collected by two trained coders during the students' interactions with ASSISTments. The log data entry and the field observations were synchronized and segmented in 20 second windows to develop the detectors.

Detector performance was evaluated using student-level cross-validation (5-fold). All detectors performed substantially better than chance, being able to distinguish each affective state/behavior between 63%-82% of the time (the A' statistic), performance that was 23%-51% better than chance (the Kappa statistic). The detectors provide confidence values of the probability that an affective state or behavior occurred. To support the association rule mining analyses discussed below, we convert these probabilities into binary predictions, using a 50% probability threshold (the Kappa values listed above represent the model goodness when this transformation is used). Pardos et al. (2013) and San Pedro et al. (2013) provide a detailed description of the detectors and their use in multiple discovery with models analyses. Table 1 summarizes the frequency and proportion of each of these behaviors/affective states. Regarding table 1, it shows some of the average confidences are higher than what should be expected. Here we point out that, as it is indicated in San Pedro et al. (2013), some detectors used in the current research presented some systematic error in prediction, which impacted in a higher or lower average confidence of the resultant models compared to the proportion of the affective states in the original data set. This type of bias does not affect correlation to other variables since relative order of predictions is unaffected, neither affects A' or Kappa, but it can reduce model interpretability. We did not rescaled the detectors, as it is proposed in Pardos et al. (2013) since we are considering final binary predictions from the detectors, where Kappa is the relevant goodness statistic, we use non-rescaled confidences in this paper.

The association rules were created in way that each rule described how a set of the affective states/ behaviors seen in the first attempt at a problem was associated with a single affective state or behavior in the student's first action on the next problem. In this analysis, simple association rules were created that predicted affect or behavior from a combination of the elements at the previous action.

2.2 Generation of Association Rules

Association rules were created using the arules package (Hahsler, Gruen, & Hornik, 2005; Hahsler et al., 2009) in R version 2.15.2 (R Development Core Team, 2012). In specific, the apriori algorithm implemented within the arules package was used to discover the association rules (Agrawal et al., 1994). This process in R resulted in a list of 431,768 rules, for which support, confidence, and lift were automatically computed. A total of 120 different association rules were selected from the 431,768 measures obtained; these 120 rules were selected to be the rules with the highest support and confidence that were representative of different numbers of elements in the if-clauses and were representative of all variables in the then-clauses of the rules. All rules selected had a support over 0.05 and confidence over 0.1; most were considerably higher.

Table 1. Frequency and average confidence for each affective /behavioral state in the data

	Frequency	Percentage	Rescaled Average Confidence
Bored	52080	48.49	0.2469
Engaged concentration	47854	44.56	0.5160
Frustrated	10929	10.17	0.0988
Confused	20308	18.91	0.1372
Off-Task	18135	16.88	0.0406
Gaming the system	9805	9.13	0.0182
Used Hints	16216	0.15	
Answer was Correct	45116	0.42	

2.3 Expert Rating of Association Rules

Once the rules had been created, they were rated for their interestingness by domain experts. In specific, four scientific researchers with scientific expertise in the areas of affect and disengagement in online learning. They rated the extent to which each of the 120 association rules was "scientifically interesting".

A Likert scale was used in rating, ranging from 1 to 5, where 1 was "*Not at all interesting*" and 5 was "*Extremely interesting*". Based on these expert ratings, the average inter-rater interestingness value was calculated for each rule, giving an indicator of how interesting the experts found each rule. In addition, measures of the degree of agreement between the experts were calculated, and are discussed in Section 3.1.

2.4 Computing Association Rule Metrics

After the expert coders rated the 120 selected association rules, additional interestingness measures from Tan et al. (2004) were computed in Microsoft Excel. The following metrics were computed for each rule:

- Phi Coefficient
- Cosine
- Piatetsky-Shapiro
- Jaccard, Laplace
- Certainty Factor
- Added Value
- Klosgen
- Odds Ratio
- Cohen's Kappa
- Gini Index
- Conviction
- J Measure
- Collective Strength

In addition, non-standard metrics were created, under the hypothesis that these metrics might also capture some key aspects of expert perception of interestingness in this domain, where an expert might be looking for evidence of successful students or unsuccessful students:

- The number of elements in a rule with values equal to Yes, Correct, and/or On task behavior.
- The number of elements in a rule with values equal to No, Incorrect, and/or Off task behavior.

3. Results

The findings of the research are presented in this section. First, examples of some association rules rated as very interesting, not interesting, and with mixed rating, are presented. Then, results about the inter-rater agreement are included. Finally, correlations between the experts´ ratings and the association metrics are described, and regression models are presented that make combined predictions of expert ratings from a combination of association metrics.

3.1 The Most and Least Interesting Rules

As discussed in the previous section, each rule was rated for perceived interestingness by each of the four expert coders. Below, we present some of the most interesting and least interesting rules, in their perception. Note that each rule represents a transition from time t_1 (left side of rule) and time t_2 (right side of rule). Note also that rules are presented with the exact same operators as generated by the algorithm, which means that some redundancy is present.

The most interesting rules according to the experts (e.g. the rules with the highest average interestingness) were:

{Got incorrect answer, not frustrated} → {Gaming the system}

{Gaming the system, bored, not in engaged concentration, got the incorrect answer and did not request a hint}} → {Confused}

{Off-task, confused, not bored, got the correct answer, and did not request a hint} → { Off-task}

The following rules were rated as least interesting by the experts (in terms of average rating).

{In engaged concentration, did not request a hint, not bored or frustrated or confused or off-task or gaming the system} → {Off-task}

{In engaged concentration, got correct answer, did not request a hint, not frustrated or confused or off-task or gaming } → {Not gaming the system}

{In engaged concentration, got correct answer, did not request a hint, not bored or frustrated or confused or off-task or gaming the system} → {Not frustrated}

However, some rules obtained a high rating from two experts but low rating from the other two:

{Got incorrect answer, did not request a hint, not in engaged concentration or frustrated or off-task or gaming} → {Confused}

{Got incorrect answer, did not request a hint, bored, not concentrated or frustrated or confused or gaming} → {Not being frustrated}

The first of these rules was rated as not interesting by two members of the same research group (experts 2 and 3 below) but rated as very interesting by two members of other research groups. The second rule, however, was rated highly by experts 1 and 2, who belong to different research groups, and it was rated as less interesting by experts 3 and 4.

3.2 Agreement among raters

Though there was generally good agreement between experts, some rules led to disagreement between the coders in terms of interestingness, as shown above. To see the degree of agreement (and to evaluate whether it was feasible to use these expert codes as a basis for studying which metrics best evaluate interestingness), we checked to make sure there was consistency among the four domain experts, using multiple metrics. The estimated Cronbach´s Alpha coefficient for the consistency in rating among the four experts was 0.845, which indicates there is a high covariation among experts in their ratings of interestingness of different rules. The general Intraclass Correlation for the agreement among the four raters was 0.487, which indicates a moderate agreement among the experts (Bartko, 1966). It is worth noting that while Cronbach's Alpha expresses a measure of covariation in the ratings among experts, Intraclass Correlation estimates reliability as the magnitude of disagreement/agreement among the experts (Hallgren, 2012). Hence, the difference among both measures reflects a discrepancy of what each statistic estimates. In the context of our results, these statistics mean that while the experts showed consistency in the way they rated each rule, only a moderate agreement among experts was achieved.

Additionally, Spearman correlation coefficients were calculated to determine the degree of agreement between each pair of experts based on their rating of interestingness to the 120 association rules. Results of the Spearman correlation coefficients are included in the table 2, which indicate there was a significant degree of consistency among the four experts. As this table shows, all four experts had a reasonable degree of consistency, but experts 1 and 2 showed higher agreement with each other, while experts 3 and 4 had higher agreement with each other. Overall, there was moderate to high agreement among the experts in their rating of interestingness of different association rules.

Table 2. Spearman correlation coefficients among experts.

	Expert 1	Expert 2	Expert 3	Expert 4
Expert 1	1			
Expert 2	.744	1		
Expert 3	.548	.590	1	
Expert 4	.580	.516	.674	1

3.3 Correlation between expert judgments and association metrics

Though there was some structure in terms of agreement (e.g. coders 1 and 2 agreed more, and coders 3 and 4 agreed more), the overall agreement between coders was sufficient to create a single metric representing the interestingness of each rule. This metric was created by taking the average of the four coders' ratings for each rule.

Next, Spearman correlation coefficients were calculated to analyze the degree of association between the expert ratings of interestingness and the metrics of interestingness computed in R (R Development Core Team, 2012; Gamer et al., 2012; Fletcher, 2010) and Excel. The resultant correlation coefficients are presented in Table 3. This table shows that the experts' ratings of interestingness were highly correlated with some association rule measures. 7 of the 24 metrics were more highly correlated with the expert ratings of interestingness than the experts' ratings of interestingness correlated with one another, on average. The most highly correlated metrics were Jaccard (r= -0.838), Cosine (r= -0.835), and Support (r = -0.82). As shown in Table 3, the metrics that agreed least well with expert ratings of interestingness were Added Value (r= -0.014) and Kappa (r=-0.029). Merceron & Yacef's (2008) recommendation to use Cosine agrees with our findings here; their recommendation to use Lift does not, at least initially. But they recommend using these metrics in concert, not individually. In the next section, we consider what mixture of metrics best predicts human judgments of interestingness.

3.4 Predicting Expert Perception of Interestingness from a Combination of Metrics

After looking at the predictive power of each metric, taken individually, we built a model that predicted expert judgments using a combination of metrics. Doing so may allow us to create a meta-metric that could be a better representation of interestingness than any single metric by itself.

A linear regression model was created to predict the average expert judgment of interestingness. For this full model, no variable selection was conducted – e.g. all metrics listed above were incorporated into this model. Although the model had statistically significant fit statistics (r= 0.938, r^2 = 0.879, Cross-validated r^2 =0.73, AIC = 123.2702, BIC = 181.8075; F(19, 100) = 38.24, p-value = 0.001), it also had a high degree of multicollinearity among the predictors, measured by the Variance Inflation Factor (VIF). Multicollinearity can lead to over-fitting, as well as making it very difficult to interpret the estimated values for the regression coefficients and their standard errors. This model is reported in table 4.

Table 3. Spearman correlation among inter-rater average and association rules metrics.

	Correlation to Inter-Judge Average	p-value
Jaccard	-0.838	<0.001
Cosine	-0.835	<0.001
Support	-0.82	<0.001
Certainty Factor	0.775	<0.001
Confidence	-0.747	<0.001
Laplace rule	-0.647	<0.001
Count var. of 1´s	-0.609	<0.001
Conviction	-0.432	<0.001
Count var. of 0´s	-0.368	<0.001
Klosgen	-0.327	<0.001
Gini Index	-0.32	<0.001
Odds Ratio	-0.31	0.001
Yule's Q	-0.31	0.001
Yule's Y	-0.31	0.001
Piatetsky-Shapiro	-0.303	0.001
J Measure	-0.303	0.001
Collective Strength	-0.298	0.001
Phi Coefficient	-0.29	0.001
Lift	0.202	0.027
Kappa	-0.029	0.754
Added Value	-0.014	0.876

Table 4. Regression model with all association rules metrics and counting variables as predictors

Predictor	Coeff	S.E.	T	P-val	VIF
Intercept	106.44	33.13	3.21	0.001	
Count var. of 1´s	-0.042	0.097	-0.43	0.664	4.2
Count var. of 0´s	-0.01	0.081	-0.13	0.896	13.3
Support	44.085	19.83	2.22	0.028	2375.2
Confidence	0.899	1.617	0.55	0.579	230.6
Lift	-28.56	13.46	-2.12	0.036	2117.1
Phi Coefficient	47.673	26.51	1.79	0.075	1422.1
Cosine	34.443	47.58	0.72	0.470	24213.7
Piatetsky Shapiro	-80.69	509.0	-0.15	0.874	18302.8
Jaccard	-108.6	57.81	-1.87	0.063	16274.3
Laplace	-10.62	9.39	-1.13	0.260	3832.8
Certainty Factor	-17.37	8.473	-2.05	0.042	347.8
Added Value	49.036	36.95	1.32	0.187	2257.6
Klosgen	-78.83	187.3	-0.42	0.674	6543.2
Odds Ratio	-0.235	4.097	-0.05	0.954	10700.1
Kappa	172.97	70.26	2.462	0.015	6245.3
Gini Index	-437.2	283.4	-1.54	0.126	712.7
Conviction	-2.369	5.516	-0.42	0.668	7758.5
J Measure	1038.7	502.4	2.068	0.041	1851.9
Collective Strength	-68.02	36.70	-1.85	0.066	8265.7

A second linear regression model was tested including just statistically significant association metrics as predictors with small multicollinearity among them. The predictors excluded from this analysis were: Support, Confidence, Piatesky Shapiro, Jaccard, Laplace, Certainty Factor, Added Value, Klosgen, Odds Ratio, Kappa, Gini Index, J Measure, and Collective Strength. Those omitted predictors presented moderate to high correlations with one or more association metrics included in the model summarized in table 5. The criteria for exclusion were high correlations among the predictors that, consequently, resulted in VIF values higher than 10 for a given model.

Results of this second regression model showed that two association rule metrics –Lift and Conviction– had a positive prediction coefficient, while other two metrics – the Phi Coefficient and Cosine– had a negative coefficient. The model fit statistics were statistically significant and explained almost as much of the variance as the full model, which achieved a substantially higher cross-validated correlation (r= 0.902, r^2 = 0.814, Cross-validated r^2 =0.791, AIC = 144.4186, BIC = 161.1436; F = 126.4, df_1 = 4, df_2 = 115, p-value = 0.001). Table 5 summarizes the second regression model. The lower values of BIC in the second model confirm it is a better and more simple model compared with the former one.

Table 5. Regression model with association rules metrics with restriction for multicollinearity

Predictor	**Coeff**	**S.E.**	**T**	**P-val**	**VIF**
Intercept	0.404	1.023	0.395	0.6937	
Lift	3.848	0.790	4.870	<0.001	5.477
Phi Coef.	-11.179	2.220	-5.034	<0.001	7.491
Cosine	-5.783	0.585	-9.880	<0.001	2.752
Conviction	0.469	0.116	4.013	<0.001	2.616

Although Jaccard presented the highest correlation with the inter-rater average score, it also presented a very high correlation with many other metrics, including Cosine (r = 0.96). Thus, many models that included Jaccard also presented a high degree of multicollinearity among the predictors; as a consequence, Jaccard was excluded in the combined model presented in table 5. Table 6 demonstrates a model similar to the model in table 5 but replacing Cosine with Jaccard. The model in this case was not better in terms of multicollinearity and was only slightly better in terms of goodness-of-fit (r = 0.908, r^2 = 0.825, Cross-validated r^2 =0.81, AIC = 137.5014, BIC = 164.2263; F(4, 115) = 135.6, p-value = <0.001).

Table 6. Regression model including Jaccard instead of Cosine

Predictor	**Coeff**	**S.E.**	**T**	**P-val**	**VIF**
Intercept	0.547	0.986	0.556	0.579	
Lift	3.502	0.778	4.496	<0.001	5.638
Phi Coef.	-7.528	2.370	-3.177	0.002	9.038
Jaccard	-8.896	0.847	-10.49	<0.001	2.780
Conviction	0.207	0.121	1.721	0.088	2.945

Regression models were also computed for each individual metric used in the combined models. The results, which are summarized in table 7, show that single-feature models presented less desirable fit statistics (i.e., r^2, AIC, and BIC) than the combined model. The model including just Jaccard as predictor has the best fit statistics among the single-variable models (with Cosine close behind), but the combined model is still superior.

.

Table 7. Regression models with single predictors

Predictor	**Coeff – Intercepet (S.E.)**	**p**	**Fit Stats.**
Lift	2.81*Lift – 0.80 (0.729)	<0.001	**R^2** = 0.112 **CV-R^2**= 0.074 **AIC** = 326.43 **BIC** = 334.79
Phi Coefficient	-6.744*Phi + 2.78 (1.754)	<0.001	**R^2** = 0.111 **CV-R^2**= 0.098 **AIC** = 326.56 **BIC** = 334.93
Cosine	-7.72*Cosine + 5.39 (0.387)	<0.001	**R^2** = 0.771 **CV-R^2**= 0.754 **AIC** = 163.71 **BIC** = 172.07
Conviction	-0.69*Conviction + 3.2 (0.152)	<0.001	**R^2** = 0.149 **CV-R^2**= 0.119 **AIC** = 321.24 **BIC** = 329.61
Jaccard	-11.56*Jaccard + 4.84 (0.552)	<0.001	**R^2** = 0.787 **CV-R^2**= 0.779 **AIC** = 154. 84 **BIC** = 163.21

4. Discussion and Conclusions

As seen in this paper, several standard association rule metrics can predict human expert ratings of interestingness of an association rule. Most commonly used interestingness metrics showed statistically significant correlations with the experts' ratings of interestingness, but not all of them were included in the final combined model given the high common variation among them. The best metrics – Jaccard, Cosine, and Support – achieved an absolute correlation higher than 0.80 with the average expert human judgment, which is higher than the average correlation of the ratings between experts. Hence, we see that these association metrics are a good substitute for human ratings of interestingness

In particular, our findings agree with Merceron and Yacef (2008) that Cosine and Lift are useful, as they were successful predictors in the final combined model in this data set. Taken individually, Cosine was good predictor, while Lift explained considerably less variance. The association metric Cosine consistently had a high negative correlation with the raters' scores of interestingness and significantly predicted expert ratings of interestingness, both in a single-predictor model and in combination with other association metrics. The association metric Lift had a positive correlation and significantly predicted the average score of interestingness among the experts in combination with other metrics and in a single-predictor model; however, Lift was relatively weak compared to other metrics when taken by itself.

However, one surprise is that Cosine, while important in both our findings and in Merceron & Yacef, was correlated to interestingness in the negative direction in our findings (i.e. low, while Merceron & Yacef recommend looking for high Cosine). This finding is surprising, and merits further study. One possibility is that once support and confidence are accounted for, then interestingness links in some ways to rarity. Perhaps that is not surprising –facts that are already known are not particularly interesting– but it does show that the association rule mining conception of interestingness may not quite match intuitive notions of this construct. In our view, this finding is itself impressive. In general, this result suggests that Cosine is indeed important, but may reflect interestingness in a different way than previously understood.

In addition, other association rule metrics – the Phi Coefficient, Conviction and Jaccard – that have not been widely used in educational data mining also explained a significant proportion of the variance in the combined model and a in single-predictor models. Therefore, it might be useful to also consider these metrics in future research using association rule mining in educational data sets.

On the whole, results in this study show that the recommended metrics of interestingness proposed by Merceron and Yacef (2008) are useful, as well as other metrics not considered by those authors.

It is worth considering some limitations of this study. First, only linear correlations and linear regression models were considered. Although these approaches achieved good fit to the data, and explained much of the variance, it could be useful to consider models with non-linear relations among the association rules metrics and the expert ratings. Second, given the high correlation among different association rules metrics, other measures could be considered as alternative predictors of the inter-rater score of interestingness instead of the four measures chosen in the final regression model reported. Third, this paper represents a single analysis in a single educational research domain. Results might vary in a different educational research domain, or indeed outside of education. However, the fact that Cosine and Lift were prominent both in our models and in the recommendations in Merceron & Yacef (2008) is a positive sign, given that their work involved a very different area of educational research.

Overall, the use of association mining to understand complex and interesting relations among different variables is a method with a lot of potential in educational data mining research. Association rules can be understood at an intuitive level, and can provide useful information for a variety of stakeholders who are not experts in EDM, including students, teachers, administrators, and policy makers. However, given the huge numbers of association rules that can be generated, it is important to try to filter not just by support and confidence, but by interestingness as well. By using the metric or combination of metrics that matches an intuitive conception of interestingness, we can provide the most interesting information to users of association rules first, improving the efficiency of this method.

Acknowledgements

The authors wish to thank the National Science Foundation, Innovative Technology Experiences for Students and Teachers (Grant #DRL-1031398) and to the Bill and Melinda Gates Foundation, Vital Behaviors and Skills Associated with Engagement Derived from Learning Analytics Topic (Grant #OPP1048577) for their support. We also want to thank Leena Razzaq and Neil Heffernan for making the ASSISTments system data available to us, and to Amber Chauncey Strain and to Jonathan Rowe for providing their expert opinions.

5. References

[1] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference Very Large Data Bases* (pp. 487-499)

[2] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast Discovery of Association Rules. *Advances in knowledge discovery and data mining*, *12*, 307-328.

[3] Agrawal, R., & Srikant, R. (1995, March). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering* (pp. 3-14). IEEE.

[4] Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, *68*(4), 223-241.

[5] Baker, R., & Yacef, K. (2009). The State of Educational Data Mining in 2009: A Review and Future Visions. *Journal of Educational Data Mining, 1* (1), 3-17

[6] Baker, R., Siemens, G. (in press) Educational data mining and learning analytics. To appear in Sawyer, K. (Ed.) *Cambridge Handbook of the Learning Sciences: 2nd Edition*.Cambridge, UK: Cambridge University Press.

[7] Bartko, J. J. (1966). The intraclass correlation coefficient as a measure of reliability. *Psychological reports*, *19*(1), 3-11.

[8] Ben-Naim, D., Bain, M., & Marcus, N. (2009, July). A User-Driven and Data-Driven Approach for Supporting Teachers in Reflection and Adaptation of Adaptive Tutorials. Paper presented at *Educational Data Mining*, (pp. 21-30)

[9] Chauncey, A., & Azevedo, R. (2010, January). Emotions and motivation on performance during multimedia learning: how do i feel and why do i care?. Paper presented at *Intelligent Tutoring Systems* (pp. 369-378). Springer Berlin Heidelberg.

[10] Fletcher, T. D. (2010). Psychometric: Applied Psychometric Theory. R package version 2.2. Retrieved from URL http://CRAN.R-project.org/package=psychometric

[11] Freyberger, J. E., Heffernan, N., & Ruiz, C. (2004). *Using association rules to guide a search for best fitting transfer models of student learning* (Doctoral dissertation). Worcester Polytechnic Institute, Worcester, MA.

[12] Garcia, E., Romero, C., Ventura, S., & Calders, T. (2007, September). Drawbacks and solutions of applying association rule mining in learning management systems. In *Proceedings of the International Workshop on Applying Data Mining in e-Learning* (pp. 13-22).

[13] Garcia, E., Romero, C., Ventura, S., & De Castro, C. (2009). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Modeling and User-Adapted Interaction*,*19*(1-2), 99-132

[14] Hahsler, M., Grün, B., Hornik, K., & Buchta, C. (2009). Introduction to arules–A computational environment for

mining association rules and frequent item sets. *The Comprehensive R Archive Network.*

[15] Hahsler, M., Gruen, B., & Hornik, K. (2005). arules - A Computational Environment for Mining Association Rules and Frequent Item Sets. *Journal of Statistical Software. 14*(15). URL: http://www.jstatsoft.org/v14/i15/.

[16] Hallgren, K. A. (2012). Computing inter-rater reliability for observational data: An overview and tutorial. Tutorials in quantitative methods for psychology, 8(1), 23-34.

[17] Hershkovitz, A., Baker, R. S., Gobert, J., & Nakama, A. (2012). A Data-driven Path Model of Student Attributes, Affect, and Engagement in a Computer-based Science Inquiry Microworld. In *Proceedings of the International Conference on the Learning Sciences*.

[18] Kinnebrew, J. S., Loretz, K. M., & Biswas, G. (2012). A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, *5*(1), 190-219

[19] Gamer, M., Lemon, J., Fellows, I., & Singh, P. (2012). irr: Various Coefficients of Interrater Reliability and Agreement. R package version 0.84. Retrieved from URL http://www.R-project.org.

[20] Merceron, A., & Yacef, K. (2008). Interestingness Measures for Associations Rules in Educational Data. *Educational Data Mining*, *8*, 57-66.

[21] Lu, J. (2004). Personalized e-learning material recommender system. Paper presented at *International conference on information technology for application* (pp.374–379)

[22] Pardos, Z. A., Baker, R. S., San Pedro, M. O., Gowda, S. M., & Gowda, S. M. (2013, April). Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (pp. 117-124). ACM.

[23] Perera, D., Kay, J., Koprinska, I., Yacef, K., Zaiane, O. (2009) Clustering and Sequential Pattern Mining of Online Collaborative Learning Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*, 759-772.

[24] R Development Core Team. (2012). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from URL http://www.R-project.org.

[25] Razzaq, L., Heffernan, N., Feng, M., & Pardos, Z. (2007). Developing Fine-Grained Transfer Models in the ASSISTment System. *Journal of Technology, Instruction, Cognition, and Learning*, *5*(3), 289-304.

[26] Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, *33*(1), 135-146.

[27] Romero, C., Ventura, S., Delgado, J. A., & De Bra, P. (2007). Personalized links recommendation based on data mining in adaptive educational hypermedia systems. In E. Duval, R. Klamma, and M. Wolpers (Eds.). *Creating New Learning Experiences on a Global Scale* (pp. 292-306). Berlin: Springer.

[28] Romero, C., & Ventura, S. (2010). Educational Data Mining: A Review of the State of the Art. *IEEE transactions on systems, man and cybernetics. Part C, Applications and reviews*, *40*(6), 601-618.

[29] Sabourin, J., Rowe, J. P., Mott, B. W., & Lester, J. C. (2011, January). When off-task is on-task: the affective role of off-task behavior in narrative-centered learning environments. Paper presented at *the 15th International Conference In Artificial Intelligence in Education* (pp. 534-536).

[30] San Pedro, M. O. Z., Baker, R. S., Gowda, S. M., & Heffernan, N. T. (2013). Towards an Understanding of Affect and Knowledge from Student Interaction with an Intelligent Tutoring System. In *Proceedings of the 16th International Conference on Artificial Intelligence and Education*.

[31] Scheuer, O., & McLaren, B. M. (2012). Educational data mining. In N. M. Seel (Ed.). *Encyclopedia of the Sciences of Learning* (pp. 1075-1079). Springer US.

[32] Tan, P. N., Kumar, V., & Srivastava, J. (2004). Selecting the right objective measure for association analysis. *Information Systems*, *29*(4), 293-313

Different parameters - same prediction: An analysis of learning curves

Tanja Käser
Departement of Computer Science
ETH Zurich
kaesert@inf.ethz.ch

Kenneth R. Koedinger
Human-Computer Interaction Insitute
Carnegie Mellon
koedinger@cmu.edu

Markus Gross
Departement of Computer Science
ETH Zurich
grossm@inf.ethz.ch

ABSTRACT

Using data from student use of educational technologies to evaluate and improve cognitive models of learners is now a common approach in EDM. Such naturally occurring data poses modeling challenges when non-random factors drive what data is collected. Prior work began to explore the potential parameter estimate biases that may result from data from tutoring systems that employ a mastery learning mechanism whereby poorer students get assigned tasks that better students do not. We extend that work both by exploring a wider set of modeling techniques and by using a data set with additional observations of longer-term retention that provide a check on whether judged mastery is maintained. The data set at hand contains math learning data from children with and without developmental dyscalculia. We test variations on logistic regression, including the Additive Factors Model and others explicitly designed to adjust for mastery-based data, as well as Bayesian Knowledge Tracing (BKT). We find these models produce similar prediction accuracy (though BKT is worse), but have different parameter estimation patterns. We discuss implications for use and interpretation of these different models.

Keywords

learning curves, logistic regression models, knowledge tracing, parameter fitting, prediction accuracy

1. INTRODUCTION

Modeling student knowledge is a fundamental task when working with intelligent tutoring systems. The selection of tasks and actions is based on the student model, therefore an accurate prediction of student knowledge is essential. The accuracy of the student model depends on the quality of the parameter fit. Parameter fitting is, however, not only important for prediction accuracy; the parameters of a model also contain information on how students learn.

A variety of approaches to assess, interpret and predict student knowledge have been proposed. Popular techniques to model student learning include Bayesian Knowledge Tracing (BKT) [8], Bayesian networks [4, 9, 10], performance factors analysis [21] and Additive Factors Models (AFM) [5, 6].

BKT is one of the most popular approaches for student modeling. Prediction accuracy of the original BKT model has been improved using clustering approaches [20] or individualization techniques, such as learning student- and skill-specific parameters [16, 19, 24, 26] or modeling the parameters per school class [21].

The AFM is a generalized linear mixed model [2] applying a logistic regression. It is widely used to fit learning curves and to analyze and improve student learning. AFM helps identify flat or ill-fitting learning curves that indicate opportunities for tutor or model improvement. Consistently low error curves indicate opportunities to reallocate valuable student time [5]. Consistently high error curves with poor fit indicate a miss-specified skill model that can be improved [15, 23] and used to design better instruction [14]. However, when working with mastery learning data sets, averaging over students who have different initial knowledge states and learning rates may lead to learning curves which show little student learning. It has been shown [17] that disaggregating a learning curve into curves for different subpopulations or mastery-align the learning curves provides more accurate metrics for student learning. However, so far there exist no comparisons between the properties of the different models, such as the parameter fit. Furthermore, the models were also not validated regarding prediction accuracy.

In this work, we therefore extensively evaluate the properties and parameters of different logistic regression models when fitting learning curves to a mastery learning data set containing students with heterogeneous knowledge levels. We turn the suggestions of [17] for fitting learning curves in BKT into logistic regression models and also introduce a further alternative model to the AFM. The data set at hand was collected from an intelligent tutoring system for learning mathematics and includes log files from 64 children with developmental dyscalculia and 70 control children. Our findings show that similar regression models predict very different amounts of learning for the same data. Furthermore, we demonstrate that different parameter fits lead to the same prediction accuracy on unseen data. For further validation, we compare prediction accuracy of logistic regression models to that of BKT and analyze how well these models generalize to new students. Our results demonstrate that logistic regression models outperform BKT regarding prediction accuracy on unseen data.

2. METHOD

In the following, we first introduce different logistic regression models and their properties. We then give a short overview of BKT and finally explain the experimental setup.

2.1 Logistic regression models

Logistic regression models are used in Item Response Theory (IRT) [25] to model the response (correct/wrong) of a student to an item. IRT is based on the idea that the probability of a correct response to an item is a mathematical function of student and item parameters. The logistic regression models presented in the following are based on this concept.

Additive Factors Model (AFM). The AFM [5, 6] is a logistic regression model fitting a learning curve to the data. In a logistic regression model, the observations of the students follow a Bernoulli distribution. A Bernoulli distribution is a binomial distribution with $n = 1$. Letting $y_{pi} \in \{0, 1\}$ denote the response of student p on item i, we obtain $y_{pi} \sim binomial(1, \pi_{pi})$. The linear component π_{pi} of the AFM can then be formulated as follows:

$$\pi_{pi} = logit(\theta_p + \sum_k q_{ik} \cdot (\beta_k + \gamma_k \cdot T_{pk})), \qquad (1)$$

with $\theta_p \sim \mathcal{N}(0, \sigma_\theta^2)$. The AFM is a generalized linear mixed model with a random effect θ_p for student proficiency and fixed effects β_k (difficulty) and γ_k (learning rate) for the skills k (knowledge components). The learning rate γ_k is constrained to be greater than or equal to zero for AFMs. q_{ik} is 1, if item i uses skill k and 0 otherwise. Finally, T_{pk} denotes the number of practice opportunities student p had at skill k. The AFM is related to the linear logistic test model (LLTM) [25] and the Rasch model [25]. When removing the third term $(\gamma_k \cdot T_{pk})$ of Equation 1, we obtain an LLTM. Additionally assuming a unique-step skill model (one skill per step) results in the Rasch model. The intuition of the AFM is that the probability of a student getting a step correct is proportional to the amount of required knowledge of the student θ_p, plus the difficulty of the involved skills β_k and the amount of learning gained from each practice opportunity γ_k.
Learning curves are averaged over many students. The AFM aligns the students by opportunity count. When applied to mastery learning data, it therefore suffers from student attrition with increasing numbers of opportunities. Well performing students need few opportunities to master a skill and thus only the weaker students remain in the analysis for higher opportunity counts. This student attrition can lead to an underestimation of the learning rates γ_k. In the following, we therefore introduce alternative logistic regression models that adjust for mastery-based data.

Learning Gain Model (LG). With the LG model, we introduce a new alternative to the AFM. The LG model avoids student attrition by aligning the students at their first sample (when they start the training) and at their last sample, i.e., when they end the training (independent of whether they mastered the skill or not). The linear component of this model is very similar to that of the AFM:

$$\pi_{pi} = logit(\theta_p + \sum_k q_{ik} \cdot (\beta_k + \gamma_k \cdot N_{pk})), \qquad (2)$$

where $N_{pk} \in [0, 1]$ denotes the normalized opportunity count of student p at skill k, i.e., we normalize over all opportunities student p had at skill k during the training. Rather than measuring the amount learnt per opportunity, this model estimates the learning gain of the students over the course of the training.

Alternative logistic regression models. To adjust for mastery-based data, alternative ways to fitting the curves have been proposed [17] for BKT. In the following, we reformulate these suggestions and apply them to logistic regression models. The **Mastery-Aligned Model (MA)** can be formulated using Equation 1, but with a different definition of T_{pk}. For the MA model, we count backwards: T_{pk} is the number of opportunities student p had at skill k as seen from mastery. T_{pk} is 0 at mastery, 1 at one opportunity before mastery and so on. Thus, the MA model aligns students at mastery, which solves the problem of student attrition. A different way to deal with student attrition is to group students by the number of opportunities needed to first master a skill. The linear component of this **Disaggregated Model (DIS)** can be defined as follows:

$$\pi_{pi} = logit(\theta_p + \sum_{k,m} q_{ik} \cdot (\beta_{k,m} + \gamma_{k,m} \cdot T_{pk})), \qquad (3)$$

where the difficulty $\beta_{k,m}$ and the learning rate $\gamma_{k,m}$ are fit by skill k and mastery group m. By combining the MA and the DIS models, the **Mastery-Aligned and Disaggregated Model (DISMA)** can be constructed. This model disaggregates students into groups based on the number of opportunities needed until mastering the skill and furthermore aligns the students at mastery.

All models presented are generalized linear mixed models (GLMM) as the linear predictor π_{pi} contains random effects (for the students) in addition to the fixed effects (for the skills). GLMMs are fit using maximum likelihood, which involves integration over the random effects [3]. Integration is performed using methods such as numeric quadrature or Markov Chain Monte Carlo.

2.2 Bayesian Knowledge Tracing

BKT [8] is a popular approach for modeling student knowledge. BKT models are a special case of Hidden Markov Models (HMM) [22]. In BKT, student knowledge is modeled by one HMM per skill (or knowledge component). The latent variable of the model represents the student knowledge. It indicates whether a student has mastered the skill in question and is therefore binary. The state of this variable is inferred by binary observations, i.e., correct or wrong answers to tasks associated with the skill in question. A HMM can be specified using five parameters. The transmission probabilities of the model are defined by the probability p_L of a skill transitioning from not known to known state and the probability p_F of forgetting a previously known skill. The slip probability p_s of making a mistake when applying a known skill and the guess probability p_g of correctly applying an unknown skill define the emission probabilities of the model. And finally, p_0 denotes the probability of knowing a skill a-priori. In BKT, the forget probability p_F is assumed to be 0 and therefore a BKT model can be specified with the four parameters $\theta = \{p_0, p_L, p_s, p_g\}$.

An important task when working with BKT models is parameter learning. The learning task can be formulated as follows: Given a sequence of student observations $\mathbf{y} = \{y_t\}$ with $t \in [1, T]$, what are the parameters $\theta = \{p_0, p_L, p_s, p_g\}$ that maximize the likelihood of the data $p(\mathbf{y}|\theta)$. BKT models have been fit using expectation maximization [7], brute-force grid search [1] or gradient descent [26].

2.3 Experimental setup

The training environment we use in this work consists of `Calcularis`, a tutoring system for children with difficulties in learning mathematics [11]. The program transforms current neuro-cognitive findings into the design of different instructional games, which are classified into two parts. The first part focuses on the training of different number representations and number understanding. In the second part, addition and subtraction are trained at different difficulty levels. Task difficulty depends on the magnitude of numbers involved, the complexity of the task and the means allowed to solve the task. The employed student model is a dynamic Bayesian network modeling different mathematical skills and their dependencies. The controller acting on the skill net is rule-based and allows forward and backward movements (increase and decrease of difficulty levels) [12, 13].

The data set used for the experimental evaluation was collected in a large-scale user study in Switzerland and Germany with 134 participants (69% females). 64 participants (73% females) were diagnosed with developmental dyscalculia (DD) and 70 participants (66% females) were control children (CC). All children were German-speaking and visited the 2^{nd}-5^{th} grade of elementary school (mean age: 8.68 (SD 0.84)). Children trained with the program for six weeks with a frequency of five times per week during sessions of 20 minutes. The collected log files contain at least 24 complete sessions per child. On average, each child solved 1521 tasks (SD 269) during the training. Results of the external pre- and post-tests demonstrated a significant improvement in spatial number representation, addition and subtraction after the training [11].

We investigated 20 addition and subtraction skills in the number range $0-100$. For our analyses, we used two versions of the data set. The first version (denoted as *Version 1* in the following) contains the samples of all children at the respective skills, while the second version (denoted as *Version 2* in the following) includes only children that mastered the respective skills. *Version 2* of the data set makes the inclusion of the MA and DISMA models possible. However, it excludes students not mastering a skill from the analysis, which leads to a more homogeneous, but due to the drop-out of many children with DD, also less interesting data set. *Version 1* of the data set contains $36'350$ solved tasks, while *Version 2* consists of $20'784$ tasks. External paper-pencil and computer-based arithmetic tests conducted at the beginning and at the end of the study demonstrated significant improvement in addition and subtraction in the number range $0-100$.

3. EVALUATION AND RESULTS

In a first study, we analyzed the parameter fit of different regression models and evaluated their performance in prediction of new items. Furthermore, we compared prediction accuracy of regression models to that of traditional BKT. We used all the samples until the children mastered a skill and predicted the outcome of the first re-test. In a second experiment, we evaluated the prediction accuracy of regression models as well as BKT when generalizing to new students. We fitted the model based on a subset of students and predicted the outcome for the rest of the students. Prediction accuracy for both experiments was measured using the root mean squared error (RMSE), the accuracy (number of correctly predicted student successes/failures based on a threshold of 0.5) and the area under the ROC curve (AUC). Prediction accuracy was computed using bootstrap aggregation with re-sampling ($n = 200$) in the first experiment and a student-stratified 10-fold cross validation in the second experiment.
Fitting for the regression models was done in `R` using the `lme4` package. To be able to compare the parameter fit of the different models, we did not constrain γ_k to be greater than or equal to zero. Parameters for BKT were estimated by maximizing the likelihood $p(\mathbf{y}|\theta)$ using a Nelder Mead simplex optimization [18]. This minimization technique does not require the computation of gradients and is for example available in `fminsearch` of `Matlab`. The following constraints were imposed on the parameters: $p_g \leq 0.3$ and $p_s \leq 0.3$.

3.1 Analysis of parameter fit

In this experiment, we investigated the parameter fit of three regression models on the data set *Version 1*: The AFM, the LG model and the DIS model. The three models obtain very different parameter estimations for the same data. While the AFM model predicts learning (positive γ_k) for 50% of the skills, the LG model fits positive learning rates γ_k for all skills and the DIS model obtains positive learning rates $\gamma_{k,m}$ for 92% of the cases. We therefore analyze the residuals and prediction accuracy of the different models in the following.

Residual analyses. All three models tend to overestimate the outcome for badly performing students and underestimate the outcome for well performing students. This finding is also visible in Fig. 1, which displays the mean residuals r with $r =$ *fitted outcome - true outcome* by estimated student proficiency θ_p. Furthermore, the residuals r are strongly correlated to student proficiency ($\rho_{AFM} = -0.9621$, $\rho_{LG} = -0.9612$, $\rho_{DIS} = -0.9532$). These results are as expected, because the models' predictions are averaged over all the students. While the residuals r are very similar for the AFM and the LG models, the DIS model exhibits less variance in student proficiency. As the students are grouped by the number of opportunities needed to master a skill, student proficiency within a group is more homogeneous.

For the AFM and the LG model, we also analyzed the mean residuals r regarding the skill parameters β_k and γ_k from the models. There are no significant correlations between

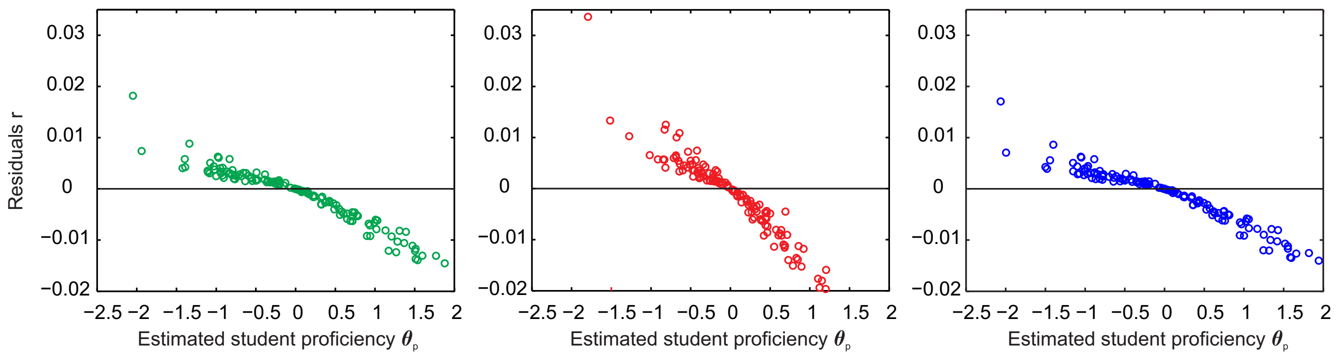


Figure 1: Mean residuals r by estimated student proficiency θ_p for the AFM (left), the DIS (middle) and the LG (right) model.

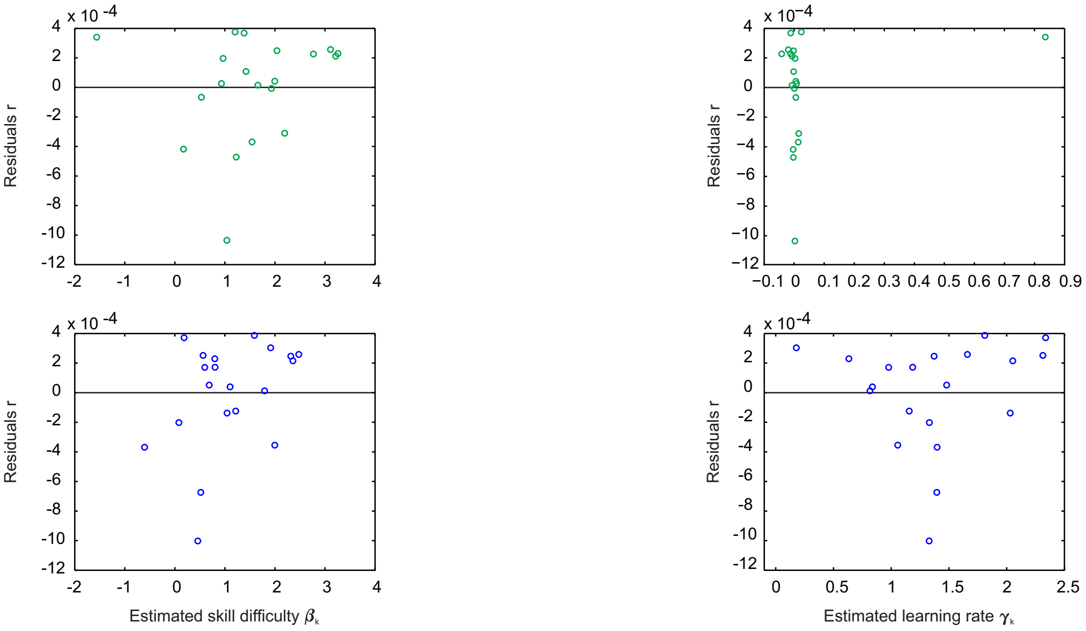


Figure 2: Mean residuals r by estimated skill difficulty β_k for the AFM (top) and the LG model (bottom).

Figure 3: Mean residuals r by estimated learning rates γ_k for the AFM (top) and the LG model (bottom).

skill difficulty β_k and mean residuals r neither for the AFM ($\rho_{AFM} = 0.1677, p_{AFM} = .4798$) nor for the LG model ($\rho_{LG} = 0.3777$, $p_{AFM} = .1066$). From Fig. 2, which displays the mean residuals r by estimated skill difficulty β_k, it is also obvious that these measures are not related for both models. The residuals r are also not correlated to the estimated learning rate γ_k ($\rho_{AFM} = 0.2058$, $p_{AFM} = .3840$; $\rho_{LG} = 0.1051$, $p_{LG} = .6592$) as displayed in Fig. 3. Figure 3 demonstrates how different the parameter fits of the two models are regarding the learning rates γ_k. The AFM fits learning rates γ_k in a very small range around 0 and 45% of the learning rates are not significantly different from zero. The outlier stems from a skill played by only two students resulting in a total of 14 solved tasks. Learning rates γ_k fitted by the LG model are all positive and exhibit a larger variance. This larger variance appears to result from AFM having a bias to underestimate learning rate (because mastery leaves more poor students contributing to high opportunity counts) and LG having a bias to overestimate learning rate (because the adjusted end-point of all learning curves, the last opportunity that achieves mastery, is always successful whether or not it is a true or false positive).

The mean residuals r over time are displayed in Fig. 4. For the AFM and the DIS model, an averaging window ($n = 10$) was used to compute the mean residuals r with increasing opportunity count. Both models underestimate the outcome for less than 20 opportunities and overestimate it for larger numbers. For the AFM, this observation is confirmed by the significant positive correlation between the opportunity

Table 1: Prediction accuracy of first re-test for data set *Version 1* and *2*. The values in brackets denote the standard deviations. The best model per error measure is marked (*).

		RMSE	Accuracy	AUC
Data set: Version 1	AFM	0.3562 (0.0101)*	0.8391 (0.0119)	0.6825 (0.0230)*
	LG	0.3587 (0.0125)	0.8451 (0.0113)*	0.6778 (0.0250)
	DIS	0.3780 (0.0140)	0.8394 (0.0122)	0.6054 (0.0255)
	BKT	0.3614 (0.0111)	0.8428 (0.0118)	0.6033 (0.0250)
Data set: Version 2	AFM	0.3563 (0.0114)*	0.8474 (0.0123)*	0.6622 (0.0250)*
	LG	0.3666 (0.0124)	0.8416 (0.0107)	0.6602 (0.0245)
	DIS	0.3765 (0.0141)	0.8416 (0.0120)	0.5998 (0.0290)
	MA	0.3633 (0.0117)	0.8401 (0.0114)	0.6508 (0.0255)
	DISMA	0.3783 (0.0133)	0.8396 (0.0116)	0.6011 (0.0256)
	BKT	0.3613 (0.0111)	0.8423 (0.0115)	0.6102 (0.0302)

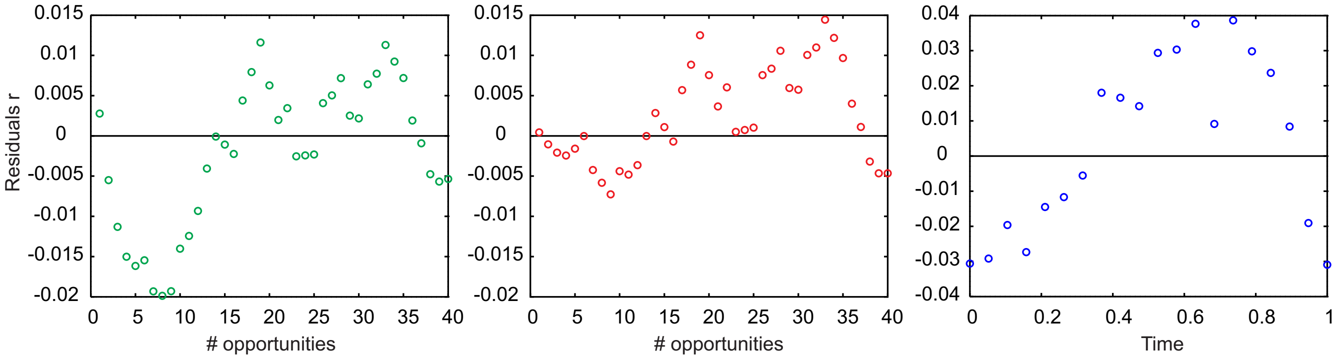


Figure 4: Mean residuals r by opportunity count for the AFM (left) and the DIS (middle) model and by normalized opportunity count for the LG (right) model.

count and the mean residuals r ($\rho_{AFM} = 0.3950$, $p_{AFM} < .001$). This result probably stems from the fact that the well performing students master the skills much faster and therefore student numbers drop with higher opportunity counts. The DIS model exhibits a lower variance, as this model groups the students by the number of opportunities needed to master a skill and thus student performance within a group is more homogeneous ($\rho_{DIS} = 0.0860$, $p_{DIS} = .4785$). For the LG model, the mean residuals r are plotted by the normalized opportunity count in Fig. 4 (right). The LG model underestimates the outcome in the beginning and in the end and overestimates in-between. Through normalizing the opportunity count, we align the beginning and the end of the training for each student. We therefore end up with more observations from low performing students in the middle and the model overestimates the outcome in this part.

Re-test prediction. The residual analyses demonstrate that the models interpret the same data very differently, i.e., the parameter fit and properties of the models vary a lot. To validate these different parameter fits, we computed the prediction accuracy for the first re-test (data set *Version 1*) and compared it to a BKT model. The observed mean outcome over all re-tests is high with 0.8419. The AFM underestimates the true outcome with an average prediction of 0.8287, while the LG (average prediction 0.9108) and DIS models (average prediction 0.9488) overestimate the true outcome. Prediction accuracy for the different models is listed in Tab. 1. The AFM shows the best RMSE ($RMSE_{AFM} = 0.3562$) and AUC ($RMSE_{AUC} = 0.6825$), while the LG models exhibits the highest accuracy ($Accuracy_{LG} = 0.8451$). As the performance of students is generally high, RMSE and AUC are, however, better quality measures than accuracy. The LG model performs second best in RMSE ($RMSE_{LG} = 0.3587$) and AUC ($AUC_{LG} = 0.6778$). However, the small differences between the AFM and the LG model along with the high variances of the error measures indicate that there are no significant differences between the two models. The DIS model on the other hand demonstrates a considerably higher RMSE ($RMSE_{DIS} = 0.3780$) and also exhibits a low AUC ($AUC_{DIS} = 0.6054$) compared to the two other regression models. The DIS model estimates the parameters $\beta_{k,m}$ and $\gamma_{k,m}$ by skill and mastery group. The resulting large number of parameters produces overfitting. Performance on the training data set supports the overfitting hypothesis: The DIS model outperforms the AFM and the LG model in RMSE, accuracy and AUC on the training data set.

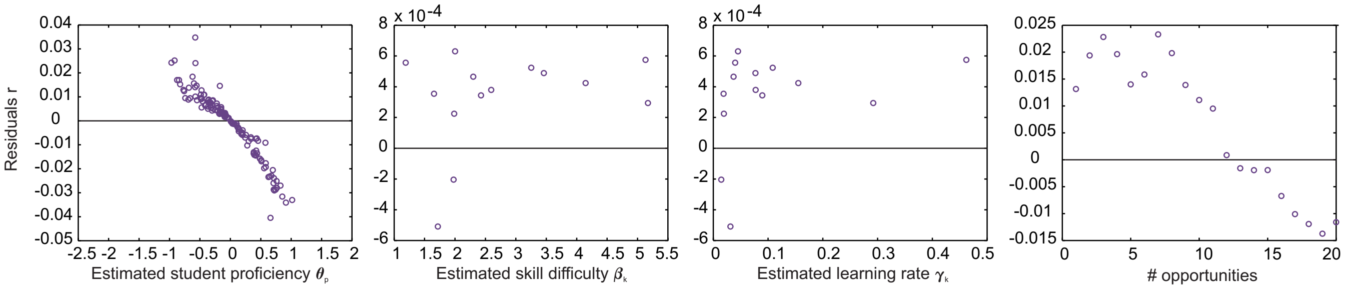


Figure 5: Mean residuals r by estimated student proficiency θ_p (left), skill difficulty β_k (center left), learning rates γ_k (center right) and opportunity count (right) for the MA model.

Interestingly, the AFM and the LG model also outperform the BKT model. The RMSE of BKT ($RMSE_{BKT} = 0.3614$) is higher than those of the two regression models, but standard deviations are again large. BKT exhibits especially a lower performance in AUC ($AUC_{BKT} = 0.6033$). The better performance of the regression models might come from two facts: First, the regression models fit the parameter θ_p for the individual student's proficiency, while traditional BKT does not do any student individualization. Second, BKT assumes that there is no forgetting, while the regression models are allowed to fit negative learning rates γ_k. However, the time between mastering a skill and the first re-test tends to be long. On average, the first re-test was done after 140 opportunities. A logistic regression analysis shows, that there is indeed a small, but significant amount of forgetting (intercept = 1.8545, slope = -0.0012) in the data. The probability of being correct at mastery amounts to 0.8647 and decreases to 0.8419 after 140 opportunities. Note, however, that the forgetting hypothesis is only valid for the AFM, as learning rates γ_k are all positive for the LG model.

Experiments on data set *Version 2*. To be able to include the MA and DISMA models in our analyses, we also evaluated prediction accuracy for the first re-test based on data set *Version 2*.

For this version of the data set, the LG and MA models predict positive learning rates γ_k for 100% of the skills, while the AFM fits positive learning rates γ_k for 54% of the skills. The DIS and DISMA models show positive learning rates $\gamma_{k,m}$ for 90% of the mastery groups. Residuals r of the DISMA model are very similar to those of the DIS model and we therefore only discuss the mean residuals r for the MA model. Figure 5 displays the mean residuals r by estimated student proficiency θ_p (left), skill difficulty β_k (center left), learning rates γ_k (center right) and over time (right). Similarly to the other models, the MA model tends to overestimate the well performing students and underestimate the weaker students (see Fig. 5 (left)). The correlation between estimated student proficiency θ_p and mean residuals r is again strong ($\rho_{MA} = -0.9497$, $p_{MA} < .001$). As for the other models, mean residuals r are uncorrelated to skill difficulty β_k ($\rho_{MA} = 0.2916$, $p_{MA} = .3118$) and to learning rates γ_k ($\rho_{MA} = -0.2993$, $p_{MA} = .2986$). The MA model fits positive learning rates γ_k for all skills k (see Fig. 5 (center right)). To compute the mean residuals r by opportunity count, we again used an averaging window ($n = 10$). Unlike the other models, the MA model overestimates the outcome in the beginning and underestimates it with increasing opportunity count. This result is due to the mastery alignment of the model: As well performing students need less opportunities to master a skill, student attrition occurs in the beginning, where only weaker students remain in the analysis.

We again validated the parameter fit of the different models by predicting the first re-test and comparing prediction accuracy to BKT. Prediction accuracy for the different models is listed in Tab. 1. The AFM performs best for all error measures ($RMSE_{AFM} = 0.3563$, $AUC_{AFM} = 0.6622$). The performance of the LG model ($RMSE_{LG} = 0.3666$, $AUC_{LG} = 0.6602$) is again very close to that of the AFM. Interestingly, the MA model performs well in RMSE ($RMSE_{MA} = 0.3633$) and also exhibits a large AUC ($AUC_{MA} = 0.6508$). The high variances again indicate that differences between the AFM, the LG and the MA models are not significant. The DIS and DISMA models perform considerably worse in RMSE and AUC than the best three regression models. The performance of BKT is similar to the first version of the data set, with an RMSE ($RMSE_{BKT} = 0.3613$) in the range of the best regression models and a significantly lower AUC ($AUC_{BKT} = 0.6102$).

3.2 Generalization to new students

In a second experiment, we investigated how well the different regression models generalize to new students using a student-stratified 10-fold cross validation. For new students (i.e., the students in the test set), the number of opportunities to mastery is not known, therefore only the AFM and the LG model were included in this analysis. Prediction accuracy along with standard deviations for the regression models as well as BKT is listed in Tab. 2. The LG model shows the best performance in all error measures for *Version 1* of the data set. The performance of the AFM is very close to that of the LG model in RMSE ($RMSE_{LG} = 0.4164$, $RMSE_{AFM} = 0.4200$). The high variance indicates that there are no significant differences between the two models regarding RMSE. The AUC of the LG model is, however, considerably higher than that of the AFM ($AUC_{LG} = 0.6931$, $AUC_{AFM} = 0.6693$).

Both regression models again outperform BKT in RMSE ($RMSE_{BKT} = 0.4236$) and AUC ($AUC_{BKT} = 0.6688$), but the high variance indicates that there are no significant differences in RMSE between all three models and also not in AUC between the AFM and the BKT model.

Table 2: Prediction accuracy of student-stratified cross-validation for data set *Version 1* and *2*. The values in brackets denote the standard deviations. The best model per error measure is marked (*).

		RMSE	**Accuracy**	**AUC**
Data set: Version 1	AFM	0.4200 (0.0184)	0.7525 (0.0300)	0.6693 (0.0222)
	LG	0.4164 (0.0175)*	0.7583 (0.0248)*	0.6931 (0.0211)*
	BKT	0.4236 (0.0216)	0.7546 (0.0304)	0.6688 (0.0244)
Data set: Version 2	AFM	0.4008 (0.0247)	0.7850 (0.0296)	0.6755 (0.0335)
	LG	0.3936 (0.0241)*	0.7859 (0.0295)*	0.7199 (0.0260)*
	BKT	0.4032 (0.0241)	0.7849 (0.0297)	0.6810 (0.0289)

The results for *Version 2* of the data set show a similar picture. As expected, all models demonstrate a higher prediction accuracy for *Version 2* of the data set. As this version of the data set includes only students that mastered a skill, overall performance is more homogeneous and therefore prediction is easier.

4. DISCUSSION

AFMs are widely used to analyze and improve student learning [5, 15, 23]. However, AFMs are prone to student attrition when applied to data from mastery learning: As students are aligned by opportunity count, the right hand side of the learning curve fitted by an AFM is dominated by students, who require a large number of opportunities to master a skill, which might in turn lead to underestimation of learning rates γ_k. Indeed, [17] observed that averaging over different students with different initial knowledge states and learning rates may result in aggregated learning curves that appear to show little student learning, even though a mastery learning student model such as BKT identified the students as mastering the skills at runtime. This issue can be solved by using alternative models for fitting the learning curves [17]. Our experiments on data from a mastery learning student model (dynamic Bayesian network) with confirmed learning (significant improvement in external post-tests) support these results: AFM fitted positive learning rates γ_k for about half of the skills and only 70% of the positive γ_k were significantly different from zero. Alternative models, such as the LG and MA models predicted positive learning for all skills and learning rates γ_k and generally showed a higher variance, i.e., learning rates differed from skill to skill. Our results demonstrate that different (although very similar) regression models explain the same data in a different way and that alternative regression models predict different patterns of learning.

Despite the different parameter fits, prediction accuracy of the regression models is very similar. When it comes to generalizing to new students, the LG model shows the most accurate prediction. However, as we observe a high variance in accuracy measures, there is most likely no significant difference in prediction accuracy between the AFM and the LG model. Although the AFM performs best in predicting the first re-test, the high variance of the error measures indicates that there is no significant difference between the AFM, the LG and the MA models. The disaggregated models (DIS, DISMA) perform significantly worse than the other regression models. As the disaggregation into different subpopulations increases the number of parameters, the lower performance of these models might be due to overfitting. This hypothesis is supported by the fact that the disaggregated models outperform the other regression models on the training data set in all error measures. Nonetheless, [17] demonstrated the potential of disaggregated models. Prediction accuracy of these models should therefore be evaluated on larger data sets.

BKT models are outperformed by most of the regression models when it comes to prediction accuracy on unseen data. The AFM and the LG model show a higher accuracy when predicting the first re-test, while the AFM, the LG and the MA model generalize better to new students than BKT. Although these differences are probably not significant (due to the high variance in the error measures), they are still interesting. One reason for this observation might be that BKT does not model forgetting. Our analyses have, however, shown that there is forgetting in the data. As the LG and MA models fit only positive learning rates γ_k, this explanation is only valid for the AFM model. Another reason for the superiority of the logistic regression models could be that traditional BKT does not have any student individualization. However, [26] demonstrated on a different data set that a student individualized parameter p_0 does not lead to significant improvements. The reason for the difference in prediction accuracy between BKT and logistic regression models therefore needs to be investigated further.

5. CONCLUSION

In this work, we presented alternative logistic regression models to AFMs, which are able to adjust for mastery-based data sets. Our results demonstrate that the parameter fits for different (although very similar) regression models vary a lot. We also showed that despite the differences in parameter fit, most of the regression models cannot be distinguished regarding prediction accuracy on unseen data. Finally, our evaluations revealed that logistic regression models outperform BKT, when assessing performance in prediction.

In the future, we would like to further analyze the differences between the proposed modeling techniques. Pre-post gain data might be used to evaluate the different logistic regression models. Is the pre-post gain better predicted by improvement on all skills, as per the LG and MA models, or improvement on a subset of skills, as per the AFM. It could be that, as with the retention measure, AFM somewhat under predicts learning gain and LG somewhat over predicts learning gain. Furthermore, we would like to analyze the differences in prediction between BKT and logistic regression models.

6. ACKNOWLEDGMENTS

This work was funded by the CTI-grant 11006.1.

7. REFERENCES

[1] R. S. Baker, A. T. Corbett, S. M. Gowda, A. Z. Wagner, B. A. MacLaren, L. R. Kauffman, A. P. Mitchell, and S. Giguere. Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. In *Proc. UMAP*, pages 52–63, 2010.

[2] P. Boeck. Random Item IRT Models. *Psychometrika*, 73(4):533–559, 2008.

[3] N. E. Breslow and D. G. Clayton. Approximate Inference in Generalized Linear Mixed Models. *Journal of the American Statistical Association*, 88(421):9–25, 1993.

[4] E. Brunskill. Estimating Prerequisite Structure From Noisy Data. In *Proc. EDM*, pages 217–222, 2011.

[5] H. Cen, K. R. Koedinger, and B. Junker. Is Over Practice Necessary? - Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining. In *Proc. AIED*, pages 511–518, 2007.

[6] H. Cen, K. R. Koedinger, and B. Junker. Comparing Two IRT Models for Conjunctive Skills. In *Proc. ITS*, pages 796–798, 2008.

[7] K.-M. Chang, J. Beck, J. Mostow, and A. Corbett. A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. In *Proc. ITS*, pages 104–113, 2006.

[8] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*, 4(4):253–278, 1994.

[9] J. P. González-Brenes and J. Mostow. Dynamic Cognitive Tracing: Towards Unified Discovery of Student and Cognitive Models. In *Proc. EDM*, pages 49–56, 2012.

[10] J. P. González-Brenes and J. Mostow. Topical Hidden Markov Models for Skill Discovery in Tutorial Data. *NIPS - Workshop on Personalizing Education With Machine Learning*, 2012.

[11] T. Käser, G.-M. Baschera, J. Kohn, K. Kucian, V. Richtmann, U. Grond, M. Gross, and M. von Aster. Design and evaluation of the computer-based training program Calcularis for enhancing numerical cognition. *Front. Psychol.*, 2013.

[12] T. Käser, A. G. Busetto, G.-M. Baschera, J. Kohn, K. Kucian, M. von Aster, and M. Gross. Modelling and optimizing the process of learning mathematics. In *Proc. ITS*, pages 389–398, 2012.

[13] T. Käser, A. G. Busetto, B. Solenthaler, G.-M. Baschera, J. Kohn, K. Kucian, M. von Aster, and M. Gross. Modelling and Optimizing Mathematics Learning in Children. *IJAIED*, 23(1-4):115–135, 2013.

[14] K. Koedinger and E. McLaughlin. Seeing language learning inside the math: Cognitive analysis yields transfer. In *Proc. of the 32nd Annual Conference of the Cognitive Science Society*, pages 471–476, 2010.

[15] K. Koedinger, J. Stamper, E. McLaughlin, and T. Nixon. Using Data-Driven Discovery of Better Student Models to Improve Student Learning. In *Proc. AIED*, pages 421–430, 2013.

[16] J. I. Lee and E. Brunskill. The Impact on Individualizing Student Models on Necessary Practice Opportunities. In *Proc. EDM*, pages 118–125, 2012.

[17] R. Murray, S. Ritter, T. Nixon, R. Schwiebert, R. Hausmann, B. Towle, S. Fancsali, and A. Vuong. Revealing the Learning in Learning Curves. In *Proc. AIED*, pages 473–482, 2013.

[18] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 7:308–313, 1965.

[19] Z. A. Pardos and N. T. Heffernan. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In *Proc. UMAP*, pages 255–266, 2010.

[20] Z. A. Pardos, S. Trivedi, N. T. Heffernan, and G. N. Sárközy. Clustered knowledge tracing. In *Proc. ITS*, pages 405–410, 2012.

[21] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance Factors Analysis - A New Alternative to Knowledge Tracing. In *Proc. AIED*, pages 531–538, 2009.

[22] J. Reye. Student Modelling Based on Belief Networks. *IJAIED*, 14(1):63–96, 2004.

[23] J. C. Stamper and K. R. Koedinger. Human-machine Student Model Discovery and Improvement Using DataShop. In *Proc. AIED*, pages 353–360, 2011.

[24] Y. Wang and N. T. Heffernan. The student skill model. In *Proc. ITS*, pages 399–404, 2012.

[25] M. Wilson and P. De Boeck. Descriptive and explanatory item response models. In P. De Boeck and M. Wilson, editors, *Explanatory item response models: A generalized linear and nonlinear approach*. New York: Springer-Verlag, 2004.

[26] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized Bayesian Knowledge Tracing Models. In *Proc. AIED*, pages 171–180, 2013.

Discovering Gender-Specific Knowledge from Finnish Basic Education using PISA Scale Indices

Mirka Saarela
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
mirka.saarela@gmail.com

Tommi Kärkkäinen
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
tommi.karkkainen@jyu.fi

ABSTRACT

The Programme for International Student Assessment, PISA, is a worldwide study to assess knowledge and skills of 15-year-old students. Results of the latest PISA survey conducted in 2012 were published in December 2013. According to the results, Finland is one of the few countries where girls performed better in mathematics than boys. The purpose of this work is to refine the analysis of this observation by using education data mining techniques. More precisely, as part of standard PISA preprocessing phase certain scale indices are constructed based on information gathered from the background questionnaire of each participating student. The indices describe, e.g., students' engagement, drive and self-beliefs, especially related to mathematics, the main assessment area in PISA 2012. However, around 30% of the scale indices are missing so that a nonstructured sparsity pattern must be dealt with. We handle this using a special, robust clustering technique, which is then applied to Finnish subset of PISA data. Already direct interpretation of the created clusters reveals interesting patterns. Clusterwise analysis through relationship mining refines the confidence on our final conclusion that attitudes towards mathematics which are often gender-specific are the most important factors to explain the performance in mathematics.

Keywords

PISA, robust clustering, frequent itemset, association rule

1. INTRODUCTION

PISA (Programme for International Student Assessment) is an international assessment programme by the Organisation for Economic Co-operation and Development (OECD) that studies students' learning outcomes in reading, mathematics, and scientific literacy triennially. It is referred as the "world's premier yardstick for evaluating the quality, equity and efficiency of school systems" [21]. More than seventy countries and economies have already participated in PISA. Finland has consistently been one of the top-performing countries in the assessment [11]. Each time the study is repeated the main learning outcome focus area changes. In the latest assessment (PISA 2012) it was mathematics. A database of the results is publicly available[1].

One general key finding from PISA 2012 was the gender difference in mathematics performance: On average, boys outperform girls in mathematics. Finland, however, is, according to the assessment, one of the eight countries where girls perform better than boys in mathematics: The mean score of girls in mathematics was 520 while boys had the mean score of 517 [23]. Despite the slightly better performance in mathematics women are, also in Finland, underrepresented in mathematics related jobs [28].

The purpose of this work is to apply educational data mining approch and corresponding techniques to study the performance of Finnish student population in mathematics, focusing especially on gender-related findings. As part of standard PISA preprocessing phase, certain *scale indices* are constructed based on information gathered from the background questionnaire for each participating student [21]. These indices describe, e.g., students' engagement, drive and self-beliefs, especially related to mathematics. However, around 30% of the scale indices are missing due to lack of reliable student responses for the background questions. This means that the knowledge discovery process is realized with data having a nonstructured sparsity pattern. We handle this using a special, robust clustering technique as proposed in [4]. Furthermore, the clustering result obtained is further analyzed using itemset mining [1] to foster the generation of novel information and new knowledge.

The contents of the paper is as follows: First, we provide a short summary on PISA data and how students' capabilities and attributes are presented. We then describe a certain set of scale index variables that are associated with the performance in mathematics. Subsequently, we apply methods from two (see [7] for a complete categorization) of the main branches in educational data mining. In Section 3, we utilize a special clustering approach to find groups of students with similar characteristics with respect to scale indices. In order to further refine the characterization of student groups, we then apply frequent itemset mining and association rule learning to selected clusters in Section 4. Finally, we sum-

[1] See `http://www.oecd.org/pisa/pisaproducts/`.

marize and conclude our study in Section 5.

2. ON PISA DATA

We apply educational data mining for the PISA 2012 data subset of Finland. In each country participating PISA, the schools and students selected for the survey reflect the whole population and characteristics of the educational context. In Finland, 311 schools and 10157 students from these schools were sampled for the assessment in 2012. Out of the sampled students 8829 participated in the actual PISA test. Hereby, each student that takes part has to (i) solve a set of cognitive items/tasks and (ii) fill out one background questionnaire[2] with demographic questions.

Finnish PISA data is stored in two different data sets: One data set includes all the students that participated in the test, and the second one includes all sampled schools. The student data set has more than 600 variables. A set of those variables directly encode the students answers given in the background questionnaire. Moreover, since the participating students should reflect all 15-year-old students in Finland, certain weights are assigned to each student to align the sample with the true population. In PISA reports and learning analysis, student abilities are not given as direct responses to task questions but in the form of the so-called *Plausible Values* (PVs).

Since a very broad domain of knowledge and skills should be tested but the testing time for each student is limited, only certain subsample of students respond to each item/task. In order to reliably compare results of different students, even if they have not answered exactly to the same set of items, PISA uses a generalized form of the Rasch Model [19]. Depending on how many students have solved a task correctly, a certain "difficulty value" is assigned to each tasks and depending on how many tasks a student solved, a certain "competence value" is assigned to each student. PVs are estimated based on difficulty and competence scores and then scaled so that the OECD average in each domain (mathematics, reading and science) is 500 and the standard deviation is 100.

Usually, five PVs are drawn from each student's competence distribution for each main assessment area to describe the performance. For instance, in the Finnish data set for 2012 we have have five PVs for each student in reading, science, and mathematics. Moreover, since mathematics was the main assessment area, five PVs for each of the 7 subscales, i.e. subtopics in mathematics (change and relationship, quantity, space and shape, uncertainty and data, formulate, employ, interpret) are enclosed.

2.1 PISA Scale Indices

PISA scale indices (see Table 1) are derived variables based on information gathered from the background questionnaires. The scale indices are constructed in order to better characterise students dispositions, behaviours, and self-beliefs. Indeed, many of the self-reported indicators of engagement in school are strongly associated with the performance in mathematics. Especially, the *index of economic, social and cultural status* (ESCS) explains 46% of the performance variation among OECD countries so that a socio-economically more advantaged student scores 39 points higher in mathematics[3] than a less advantaged student [20]. Furthermore, according to [19], the ESCS is the "strongest single factor associated with performance in PISA".

Table 1 provides an overview of the PISA scale indices used in this study. In the first two columns, we provide the name of the index and it's abbreviation used in the data set. It should be noted that some indices emphasize negative orientation with respect to mathematics. For example, it usually is not beneficial to the performance in mathematics if a student has a high value in the index which measures the anxiety towards mathematics (ANXMAT). Each index in the PISA data is standardized to have mean zero and scaled to have standard deviation one across OECD countries. Hence, a positive score index does not necessarily mean that a student has replied positively to the corresponding questions but that the answers are above the OECD average.

Correlations between the scale indices and the overall performance in mathematics are provided in the third column in Table 1. In the fourth column, ranking of the correlations based on their absolute values is given. We notice that the three indices having highest linear relationship with performance in mathematics are mathematics specific whereas the fourth index in ranking describes readiness for problem solving, and only the fifth one is the already mentioned status indicator ESCS. The correlations are computed using the subset of Finnish students for which a particular index is available. In order to obtain reliable estimates we have, as recommended in [19], analyzed each PV separately. This means that we have first computed five correlation coefficients and then used their mean as the actual result.

As already observed, not every student in the data set has a value for each of the indices. In fact, 33.24% of the index values are missing/invalid. There are different reasons why a specific scale index for a particular student is unusable. First of all, not all background questions were administered to all students. Students, that were not administered the questions included in the index had missing value by design. Second of all, it might be that the student got the questions but did not answer them. Finally, a reason for a missing index value can be that questions were answered but answers were found to be unreliable or invalid in manual scanning.

3. CLUSTER ANALYSIS USING ROBUST PROTOTYPES

Clustering is an unsupervised data analysis technique, where a given set of objects is divided into subsets (clusters) such that objects in the same cluster are similar to each other and dissimilar to objects in other clusters. Even if this appears as a simple rule, there are many approaches for clustering [10]. The classical division of algorithms is the separation into *partitional* and *hierarchical* clustering methods [16, 29]. Hierarchical clustering is usually applied for small data sets since most of the algorithms have quadratic or higher computational complexity [9]. However, the main difference be-

[2] An example of such background questionnaire can be found from `http://nces.ed.gov/surveys/pisa/pdf/MS12_StQ_FormA_ENG_USA_final.pdf`.

[3] 39 score points equal nearly one year of schooling.

Table 1: PISA scale indices and correlation to mathematics performance

PISA scale index	abbreviation	corr	rank
economic, social and cultural status	ESCS	0.36	5
sense of belonging	BELONG	0.01	15
attitude towards school: learning outcome	ATSCHL	0.15	11
attitude towards school: learning activities	ATTLNACT	0.08	12
perseverance	PERSEV	0.31	6
openness to problem solving	OPENPS	0.42	4
self-responsibility for failing in mathematics	FAILMAT	-0.20	10
interest in mathematics	INTMAT	0.25	7
instrumental motivation to learn mathematics	INSTMOT	0.23	9
self-efficacy in mathematics	MATHEFF	0.51	2
anxiety towards mathematics	ANXMAT	-0.44	3
self-concept in mathematics	SCMAT	0.52	1
behaviour in mathematics	MATBEH	0.04	13
intentions to use mathematics	MATINTFC	0.23	8
subjective norms in mathematics	SUBNORM	-0.02	14

tween these methods is related to the shape of clusters which is readily amplified in the interpretation of the clustering result. Hierarchical clustering is based on connecting locally similar objects so that the global shape of a cluster can be almost arbitrary. Partitional methods, which rely on creating subsets with respect to global similarities, are quaranteed to produce geometrically closed subsets. Moreover, the special prototype characterizing the properties of all the cluster members provides a well-defined pattern for the interpretation of the clustering result.

Prototype-based partitional clustering methods, such as *k-means*, a popular algorithm utilized also in many EDM studies [30], can be described using an iterative relocation algorithmic skeleton with an explicitly defined score function [12] (see Algorithm 1). Partitional clustering creates a $k-$partion $C = \{C_1, ..., C_k\}$ $(k \leq n)$ of data $\mathbf{X}$, such that

1) $C_i \neq \emptyset$ with $i = 1, ..., k$;
2) $\bigcup_{i=1}^{k} C_i = \mathbf{X}$; and
3) $C_i \bigcap C_j = \emptyset$ with $i, j = 1, ..., k$ and $i \neq j$.

In order to realize a prototype-based partitative clustering algorithm some further issues need to be addressed. First of all, all iterative relocation algorithms search better partitions locally so that the final result depends on the initialization. Although a lot of work has been attributed to this problem, still no universal method for identifying the initial partition exists (actually such an approach would provide an approximate solution to the clustering problem itself). Another main issue is to define the similarity measure that reflects the closedness in the data space. To this end, the amount of clusters must be determined in order to end up with one, final clustering result for the interpretation.

Our data to be clustered is problematic, because there is an arbitrary pattern of missing scale indices to deal with. Such missing values could be considered as extreme outliers because they can have any value from each variable's value range. Hence, second order statistics and least-mean-squares estimates that are sensitive to nonnormal degredations are not suitable, and we use instead the so-called nonparametric, robust statistical techniques and distance measures [15, 27, 14]. Out of the simplest robust location estimates, median and spatial median, we use spatial median due to it's multidimensional nature which allows better utilization of the local/clusterwise available data pattern [17]. Spatial median has many attractive statistical properties and, especially, it's breakdown point is 0.5, i.e. it can handle up to 50% of contaminated data.

In [4], a robust approach utilizing the spatial median to cluster sparse and noisy data was introduced. The *k-spatial-medians* clustering algorithm is based on the algorithmic skeleton as presented in Algorithm 1. As the score function one utilizes

$$\mathcal{J} = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \|\boldsymbol{P}_i(\boldsymbol{x}_i - \boldsymbol{c}_j)\|_2, \qquad (1)$$

where the last sum is computed over the subset of data attached to cluster j. Here the projections $\boldsymbol{P}_i, i = 1, \ldots, N$, capture the existing variable values of the ith observation, i.e.

$$(\boldsymbol{P}_i)_j = \begin{cases} 1, \text{if } (\boldsymbol{x}_i)_j \text{ exists,} \\ 0, \text{otherwise.} \end{cases}$$

In Algorithm 1, the projected distance as defined in (1) is used in the first step, and recomputation of the prototypes, as spatial median with the available data, is realized using the SOR (Sequential Overrelaxation) algorithm [4] with the overrelaxation parameter $\omega = 1.5$.

3.1 Initialization and Number of Clusters

It is a well-known problem that all iterative clustering algorithms are highly sensitive to the initial placement of the cluster prototypes and, thus, such algorithms do not guarantee unique clustering [18, 9, 6, 16]. Numerous methods have been introduced to address this problem. Random initialization is still often chosen as the general strategy [31]. However, several researchers (e.g., [3, 5]) report that having some other than random strategy for the initialization often improves final clustering results significantly. Having these issues in mind, we developed the following deterministic and context-sensitive approach to find good initial prototypes.

For a subset of 2520 students in the Finnish data, there are

```
Algorithm 1: Iterative relocation clustering algorithm
Input: Dataset X with $n$ observations and a given number
       of clusters $k$.
Output: A set of $k$ clusters, which minimizes the score
        function.
Select $k$ points as the initial prototypes;
repeat
    1. Assign individual observation to the closest
    prototype;
    2. Recompute the prototypes with the assigned
    observations;
until The partition does not change;
```

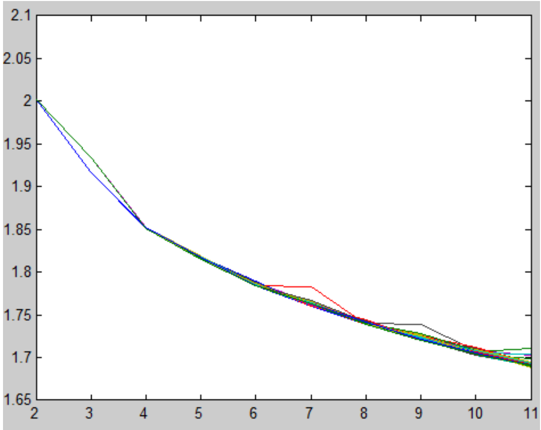


Figure 1: Ray-Turi index for $k = 2, \ldots, 11$

no missing scale index values. For this subset we want to find (i) the most suitable amount of clusters k and (ii) the initial prototypes for the clustering algorithm with the whole data. For this purpose, we utilize a simple search strategy with two nested loops. The first loop iterates through different values of k and the second loop repeats the *k-spatialmedians* algorithm with random initialization ten times. For each clustering result, we then compute the so-called Ray-Turi index, see [25]. This index captures the principal purpose of clustering prototypes, i.e. accurate presentation of separate subset of data, and it is computed by simply dividing the score function (1) with the distance of the two closest prototypes. Figure 1 visualizes the plot of the Ray-Turi index for a set of values for the number of clusters. From the visualization we observe that the clustering result (Ray-Turi index) is decreasing when more clusters are introduced. However, after four clusters the speed of improvement is decreased. Moreover, for four clusters the result is very stable because all the ten random repetitions provide exactly the same clusters and prototypes. To this end, based on these observations, $k = 4$ is used as the number of clusters and the unique result for the full data as initialization for the whole, sparse data set clustering with Algorithm 1. The obtained result, characterized by four prototypes with available value for all scale indices, is to be interpreted next.

3.2 Interpretation of Clustering Result

The four cluster prototypes are depicted in Figure 2. Table 2 provides information about the students in the different clusters. Hereby, *valid indices* shows the percentage of existing index values in each cluster. As can be seen, the available data is quite evenly distributed among the clusters. While *sample size* denotes the actual number of students in the data, *population size of target group* is the same but each student is weighted so that they represent the whole Finnish population of 15-year-old students. *WA math score* is the weighted average of the mathematics scores from the students in the respective cluster.

As can be inferred from Figure 2 in combination with Table 2, we have one clear ”high performance” and one clear ”low performance” national cluster: The students in *Cluster 1* have mean performance in mathematics of 571.53 and they are on average the most advantaged students with highest beliefs in themselves. In all indices that are associated with highperformance in mathematics, the prototype that represents this cluster has the highest value. Solely in the ”intentions” to use mathematics later in their life, the students in *Cluster 1* lack behind the students in *Cluster 3*. *Cluster 4*, on the other hand, represents the most disadvantaged students in Finland, with lowest mean score in mathematics, and also lowest beliefs in themselves.

Cluster 2 and *Cluster 3* are, at the same time, similar and very different. According to the average performance of the students in those two clusters, both belong to PISA score Level 3 (see Table 4). As specified in the proficiency level descriptions in [22] this means that students in both of these clusters are able to, for example, solve tasks with clearly described procedures, but are unlikely to be able to (this proficiency is attributed to students from higher levels) also solve tasks that involve constraints or call for making assumptions. However, the prototypes (see Figure 2) show that students from these clusters can be opposite to each other by means of many scale indices.

While the students in *Cluster 2* generally are slightly more socially and economically advantaged, feel that they belong to school, and commonly have very positive attitude towards school, they definitely have below OECD average intentions to use mathematics, so that they also score worse in mathematics. *Cluster 2* is predominantly populated by girls. *Cluster 3*, on the other hand, has the lowest percentage of girls in it. This cluster consists of mostly boys who do not have the best attitude towards school. They also do not feel like they belong to school and generally are socially and economically less advantaged than the students in *Clusters 1* and *2*. However, they have the highest intentions to use mathematics later in their life, and pursue mathematics-related studies or careers in the future. They also tend to attribute failure in mathematics more to external factors than to themselves, have less anxiety towards mathematics than the OECD average, and are (although they do not seem to be interested in school in general) more interested in mathematics than the OECD average. It seems that they have already decided to have a career in a mathematics related profession, on the contrary to the (mostly female) students in *Cluster 2*.

As for the correlations before, we also created a ranking of indices to clarify the interpretation of the clustering result. The distance that defines the ranking to distinguish *Clusters 2* and *3* is just the absolute difference between the

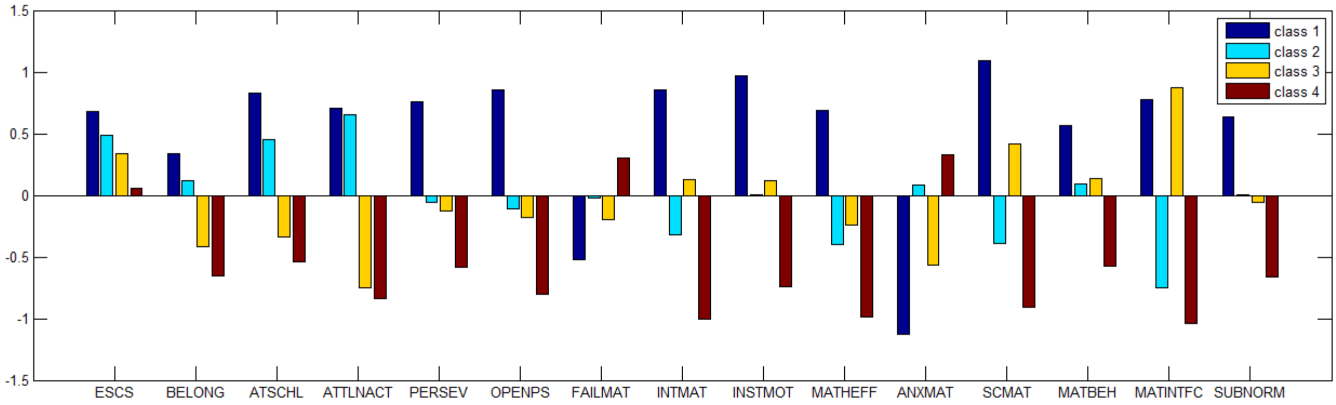


Figure 2: Clustering results

Table 2: Facts of clusters

	valid	sample	**population size of target group**			**WA math score**		
cluster	indices	size	all	♀ (in %)	♂	∅	♀	♂
C1	64%	1967	12884	5302 (41%)	7582	571.53	578.66	566.55
C2	69%	2192	14038	8598 (61%)	5440	509.82	516.76	498.85
C3	67%	2450	16751	6434 (38%)	10317	536.02	541.74	532.45
C4	66%	2220	16374	8876 (54%)	7498	467.21	472.96	460.40
C1-C4	67%	8829	60047	29210 (49%)	30837	518.75	520.19	517.39

Table 3: Separation of clusters

	all clusters		**Cluster 2 -3**	
index	distance	rank	distance	rank
ESCS	0.62	15	0.15	10
BELONG	0.98	13	0.53	6
ATSCHL	1.38	9	0.78	4
ATTLNACT	1.54	7	1.40	2
PERSEV	1.35	10	0.07	13
OPENPS	1.66	6	0.08	12
FAILMAT	0.83	14	0.17	8
INTMAT	1.86	3	0.44	7
INSTMOT	1.71	4	0.11	11
MATHEFF	1.68	5	0.16	9
ANXMAT	1.46	8	0.65	5
SCMAT	2.00	1	0.81	3
MATBEH	1.14	12	0.04	15
MATINTFC	1.91	2	1.63	1
SUBNORM	1.30	11	0.06	14

index values of the two prototypes. This is generalized as the distance between *all clusters* by simply summing the three absolute differences between individually ordered prototype indices. These two distances and the implied rankings are provided in Table 3. As can be seen from Table 3, the students' self-concept in mathematics, the index which also correlates the most with the performance in mathematics (see Table 1), discriminates all the clusters the most. It seems that students' beliefs in their own mathematics abilities capture their true knowledge and skills fairly well. Additionally, the intentions to use mathematics and the interest in this subject provide a good separation of the four clusters. Those two indices describe the students' drive and interest to learn mathematics because they perceive this subject as profitable and appealing to their future. The two interesting clusters, *Cluster 2* and *Cluster 3*, are separated the most by the intentions to pursue a career in mathematics and by the attitudes towards school concerning learning activities.

4. ASSOCIATION RULE DISCOVERY

The goal of association rule mining, one of the most utilized methods in EDM according to [8, 26], is to automatically find patterns that describe strongly associated attributes in data. The discovered patterns are usually represented in the form of implication rules or attribute subsets [1, 32]. We have two explicit clusters - *Cluster 1* which consists of the highest performing students and *Cluster 4* which consists of the lowest performing students - but for the two remaining clusters with mixed profile, *Cluster 2* and *Cluster 3*, we want to find patterns/rules that further characterize these students. Hence, we form for each student that belongs to one of these two clusters an itemset which contains the gender of the student (first subset in Table 4), all the scale indices (central subset in Table 4), and the categorized proficiency level in mathematics (last subset in this table).

PISA score levels define the performance level of the students. For example, for PISA 2012 the range of difficulty of tasks generates six levels of mathematics proficiency. Students with a performance score within the range of Level 1 are likely to be able to successfully complete Level 1 tasks, but are unlikely to be able to complete tasks at higher levels. Level 6 reflects tasks that are the most difficult in terms of mathematical skills and knowledge [22]. On average, both student clusters of interest belong to performance Level 3 (see Table 2). Therefore, in the corresponding item, we only distinguish three categories: below, within, or above Level 3 (see the last subset in Table 4).

Table 4: Items for Association Rules

id	item
1	girl
2	boy
3 & 4	$(+,-)$ ESCS
5 & 6	$(+,-)$ BELONG
7 & 8	$(+,-)$ ATSCHL
9 & 10	$(+,-)$ ATTLNACT
11 & 12	$(+,-)$ PERSEV
13 & 14	$(+,-)$ OPENPS
15 & 16	$(+,-)$ FAILMAT
17 & 18	$(+,-)$ INTMAT
19 & 20	$(+,-)$ INSTMOT
21 & 22	$(+,-)$ MATHEFF
23 & 24	$(+,-)$ ANXMAT
25 & 26	$(+,-)$ SCMAT
27 & 28	$(+,-)$ MATBEH
29 & 30	$(+,-)$ MATINTFC
31 & 32	$(+,-)$ SUBNORM
33	Level 2 or below: ≤ 482.38
34	Level 3: $482.38 - 544.68$
35	Level 4 or above: ≥ 544.68

In order to separate an individual student from main bulk of students, we fix a threshold value of 0.2 to define whether an item is part of the itemset for that particular student. The threshold 0.2 is chosen because it provides the median (rounded to one decimal place) of the absolute values of scale indices of all cluster prototypes. If a positive index value for a certain student is above the threshold, then the first *id* in the matrix (see Table 4) will be part of the itemset. Similarly, if a negative index value is below the negative threshold, then the second *id* (see Table 4) will belong to the itemset. Again, we utilize only the available indices. This means that in case the student's index value is inside $[-0.2, 0.2]$ or missing/invalid, it is not included in the itemset. For finding frequent itemsets based on the encoding, we used the implementation described in [13], and for generating association rules from the obtained frequent itemsets we utilized the implementation explained in [2].

4.1 Basic Concepts of Frequent Itemsets

Let I be the set of all items. An important property of an itemset is its *support count*, which refers to the number of transactions that contain a particular itemset. Let S_1 be a subset of the set of items ($S_1 \subseteq I$). Logically, a transaction $t_i \in T$, where T denotes the set of all transactions, is said to contain itemset S_1 if S_1 is a subset of t_i. Mathematically, the support count, $\sigma(S_1)$, for an itemset S_1 can be stated as follows:

$$\sigma(S_1) = |\{t_i \mid S_1 \subseteq t_i, t_i \in T\}|,$$

where $|\cdot|$ stands for the number of elements in a set. An *Association Rule* is then an implication expression of the form $S_1 \rightarrow S_2$, where $S_1, S_2 \subseteq I$ and $S_1 \cap S_2 = \emptyset$.

The support, $s(S_1 \rightarrow S_2)$, determines how often a rule is applicable to a given data set. Furthermore, the confidence, $c(S_1 \rightarrow S_2)$, determines how frequently items in S_2 appear in the transactions that contain S_1. Mathematically this can be expressed as follows:

$$s(S_1 \rightarrow S_2) = \frac{\sigma(S_1 \cup S_2)}{|T|} \text{ and } c(S_1 \rightarrow S_2) = \frac{\sigma(S_1 \cup S_2)}{\sigma(S_1)},$$

Support measures how well a rule is covered by the data. Therefore, if a rule has a too low support, it could be that it occurred solely by chance. Confidence is an important measures as it provides the the reliability and accuracy of a rule.

4.2 Obtained Rules and Interpretation

When we use the applied implementation of the famous Apriori Algorithm, we obtain many trivial rules. For example, it is already obvious from the clustering prototypes that those students who have highly positive attitude towards learning activities have also highly positive attitude towards learning outcomes. However, as already discussed, our itemsets can be divided into three subsets: the set that contains the gender, the set which contains the performance in mathematics, and the set which contains the different scale indices. We are interested in the gender differences and the performance in mathematics. Therefore, we search inside the algorithm's output for rules that have items of the gender and/orperformance interval subsets at the right hand side of the rule.

We start with high values for support and confidence and lower then the confidence threshold. Since we are especially interested in rules that contain the gender, the support has to have a relatively small value, so we choose the minimum value 0.1 while trying to keep the confidence value as high as possible. Starting with confidence of 1 and lowering it successively, we obtain the first rule that has gender on the right side with confidence 0.71:

$$\{\text{-ATTLNACT, +SCMAT, +MATINTFC}\} \Rightarrow \{boy\} \tag{2}$$

In words (2) means that those students who have negative attitudes towards school but a high self-concept and high intentions in mathematics are boys.

The first rule that we obtain for girls with confidence 0.69 is of the form:

$$\{\text{ -MATHEFF, - MATINTFC}\} \Rightarrow \{girl\} \tag{3}$$

Rule (3) says that those students who have negative self-efficacy and no intention to use mathematics are girls.

If we lower the minimal acceptable support into 0.095, we obtain the following interesting rule (4): Those students who have positive attitudes towards school but no intention to use mathematics later in life are girls.

$$\{\text{+ATTLNACT, -MATINTFC}\} \Rightarrow \{girl\} \tag{4}$$

Next, with the same minimal support we are searching explicitly for rules that have performance value below or above Level 3 at the left-hand side of the rule and gender at the right-hand side. Here, we first obtain the following rule with a confidence value of 0.6:

$$\{\text{+ATTLNACT, above Level 3 performance}\} \Rightarrow \{girl\} \tag{5}$$

According to (5), those students with a proficiency level above 3 and a clearly above average positive attitude towards learning activities in school are girls.

With confidence 0.52 we obtain the first rule for boys:

$$\{\text{+SCMAT, above Level 3 performance}\} \Rightarrow \{boy\} \tag{6}$$

Rule (6) means that those students with a proficiency level above 3 and a clear above average self-concept in mathematics are boys.

Subsequently, we are searching for rules wich have both gender and below or above Level 3 performance at the left-hand side of the rule. Such rule with the highest confidence (0.65) reads as:

$$\begin{gathered}\{\text{-ATSCHL -ATTLNACT +OPENPS -FAILMAT} \\ \text{+SCMAT}\} \Rightarrow \{boy, \text{above Level 3 performance}\}\end{gathered} \tag{7}$$

According to (7), those students with negative attitudes towards school (both, learning outcome as well as learning activities) but with clearly above average openness to problem solving, a high self-concept in mathematics and strictly below average self-responsibility for failing in mathematics, are boys that perform above Level 3.

For girls the rule with the highest confidence (0.63) is given by (8):

$$\begin{gathered}\{\text{-ESCS +ATTLNACT +ANXMAT -SCMAT}\} \\ \Rightarrow \{girl, \text{below Level 3 performance}\}\end{gathered} \tag{8}$$

This means that those students who are socially and economically less advantaged, have high anxiety towards mathematics and a low self-concept in mathematics, but still clearly above average attitude towards school, are girls who perform below Level 3.

If we unite the rules given in (2)-(8), we see that in all the rules that contain boys the item which represents the high self-concept in mathematics is present. In general, high-performing boys are also convinced that they can succeed (see 6). Moreover, even when they fail in mathematics, they are more likely to see other individuals or factors responsible on this than themselves (see 7). In addition, they have the highest intentions to use mathematics later in their life (see 2). However, according to the rules, male students can have negative attitude towards school (see 2 and 8), whereas the most positive attitudes appear only in the rules that include girls. Even the below average performing and socially and economically more disadvantaged girls with low self-concept and high anxiety towards mathematics, perceive the learning activities in their schools as very important (see 8). The same positive attitude towards school is also associated with the highest performing girls (see 5). Moreover, female students are much less confident about their mathematic skills (see 3) and have least intentions to pursue a mathematics related career (see 3 and 4).

To sum up, we conclude that specific characteristics and attitudes in the two middle performing clusters are, indeed, often gender-specific. Since we explicitly searched for rules that have certain items in them, we can not express precisely how typical these situations are. Nevertheless, when we combine all obtained rules with the clustering result two main characterizations appear: On the one hand, we have a specific subgroup of mainly girls who we nominated "to-be-nurses": they seem to be capable of performing well if they want to, having strongly positive attitude towards school. However, these students have low beliefs in themselves to be able to succeed in mathematics, and even a somewhat fear towards mathematics. On the other hand, we have a subgroup of mainly boys which we refer as "to-be-engineers". These students do not seem very interested in school in general. Yet, they trust in their capabilities and are extremely confident about their skills to perform well in mathematics. Even if they fail, they attribute this failure rather to other external factors than to themselves.

5. SUMMARY AND CONCLUSIONS

Although Finland is one of the few countries in which, on average, girls perform slightly better than boys in mathematics, professional careers related to this subject are also in here still dominated by men. We have applied methods from two of the main educational data mining branches on PISA data to obtain more gender-specific knowledge which might explain this observation.

First of all, we utilized a special robust clustering approach to group the students according to those PISA scale indices that are associated with performance in mathematics. The index that represents the students' self-concept in mathematics (SCMAT), and which also was the variable that correlates the most with the students' performance in mathematics (see Table 1), is the most important discriminator for the four clusters that we obtained (see Table 3). Combined with the other attributes we conclude that those students who have a higher self-concept, and tend to be socially and economically more advantaged, perform better than their less advantaged peers. They also have better attitudes to school, trust more in their own capabilities, and have greater expectation for their future careers (see Figure 2).

Two of the clusters we obtained, *Cluster 1* representing the "high performing" and *Cluster 4* representing the "low performing" students, can to a large extend be explained by these differences. However, the two "medium" clusters show the opposite behaviour: Socially and economical more advantaged students with very positive attitudes towards school and learning from *Cluster 2* perform worse in mathematics than the somewhat more disadvantaged students in *Cluster 3*. We found that these clusters are separated the most by the index that measures the student's intentions to pursue a mathematics related career. Since *Cluster 2* is with 61% dominated by girls, while *Cluster 3* consists of a larger percentage (62%) of boys we assumed that this difference could be explained by the gender of the student.

Association rule mining in the data subset of these two remaining medium clusters revised the gender-specific attitudes even more, and confirmed our assumption. Those 15-year-old students from this subset who already seem to have decided to pursue a mathematics related career are mostly boys. On the other hand, the attribute that is the most ascribable to girls is the positive attitude towards school. Altogether, the results of our study suggest that there are distinct groups of high and low performing students. However, the bulk of the girls with average performance seem to have no intentions to pursue a mathematics related profession. This is neither connected to their social status nor to their attitudes towards school. In fact, they often show a better feeling of belonging to school and have very positive attitudes towards school and learning. While boys often consider mathematics as a great part of their future even when they do not show obvious skills, girls tend to be discouraged much faster and to easier favour other subjects. We feel that this is an important finding that should be studied further, especially concerning when such a gender-specific orientation starts to emerge.

6. REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM, 1993.

[2] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[3] R. T. Aldahdooh and W. Ashour. DIMK-means ”Distance-based initialization method for K-means clustering algorithm”. *International Journal of Intelligent Systems and Applications (IJISA)*, 5(2):41, 2013.

[4] S. Äyrämö. *Knowledge Mining Using Robust Clustering*, volume 63 of *Jyväskylä Studies in Computing*. University of Jyväskylä, 2006.

[5] L. Bai, J. Liang, and C. Dang. An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data. *Knowledge-Based Systems*, 24(6):785–795, 2011.

[6] L. Bai, J. Liang, C. Dang, and F. Cao. A cluster centers initialization method for clustering categorical data. *Expert Systems with Applications*, 39(9):8022–8029, 2012.

[7] R. Baker et al. Data mining for education. *International Encyclopedia of Education*, 7:112–118, 2010.

[8] R. Baker and K. Yacef. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, 1, 2010.

[9] M. Emre Celebi, H. A. Kingravi, and P. A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 2012.

[10] V. Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, 2002.

[11] A. L. Goodwin. Perspectives on high performing education systems in Finland, Hong Kong, China, South Korea and Singapore: What lessons for the US? In *Educational Policy Innovations*, pages 185–199. Springer, 2014.

[12] J. Han, M. Kamber, and A. Tung. Spatial clustering methods in data mining: A survey, 2001.

[13] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1):53–87, 2004.

[14] T. P. Hettmansperger and J. W. McKean. *Robust nonparametric statistical methods*. Edward Arnold, London, 1998.

[15] P. J. Huber. *Robust Statistics*. John Wiley & Sons Inc., New York, 1981.

[16] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[17] T. Kärkkäinen and E. Heikkola. Robust formulations for training multilayer perceptrons. *Neural Computation*, 16:837–862, 2004.

[18] M. Meilă and D. Heckerman. An experimental comparison of several clustering and initialization methods. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 386–395. Morgan Kaufmann Publishers Inc., 1998.

[19] OECD. *PISA Data Analysis Manual: SPSS and SAS, Second Edition*. OECD Publishing, 2009.

[20] OECD. *PISA 2012 Results: Excellence Through Equity: Giving Every Student the Chance to Succeed (Volume II)*. PISA, OECD Publishing, 2013.

[21] OECD. *PISA 2012 Results: Ready to Learn - Students' Engagement, Drive and Self-Beliefs (Volume III)*. PISA, OECD Publishing, 2013.

[22] OECD. *What Makes Schools Successful? Resources, Policies and Practices (Volume IV)*. PISA, OECD Publishing, 2013.

[23] OECD. *PISA 2012 Results: What Students Know and Can Do. Student Performance in Mathematics, Reading and Science (Volume I, Revised edition, February 2014)*. PISA, OECD Publishing, 2014.

[24] N. Raheja and R. Kumar. Optimization of association rule learning in distributed database using clustering technique. *International Journal on Computer Science & Engineering*, 4(12), 2012.

[25] S. Ray and R. H. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, pages 137–143, 1999.

[26] C. Romero and S. Ventura. Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(6):601–618, 2010.

[27] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons Inc., New York, 1987.

[28] M. Saari. Promoting gender equality without a gender perspective: Problem representations of equal pay in Finland. *Gender, Work & Organization*, 20(1):36–55, 2013.

[29] M. Steinbach, L. Ertöz, and V. Kumar. The challenges of clustering high dimensional data. In *New Directions in Statistical Physics*, pages 273–309. Springer, 2004.

[30] B. Xu, M. Recker, X. Qi, N. Flann, and L. Ye. Clustering educational digital library usage data: A comparison of latent class analysis and k-means algorithms. *Journal of Educational Data Mining*, 5(2):38–68, 2013.

[31] R. Xu and D. C. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[32] Q. Zhao and S. S. Bhowmick. Association rule mining: A survey. *Nanyang Technological University, Singapore*, 2003.

EduRank: A Collaborative Filtering Approach to Personalization in E-learning

Avi Segal, Ziv Katzir, Ya'akov (Kobi) Gal,
Guy Shani
Dept. of Information Systems Engineering
Ben-Gurion University, Israel

Bracha Shapira
Dept. of Information Systems Engineering and
Telekom Innovation Laboratories
Ben-Gurion University, Israel

ABSTRACT

The growing prevalence of e-learning systems and on-line courses has made educational material widely accessible to students of varying abilities, backgrounds and styles. There is thus a growing need to accomodate for individual differences in such e-learning systems. This paper presents a new algorithm for personliazing educational content to students that combines collaborative filtering algorithms with social choice theory. The algorithm constructs a "difficulty" ranking over questions for a target student by aggregating the ranking of similar students, as measured by different aspects of their performance on common past questions, such as grades, number of retries, and time spent solving questions. It infers a difficulty ranking directly over the questions for a target student, rather than ordering them according to predicted performance, which is prone to error. The algorithm was tested on two large real world data sets containing tens of thousands of students and a million records. Its performance was compared to a variety of personalization methods as well as a non-personalized method that relied on a domain expert. It was able to significantly outperform all of these approaches according to standard information retrieval metrics. Our approach can potentially be used to support teachers in tailoring problem sets and exams to individual students and students in informing them about areas they may need to strengthen.

1. INTRODUCTION

Education is increasingly mediated by technology, as attested by the prevalence of educational software in schools and the explosion of on-line course opportunities. As a result, educational content is now accessible to student communities of varied backgrounds, learning styles and needs. There is thus a growing need for personalizing educational content to students in e-learning systems in a way that adapts to students' individual needs [20, 1]. A popular approach towards personalization in e-learning is to sequence students' questions in a way that best matches their learning styles or gains [2, 28].

This paper provides a novel algorithm for sequencing content in e-learning systems that directly creates a "difficulty ranking" over new questions. Our approach is based on collaborative filtering [6], which generates a difficulty ranking over a set of questions for a target student by aggregating the known difficulty rankings over questions solved by other, similar students. The similarity of other students to the target student is measured by their grades on common past question, the number of retries for each question, and other features. Unlike other uses of collaborative filtering in education, our approach directly generates a difficulty ranking over the test questions, without predicting students' performance directly on these questions, which may be prone to error.[1]

Our algorithm, called EduRank, weighs the contribution of these students using measures from the information retrieval literature. It allows for partial overlap between the difficulty rankings of a neighboring student and the target student, making it especially suitable for e-learning systems where students differ in which questions they solve. The algorithm extends a prior approach for ranking items in recommendation systems [15], which was not evaluated on educational data, in two ways: First, by using social choice theory to combine the difficulty rankings of similar students and produce the best difficulty ranking for the target student. Second, EduRank penalizes disagreements in high positions in the difficulty ranking more strongly than low positions, under the assumption that errors made in ranking more difficult questions are more detrimental to students than errors made in ranking of easier questions.

We evaluated EduRank on two large real world data sets containing tens of thousands of students and about a million records. We compared the performance of EduRank to a variety of personalization methods from the literature, including the prior approach mentioned above as well as other popular collaborative filtering approaches such as matrix factorization and memory-based K nearest neighbours. We also compared EduRank to a (non-personalized) ranking created by a domain expert. EduRank significantly outperformed all other approaches when comparing the outputted difficulty rankings to a gold standard.

The contribution of this paper is two-fold. First, we present a novel algorithm for personalization in e-learning according to the level of difficulty by combining collaborative filtering with social choice. Second, we outperform alternative solutions from the literature on two real-world data sets. Our approach can potentially be used to support both teachers and students, by automatically tailoring problem sets or exams to the abilities of individual students in the classroom, or by informing students about topics which they need to strengthen. Lastly, it can also augment existing ITS systems by integrating a personalized order over questions into the interaction process with the student.

[1] To illustrate, in the KDD cup 2010, the best preforming grade prediction algorithms exhibited prediction errors of about 28% [25]

2. BACKGROUND

In this section we briefly review relevant approaches and metrics in recommendation systems, social choice, and information retrieval.

2.1 Recommendation Systems and Collaborative Filtering

Recommender systems actively help users in identifying items of interest. For example, the prediction of users' ratings for items, and the identification of the top-N relevant items to a user, are popular tasks in recommendation systems. A commonly used approach for both tasks is Collaborative Filtering (CF), which uses data over other users, such as their ratings, item preferences, or performance in order to compute a recommendation for the active user.

There are two common collaborative filtering approaches [6]; In the memory-based K nearest neighbor approach, a similarity metric, such as the Pearson correlation, is used to identify a set of neighboring users. The predicted rating for a target user and a given item can then be computed using a weighted average of ratings of other users in the neighborhood. In the model based approach, a statistical model between users and items is created from the input data. For example, the SVD approach [21] computes a latent feature vector for each user and item, such that the inner product of a user and item vectors is higher when the item is more appropriate for the user.

While rating prediction and top-N recommendations are widely researched, not many recommendation system applications require ranking. Thus, there were only a few attempts to use CF approaches to generate rankings. Of these, most methods order items for target users according to their predicted ratings. In contrast, Liu et al. developed the EigenRank algorithm [15] which is a CF approach that relies on the similarity between item ratings of different users to directly compute the recommended ranking over items. They show this method to outperform existing collaborative filtering methods that are based on predicting users' ratings.

Using the ratings of similar users, EigenRank computes for each pair of items in the query test set so-called potential scores for the possible orderings of the pair. Afterward, EigenRank converts the pair-wise potentials into a ranked list. EigenRank was applied to movie recommendation tasks, and was shown to order movies by rating better than methods based on converting rating predictions to a ranked list.

2.2 Social Choice

Social Choice Theory originated in economics and political science, and is dealing with the design and formal analysis of methods for aggregating preferences (or votes) of multiple agents [11]. Examples of such methods include voting systems used to aggregate preferences of voters over a set of candidates to determine which candidate(s) should win the election, and systems in which voters rank a complete set of candidates using an ordinal scale. One such approach which we use in this paper is Copeland's method [8, 17] ordering candidates based on the number of pairwise defeats and victories with other candidates.

The Copland score for an alternative q_j is determined by taking the number of those alternatives that q_j defeats and subtracting from this number those alternatives that beat q_j. A partial order over the items can then be inferred from these scores. Two advantages of this method that make it especially amenable to e-learning systems with many users (e.g., students and teachers) and large data sets are that they are quick to compute and easy to explain to users [22]. Pennock et al. [19] highlighted the relevance of social choice to CF and the importance of adapting weighted versions of voting mechanisms to CF algorithms. Our algorithm represents an application of this approach to e-learning systems.

2.3 Metrics for Ranking Scoring

A common task in information retrieval is to order a list of results according to their relevance to a given query [29]. Information retrieval methods are typically evaluated by compering their proposed ranking to that of a gold standard, known as a "reference ranking", which is provided by the user or by a domain expert.

Before describing the comparison metrics and stating their relevance for e-learning systems, we define the following notations: Let $\binom{L}{2}$ denote the set of all non ordered pairs in L. Let $\succ$ be a partial order of a set of questions L. We define the reverse order of $\succ$ over L, denoted $\overline{\succ}$ as a partial order over L such that if $q_j \succ q_k$ then $q_k \overline{\succ} q_j$. Let $\succ_1$ and $\succ_2$ be two partial orders over a set of questions L, where $\succ_1$ is the reference order and $\succ_2$ is the system proposed order. We define an agreement relation between the orders $\succ_1$ and $\succ_2$ as follows:

- The orders $\succ_1$ and $\succ_2$ *agree* on questions q_j and q_k if $q_j \succ_1 q_k$ and $q_j \succ_2 q_k$.
- The orders $\succ_1$ and $\succ_2$ *disagree* on questions q_j and q_k if $q_j \succ_1 q_k$ and $q_k \succ_2 q_j$.
- The orders $\succ_1$ and $\succ_2$ *are compatible* on questions q_j and q_k if $q_j \succ_1 q_k$ and neither $q_j \succ_2 q_k$ nor $q_k \succ_2 q_j$.

Given a partial order $\succ$ over questions Q, the restriction of $\succ$ over $L \subseteq Q$ are all questions (q_k, q_l) such that $q_k \succ q_l$ and $q_k, q_l \in L$.

2.3.1 Normalized Distance based Performance

The Normalized Distance based Performance Measure (NDPM) [26, 24] is a commonly used metric for evaluating a proposed system ranking to a reference ranking . It differentiates between correct orders of pairs, incorrect orders and ties. Formally, let $\delta_{\succ_1,\succ_2}(q_j, q_k)$ be a distance function between a reference ranking $\succ_1$ and a proposed ranking $\succ_2$ defined as follows:

$$\delta_{\succ_1,\succ_2}(q_j,q_k) = \begin{cases} 0 & \text{if } \succ_1 \text{ and } \succ_2 \text{ agree on } q_j \text{and } q_k, \\ 1 & \text{if } \succ_1 \text{ and } \succ_2 \text{ are compatible on } q_j \text{and } q_k, \\ 2 & \text{if } \succ_1 \text{ and } \succ_2 \text{ disagree on } q_j \text{and } q_k. \end{cases} \tag{1}$$

The total distance over all question pairs in L is defined as follows

$$\beta_{\succ_1,\succ_2}(L) = \sum_{(q_j,q_k)\in\binom{L}{2}} \delta_{\succ_1,\succ_2}(q_j,q_k) \tag{2}$$

Let $m(\succ_1) = \text{argmax}_{\succ} \beta_{\succ_1,\succ}(L)$ be a normalization factor which is the maximal distance that any ranking $\succ$ can have from a reference ranking $\succ_1$. The NDPM score $s_{ND}(L, \succ_1, \succ_2)$ comparing a proposed ranking of questions $\succ_2$ to a reference ranking $\succ_1$ is defined as

$$s_{ND}(L,\succ_1,\succ_2) = \frac{\beta_{\succ_1,\succ_2}(L)}{m(\succ_1)} \tag{3}$$

Intuitively, the NDPM measure will give a perfect score of 0 to difficulty rankings over the set in L that completely agree with the reference ranking, and a worst score of 1 to a ranking that completely disagrees with the reference ranking. If the proposed ranking does not contain a preference between a pair of questions that are ranked in the reference ranking, it is penalized by half as much as providing a contradicting preference.

The evaluated ranking is not penalized for containing preferences that are not ordered in the reference ranking. This means that for any question pair that were not ordered in the true difficulty ranking, any ordering predicted by the ranking algorithm is acceptable. Not penalizing unordered pairs is especially suitable for e-learning systems in which some questions for the target student in L may not have been solved by other students and these questions may remain unordered in the difficulty ranking.

2.3.2 AP Rank Correlation

A potential problem with the NDPM metric is that it does not consider the location of disagreements in the reference ranking. In some cases it is more important to appropriately order items that should appear closer to the head of the ranked list, than items that are positioned near the bottom. For example, when ranking movies, it may be more important to properly order the movies that the user would enjoy, than to properly order the movies that the user would not enjoy. Similarly, we assume that the severity of errors in ranking questions depends on their position in the ranked list. As we are interested in sequencing questions by order of difficulty, properly predicting how easy questions should be ordered is not as important as avoiding the presentation of a difficult question too early, resulting in frustration and other negative effects on the student learning process. Therefore, when evaluating a ranked list of questions, it is often important to consider the position of the questions in the ranked list. We would like to give different weights to errors depending on their position in the list.

To this end, we can use the AP correlation metric [27], which gives more weight to errors over items that appear at higher positions in the reference ranking. Formally, let $\succ_1$ be the reference ranking and $\succ_2$ be a proposed ranking over a set of items. The AP measure compares the order between each item in the proposed ranking $\succ_2$ with all items that precede it with the ranking in the reference ranking $\succ_1$.

For each $q_k, q_j \in L, k \neq j$, let the set $Z^k(L, \succ_2)$ denote all question pairs (q_k, q_j) in L such that $q_j \succ_2 q_k$. These are all the questions that are more difficult to the student than question q_k.

$$Z^k(L, \succ_2) = \{(q_j, q_k) \mid \forall q_j \neq q_k \text{ s.t. } q_j \succ_2 q_k \text{ and } q_j, q_k \in L\} \tag{4}$$

We define the indicator function $I^A(q_j, q_k, \succ_1, \succ_2)$ to equal 1 when $\succ_1$ and $\succ_2$ agree on questions q_j and q_k.

Let $A^k(L, \succ_1, \succ_2)$ be the normalized agreement score between $\succ_2$ and the reference ranking $\succ_1$ for all questions q_j such that $q_j \succ_i q_k$.

$$A^k(L, \succ_1, \succ_2) = \frac{1}{k-1} \sum_{(q_j, q_k) \in Z^k(L, \succ_2)} I^A(q_j, q_k, \succ_1, \succ_2) \tag{5}$$

The AP score of a partial order $\succ_2$ over L given partial order $\succ_1$ is defined as

$$s_{AP}(L, \succ_1, \succ_2) = \frac{1}{|L|-1} \sum_{k \in |L|} A^k(L, \succ_1, \succ_2) \tag{6}$$

The s_{AP} score gives a perfect score of 1 to systems where there is total agreement between the system proposed difficulty ranking and the reference ranking for every question pair above location i for all $i \in L$. The worst score of 0 is given to systems were there is no agreement between the two ranked lists.

3. PROBLEM DEFINITION AND APPROACH

We now formalize our problem and the approach used. The "difficulty ranking problem" includes a target student s_i, and a set of questions L_i, for which the algorithm must predict a difficulty ranking $\hat{\succ}_i$ over L_i. The predicted difficulty ranking $\hat{\succ}_i$ is evaluated with respect to a difficulty reference ranking $\succ_i$ over L_i using a scoring function $s(\hat{\succ}_i, \succ_i, L_i)$.

To solve this problem, we take a collaborative filtering approach, which uses the difficulty rankings on L_i of other students similar to s_i to construct a difficulty ranking over L_i for student s_i. Specifically, the input to the problem includes: (1) A set of students $S = \{s_1, s_2, ..., s_m\}$; (2) A set of questions $Q = \{q_1, q_2, ..., q_n\}$; (3) For each student $s_j \in S$, a partial difficulty ranking $\succ_j$ over a set of questions $T_j \subseteq Q$.

For every student $s_j \in S$ there are two disjoint subsets $T_j, L_j \in Q$, where the difficulty ranking of s_j over T_j is known, and is a restriction of $\succ_j$ over all the questions in Q. Intuitively, for a a target student $s_i \in S$, T_i represent the set of questions that the target student s_i has already answered, while L_i is the set of questions for which a difficulty ranking needs to be produced.

The collaborative filtering task is to leverage the known rankings of all students s_j over T_j in order to compute the required difficulty ranking $\hat{\succ}_i$ over L_i for student s_i.

4. THE EDURANK ALGORITHM

We now present our EduRank algorithm for producing a personalized difficulty ranking over a given set of questions L_i for a target student s_i. EduRank estimates how similar other student are to s_i, and then combines the ranking of the similar students over L_i to create a ranking for s_i. There are two main procedures to the algorithm: computing the student similarity metric, and creating a difficulty ranking based on the ranking of similar users.

For comparing the target student s_i to potential neighbors, we use the s_{AP} metric to encourage greater similarity between students with high agreement in top positions in their respective rankings.

For aggregating the different students' rankings to create a difficulty ranking for the target student, we use the Copeland method (Section 2.2). We treat each question as a candidate and look at the aggregated voting of neighbors based on their similarity metric. In our aggregated voting calculation, candidate i beats candidate j if the similarity normalized number of wins of i over j computed over all neighbors is higher than the similarity normalized number of loses. The Copeland method then computes for each candidate question the overall number of aggregated victories and aggregated defeats and ranks the candidates accordingly. Before presenting the algorithm we first define $\gamma(q_k, q_l, \succ)$ over question pairs q_k, q_l

Algorithm 1 EduRank

```
INPUT:
Set of students $S$.
Set of questions $Q$.
For each student $s_j \in S$, a partial ranking $\succ_j$ over $T_j \subseteq Q$.
Target student $s_i \in S$.
Set of questions $L_i$ to rank for $s_i$.
OUTPUT: a partial order $\hat{\succ}_i$ over $L_i$.
for each $q \in L_i$ do
    $c(q) = \sum_{q_l \in L \setminus q} rv(q, q_l, S)$
end for
$\hat{\succ}_i \leftarrow \{\forall (q_k, q_l) \in \binom{L_i}{2}, q_k \hat{\succ}_i q_l$ iff $c(q_k) > c(q_l)\}$
return $\hat{\succ}_i$
```

given a difficulty ranking $\succ$ as follows:

$$\gamma(q_k, q_l, \succ) = \begin{cases} 1 & \text{if } q_k \succ q_l \\ -1 & \text{if } q_l \succ q_k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The relative voting $rv(q_k, q_l, S)$ of two questions q_k, q_l given the difficulty rankings of a group of (neighboring) students S is

$$rv(q_k, q_l, S) = sign(\sum_{j \in S \setminus i} s_{AP}(T_i, \succ_i, \succ_j) \cdot \gamma(q_k, q_l, \succ_j)) \tag{8}$$

The Copeland score $c(q, S, L_i)$ of a question q given the difficulty rankings of students S and test questions L_i is

$$c(q, S, L_i) = \sum_{q_l \in L_i \setminus q} rv(q, q_l, S) \tag{9}$$

The EduRank algorithm is shown in Algorithm 1. The input to the EduRank algorithm is a set of students $S = \{s_1, \ldots, s_n\}$, each with a known ranking over a set of questions T_j, such that $Q = T_1 \cup \ldots \cup T_n$. In addition the algorithm is given a target student $s_i \in S$, and a set of questions $L_i \subseteq Q$ that needs to be ranked for s_i. The output of the algorithm is a ranking of the questions in L_i.

The algorithm computes a ranking score $c(q)$ for each question $q \in L_i$, which is the Copeland score for that question, as defined above. The algorithm returns a partial order for student s_i over the test set L_i where questions are ranked by decreasing Copeland score $c(q)$.

5. EMPIRICAL EVALUATION

We now describe a set of experiments comparing EduRank to other algorithms on the difficulty ranking problem. We describe the datasets that were used and our method for defining a difficulty ranking, then we discuss the performance of the various algorithms.

5.1 Datasets

We conducted experiments on two real world educational datasets. The first dataset was published in the KDD cup 2010 by the Pittsburgh Science of Learning Center (PSLC) [2] [13]. We used the Algebra 1 dataset from the competition, containing about 800,000 answering attempts by 575 students, collected during 2005-2006. We used the following features for each question: question ID, the number of retries needed to solve the problem by the student, and the duration of time required by the student to submit the answer.[3] If the number of retries needed to solve the problem was 0, this means the students solved the problem on a first attempt (we refer to this event as "correct first attempt").

The second dataset, which we call K12, is an unpublished dataset obtained from an e-learning system installed in 120 schools and used by more than 10,000 students. The records in this dataset were anonymized and approved by the institutional review board of the Ben-Gurion university. This dataset contains about 900,000 answering attempts in various topics including mathematics, English as a second language, and social studies. We used the following features for each question: question ID, the answer provided by the student and the associated grade for each attempt to solve the question. Unfortunately, this dataset does not contain time stamps for each provided response, so we cannot compute the duration of time until a question was answered.

5.2 Feature Selection for Difficulty Ranking

EduRank assumes that each student has a personal difficulty ranking over questions, as described in Section 3. In this section we show how we inferred this ranking from the features in the dataset. An obvious candidate for the difficulty ranking are the grades that the student got on each question. There are several reasons however as to why grades are an insufficient measurement of difficulty. First, in all questions in the PSLC dataset, the "Correct First Attempt" score is either 0 or 1. There were a number of multiple choice questions (between 3 and 4 possible answers) in the datasets, but the dichotomy between low and high grades was also displayed here. To understand this dichotomy, note that students were allowed to repeat the question until they succeeded. It is not surprising that after several retries most students were able to identify the correct answer. A zero grade for a question occurs most often when it was not attempted by the student more than once.

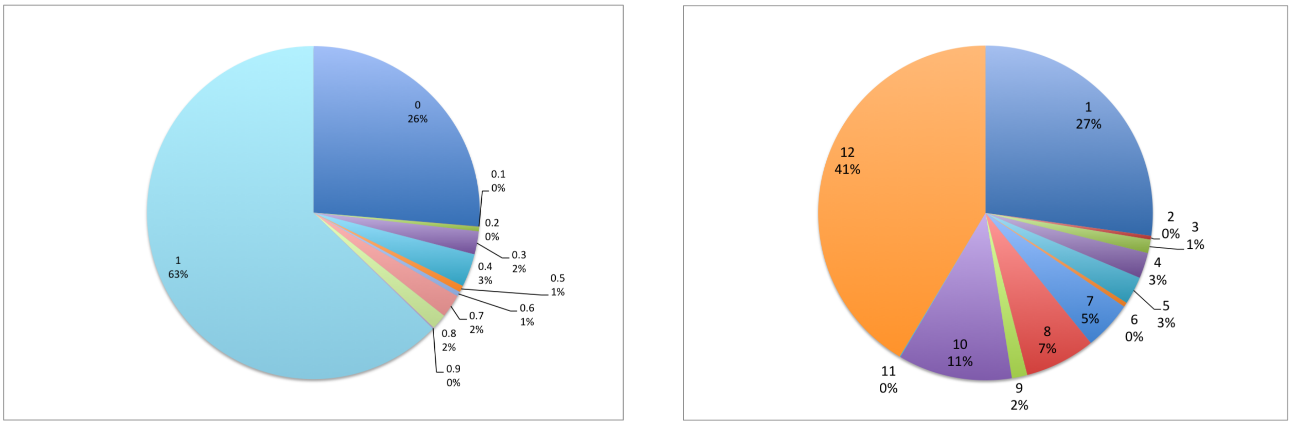
An alternative approach is to consider additional features in addition to grades (or correct first attempts), that are present in the datasets, and which correlate with the difficulty of the question for the individual student. Specifically, we assumed that questions that were answered correctly on a first attempt were easier for the student, while questions that required multiple attempts were harder. We also assumed that questions that required more solution time, as registered in the log, were more difficult to the students.

We realize that these two properties are not perfect indicators of question difficulty for the student. Indeed, it may occur in multiple choice questions that a student guessed the correct answer on the first attempt, even though the question was quite difficult. We also do not account for "gaming the system" strategies that have been modeled in past ITS work [4]. It may also be the case that the length of time reported by the system represents idle time for the student who was not even interacting with the e-learning software, or simply taking a break. However, as we show later in this section, these properties provide a reasonable estimation for the difficulty of the question.

We proceed to describe the following method for identifying the difficulty ranking. We begin by ranking questions by grades. In the PSLC dataset we use "correct first attempt" for this, and in the K12 dataset we find it more informative to use the grade that the

[2] **https://pslcdatashop.web.cmu.edu/KDDCup**

[3] Note there were other features in this data-set that were not used in the study.



(a) Grades

(b) Difficulty Ranking

Figure 1: Distribution over grades and difficulty ranking positions for K12 dataset

student got on her first attempt. After ranking by grade, we break ties by using the number of attempts that the student took before submitting the correct answer. When the student did not achieve a correct answer we use all the attempts that the student has made. Then, we break ties again on the PSLC dataset using the elapsed time.

To demonstrate that in general, these properties provide a reasonable estimation for the difficulty of the question, Figure 1 shows a distribution over students' grades (left) and positions in the inferred difficulty ranking which considered grades and retries (right). Note that the different values for grades represent answers to multiple select questions. For example, a grade of 0.9 will be achieved when 9/10 correct answers were selected by the student. As can be clearly seen from the figure, there are substantially more classes in the difficulty ranking when adding additional features.

5.3 Methods

We used two ranking scoring metrics — NDPM and AP (Section 2.3). Many papers in information retrieval also report NDCG, which is a ranking metric for datasets where each item has a score, and thus measures the difference in scores when ranking errors occur. In our case, where we do not have meaningful scores, only pure rankings, NDCG is less appropriate [12].

We compared the performance of a number of different methods to EduRank. First, we used the original EigenRank algorithm, which differs from EduRank in the similarity metric between users as well as the aggregation of the neighbor rankings.

As we explained in section 2.1, a popular alternative in the recommendation systems literature is to predict a score for an item, and then rank by sorting predicted scores. Thus we also used two popular collaborative filtering methods — a memory-based user-user KNN method using the Pearson correlation (denoted UBCF for User Based Collaborative Filtering), and a matrix factorization method using SVD (denoted SVD) to compute latent factors of items and users [6, 30]. In both cases we used the Mahout[4] implementation of the algorithms [23].

The collaborative filtering algorithms require an item score as an input. We used the following scores; We began with the grade (first attempt) that the user got on a question, normalized to the $[0-1]$ range. For each retry of the question we reduce this grade by 0.2 points. For the PSLC dataset, we reduce the (normalized) elapsed time solving the question from the score. This scoring method closely captures the difficulty ranking order we describe above. In the K12 dataset we also compared to a non-personalized difficulty ranking from 1-5 for each question, supplied by a domain expert (typically the researcher or teacher authoring the question). We denote this content expert ranking using CER.

Finally, it is common in the educational literature to identify the mastery level of the student on various topics, and then predict the performance of a question from the mastery level of the topic of the question [9]. To implement such an approach we computed the average score (using the scoring mechanism above) that the student got for all questions that belong to the same topic. We then rank the topics by decreasing average score, and rank the questions by the topic they belong to. We denote this method the Topic-Based Ranker (TBR). This measure was used only on the K12 dataset where we have available topic data.

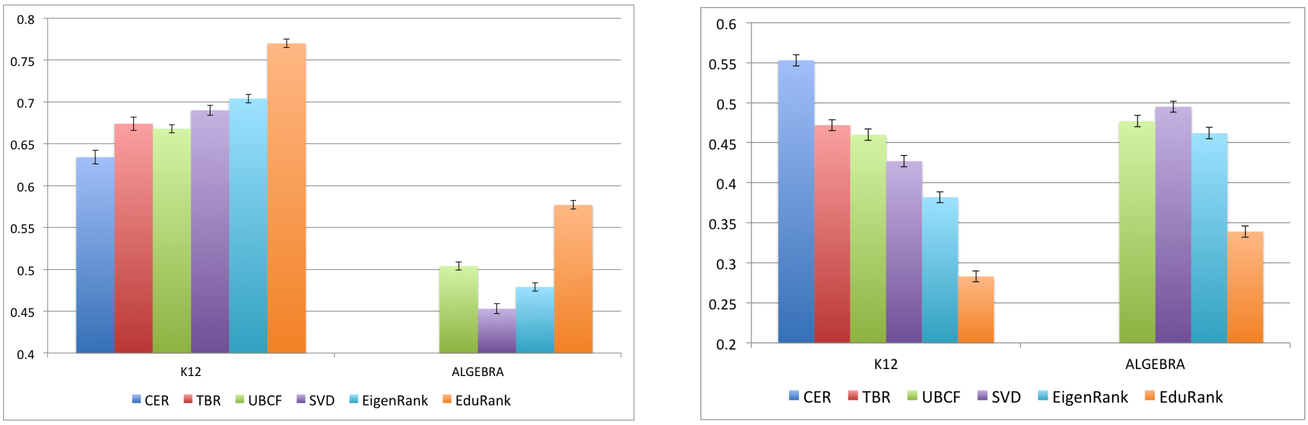
5.4 Results

Based on the problem defined in section 3, we ran the following experiment— for each student s_i we split her answered questions into two sets of equal size: a train set T_i, which is given as input to the various algorithms, and a test set L_i that the algorithms must rank. The split is performed according to the time stamp of the answers. Later answers are in the test set. We then compare the result of each algorithm to the difficulty ranking explained above using NDPM and AP.[5] Notice that for NDPM, the lower the score, the better the ranking, while for AP, better rankings result in higher scores. For all approaches, we ordered difficulty ranking in decreasing order of difficulty (harder questions were ranked higher in the list).

As can be seen in Figure 2 EduRank is better than all other approaches on both datasets using both metrics. The results are sta-

[4] `https://mahout.apache.org/`

[5] Note that the AP metric is also used to measure similarity between neighboring students in EduRank. We note that (1) it is standard practice in ML to use the same metric in the algorithm and the evaluation, and (2) the AP measure was computed over the training set in the algorithm, but over the test set in the evaluation.



(a) AP score (higher is better)

(b) NDPM score (lower is better)

Figure 2: Performance Comparison

tistically significant ($p < 0.05$, paired t-test between EduRank and the leading competing algorithm).

Looking at the other collaborative filtering methods we can see that EigenRank and UBCF present comparable performance. This is not very surprising, because these 2 methods do not take as input a ranking, but an item score, as we explain above. As the score is only a proxy to the actual ranking, it is no surprise that these algorithms do not do as well in predicting the true difficulty ranking.

Of the non-CF methods,TBR does relatively well. Our intuition is that identifying the student mastery level in topics is an important factor in establishing the difficulty of a question for that particular student. It is hence interesting to investigate in future research how EduRank can also benefit from the information encapsulated in topics. Nonetheless TBR can be too limiting in practice, because when a teacher wants to create a practice assignment in a particular topic, perhaps one that the student has not yet mastered, then TBR cannot be used to rank questions within that topic.

The method that performed the worst is the content expert ranking (CER). This is especially interesting as this is the only information that is currently available to teachers using the K12 e-learning system for deciding on the difficulty of questions. There can be two sources to this sub-optimal performance; First, it may be that it is too hard, even for experts, to estimate the difficulty of a question for students. Second, this may be an evidence that personalizing the order of questions for a particular student is truly important for this application.

5.5 Case Study

To further demonstrate the behaviour of the various algorithms, we took one of the students in the K12 dataset and present the results of the algorithms for that particular student. Table 1 presents a list of 34 test questions for this student and the rankings that were outputted by the different algorithms, in decreasing order of difficulty. The 15 most difficult questions appear in bold. Each question is denoted by (1) its knowledge component (KC) which was determined by a domain expert (this information was not in the database and the algorithms did not use it), and (2) the position of the question in the true difficulty ranking (the gold standard) of the student. This gold standard was used by the NDPM and AP metrics as a reference ranking to judge the performance of all algorithms. As shown in the table, question types involving “multiplication of big numbers” and “order of operations” appear prominently in the 15-most difficulty list, while questions in topics of geometry (“rectangles”, “polygons”) were easier for the student.

The other columns in the table show the suggested rankings by the various algorithms. For each algorithm, we present the ranking location of each question, and the true ranking of this question as obtained from the gold standard. As can be seen from the results, for this particular student, the UBCF algorithm performed poorly, placing many easy questions for the student at high positions in the ranking (e.g., “Multiply Eq 54” which appears at the top of the list but is ranked 12th in the gold standard, and “div mod” appears in 4th position in the list and ranked 11th in the gold standard.) The EigenRank and SVD algorithms demonstrated better results, but still failed to place the most difficult question for the student (e.g., order of operations) at the top of the ranked list. Only the EduRank algorithm was able to place the questions with “multiplication of big numbers” and “order of operation” type problems in the top 15 list, providing the best personalized difficulty ranking for this student.

Table 2 shows the execution time of each algorithm for building the models and computing the recommended rankings. The dataset used is the K12 dataset with 918,792 records. Our experiments were conducted on a Mac Book Air 1.7GHz Intel Core i7 with 8GB RAM.

Algorithm	Run Time (Sec)
CER	197.6
UBCF	445.2
TBR	625.2
EduRank	631.8
EigenRank	795.9
SVD	1490

Table 2: Execution Time

6. RELATED WORK

Our work relates to several areas of research in education and educational data mining. Several approaches within the educational

Gold Standard		EduRank Ranking		EigenRank Ranking		UBCF Ranking		SVD Ranking	
KC	True Rank	KC	True Rank	KC	True Rank	KC	True Rank	KC	True Rank
Order of Operations, choose options	**1**	**Order of Operations, choose options**	**1**	**Order of Operations, Brackets**	**7**	**Multuply, Equals 54**	**12**	**Multiply, Big Numbers**	**4**
Letters Order	**1**	**Natural Numbers, Verbal Claims**	**3**	**Natural numbers, In between**	**12**	**Multiply, Choose Between 2**	**12**	**Multiply, Bigger than**	**10**
Multiply, Equals 40	**2**	**Add, Sub, Equals 30**	**10**	**Div, No Mod, Mod 1**	**11**	**Multiply, Bigger than**	**10**	**Order of Operations, Brackets**	**5**
Natural Numbers, Verbal Claims	**3**	**Letters Order**	**1**	**Div, Div and Mod**	**11**	**Div, No Mod, Mod 1**	**11**	**Order of Operations, Equals 5**	**6**
Multiply, Big Numbers	**4**	**Add, Sub, Verbal Claims**	**7**	**Multiply, Big Numbers**	**7**	**Div, No Mod, Mod 2**	**12**	**Natural Numbers, Verbal Claims**	**3**
Order of Operations, Brackets	**5**	**Order of Operations, Equals 5**	**6**	**Div, Exists?**	**8**	**Multiply, Big Numbers**	**4**	**Add, Sub, Equals 30**	**10**
Zero, Equals Zero	**5**	**Order of Operations, Brackets**	**5**	**Multiply, Equals 40**	**2**	**Natural Numbers, Verbal Claims**	**3**	**Order of Operations, Brackets**	**7**
Order of Operations, Equals 5	**6**	**Zero, Equals Zero**	**5**	**Div, Mod 2**	**12**	**Order of Operations, choose options**	**1**	**Div, Mod 2**	**12**
Order of Operations, Brackets	**7**	**Multiply, Big Numbers**	**4**	**Multuply, Choose between 2**	**12**	**Order of Operations, Equals 5**	**6**	**Add, Sub, Verbal Claims**	**7**
Add, Sub, Verbal Claims	**7**	**Div, Mod 2**	**12**	**Order of Operations, Which is bigger**	**11**	**Multiply, Choose between 2**	**12**	**Order of Operations, choose options**	**1**
Multiply, Big Numbers	**7**	**Div, No Mod, Mod 2**	**12**	**Order of Operations, Brackets**	**5**	**Multuply, Choose between 2**	**12**	**Multuply, Equals 54**	**12**
Div, Exists?	**8**	**Order of Operations, Brackets**	**7**	**Div, Mod 1**	**11**	**Order of Operations, Brackets**	**7**	**Div, Exists?**	**12**
Substruction	**9**	**Order of Operations, Which is bigger**	**11**	**Order of Operations, only %, /**	**11**	**Order of Operations, Brackets**	**5**	**Div, No Mod, Mod 2**	**12**
Multiply, Bigger than	**10**	**Order of Operations, only %, /**	**11**	**Polygon, Parallel sides**	**10**	**Letters Order**	**1**	**Multiply, Big Numbers**	**7**
Add, Sub, Equals 30	**10**	**Multiply, Big Numbers**	**7**	**Letters Order**	**1**	**Rectangle, Identify**	**12**	**Natural numbers, In between**	**12**
Polygon, Parallel sides	10	Div, Exists?	12	Order of Operations, Equals 5	6	Multiply, Big Numbers	7	Zero, Equals Zero	5
Order of Operations, only +, -	11	Substruction	9	Substruction	9	Polygon, Identify	12	Order of Operations, Which is bigger	11
Order of Operations, only %, /	11	Polygon, Parallel sides	10	Add, Sub, Verbal Claims	7	Zero, Equals Zero	5	Div, Div and Mod	11
Order of Operations, Which is bigger	11	Order of Operations, only +, -	11	Multiply, Big Numbers	4	Order of Operations, only +, -	11	Letters Order	1
Div, Mod 1	11	Div, No Mod, Mod 1	11	Natural Numbers, Verbal Claims	3	Add, Sub, Equals 30	10	Angles, Find Bigger	12
Div, Div and Mod	11	Multiply, Bigger than	10	Add, Sub, Equals 30	10	Polygon, Parallel sides	10	Multiply, Choose between 2	12
Div, No Mod, Mod 1	11	Div, Exists?	8	Order of Operations, choose options	1	Add, Sub, Verbal Claims	7	Div, Mod 1	11
Natural numbers, In between	12	Div, Mod 1	11	Order of Operations, only +, -	11	Div, Mod 1	11	Multuply, Choose between 2	12
Multuply, Equals 54	12	Multiply, Equals 40	2	Zero, Equals Zero	5	Div, Mod 2	12	Div, No Mod, Mod 1	11
Multiply, Choose between 2	12	Div, Div and Mod	11	Div, No Mod, Mod 2	12	Div, Div and Mod	11	Polygon, Parallel sides	10
Multuply, Choose between 2	12	Multuply, Choose between 2	12	Div, Exists?	12	Order of Operations, only %, /	11	Div, Exists?	8
Div, Mod 2	12	Multiply, Choose Between 2	12	Multiply, Bigger than	10	Order of Operations, Which is bigger	11	Order of Operations, only %, /	11
Div, Exists?	12	Rectangle, Identify	12	Multiply, Choose Between 2	12	Div, Exists?	8	Substruction	9
Div, No Mod, Mod 2	12	Polygon, Identify	12	Rectangle, Identify	12	Div, Exists?	12	Order of Operations, only +, -	11
Angles, Find Bigger	12	Multuply, Equals 54	12	Multuply, Equals 54	12	Natural numbers, In between	12	Multiply, Equals 40	2
Angles, Find Bigger	12	Angles, Find Bigger	12	Angles, Find Bigger	12	Substruction	9	Angles, Find Bigger	12
Rectangle, Identify	12	Angles, Find Bigger	12	Angles, Find Bigger	12	Multiply, Equals 40	2	Multiply, Choose Between 2	12
Polygon, Identify	12	Natural numbers, In between	12	Multiply, Choose between 2	12	Angles, Find Bigger	12	Polygon, Identify	12
Multiply, Choose Between 2	12	Multiply, Choose between 2	12	Polygon, Identify	12	Angles, Find Bigger	12	Rectangle, Identify	12

Table 1: Rankings outputted by the different algorithms for a sample target student

data mining community have used computational methods for sequencing students' learning items. Pardos and Heffernan [18] infer order over questions by predicting students' skill levels over action pairs using Bayesian Knowledge Tracing. They show the efficacy of this approach on a test-set comprising random sequences of three questions as well as simulated data. This approach explicitly considers each possible order sequence and does not scale to handling large number of sequences, as in the student ranking problem we consider in this paper.

Champaign and Cohen [7] suggest a peer-based model for content sequencing in an intelligent tutor system by computing the similarity between different students and choosing questions that provide the best benefit for similar students. They measure similarity by comparing between students' average performance on past questions and evaluate their approach on simulated data. Our approach differs in several ways. First, we don't use an aggregate measure to compute similarity but compare between students' difficulty rankings over questions. In this way we use the entire ranked list for similarity computation, and do not lose information.[6] Second, we are using social choice to combine similar students' difficulty ranking over questions. Lastly, we evaluate our approach on two real-world data sets. Li, Cohen and Koedinger [14] compared a blocked order approach, in which all problems of one type are completed before the student is switched to the next problem type to an interleaved approach, where problems from two types are mixed and showed that the interleaved approach yields more effective learning. Our own approach generates an order of the different questions by reasoning about the student performance rather than determining order a-priori.

Lastly, multiple works have used Bayesian Knowledge Tracing as a way to infer students' skill acquisition (i.e., mastery level) over time given their performance levels on different question sequences [9]. These works reason about students' prior knowledge of skills and also account for slips and guessing on test problems. The models are trained on large data sets from multiple students using machine learning algorithms that account for latent variables [3, 10]. We solve a different problem, that of using other students' performance to personalize ranking over test-questions. In addition, these works measure students' performance dichotomously (i.e., success or failure) whereas we reason about additional features such as students' grade and number of attempts to solve the question. We intend to infer students' skill levels to improve the ranking prediction in future work.

Collaborative filtering (CF) was previously used in the educational domain for predicting students' performance. Toscher and Jahrer [25] use an ensemble of CF algorithms to predict performance for items in the KDD 2010 educational challenge. Berger et. al [5] use a model-based approach for predicting accuracy levels of students' performance and skill levels on real and simulated data sets. They also formalize a relationship between CF and Item Response Theory methods and demonstrate this relationship empirically. Lastly, Loll and Pinkwart [16] use CF as a diagnostic tool for knowledge test questions as well as more exploratory ill-defined tasks.

7. SUMMARY AND FUTURE WORK

This paper presented a novel approach to personalization of educational content. The suggested algorithm, called EduRank, combines a nearest-neighbor based collaborative filtering framework with a social choice method for preference ranking. The algorithm constructs a difficulty ranking over questions for a target student by aggregating the ranking of similar students. It extends existing approaches for ranking of user items in two ways. First, by inferring a difficulty ranking directly over the questions for a target student, rather than ordering them according to predicted performance, which is prone to error. Second, by penalizing disagreements between the difficulty rankings of similar students and the target student more highly for harder questions than for easy questions.

The algorithm was tested on two large real world data sets and its performance was compared to a variety of personalization meth-

[6] Consider student1 who has accrued grades 60 and 80 on questions (a) and (b) respectively; and student2 who has accrued grades 80 and 60 on questions (a) and (b) respectively. The average grade for both questions will be the same despite that they clearly differ in difficulty level for the students (when ordered solely based on grade).

ods as well as a non-personalized method that relied on a domain expert. The results showed that EduRank outperformed existing state-of-the-art algorithms using two metrics from the information retrieval literature.

In future work we plan to address the cold start problem by applying EduRank in a classroom setting in which we will personalize educational content to both exiting and new students. We also intend to evaluate Edurank's performance when training on small datasets and in MOOCs settings where the number of data points may dramatically change over time.

8. ACKNOWLEDGEMENTS

This work was supported in part by EU grant no. FP7-ICT-2011-9 #600854 and by ISF grant no. 1276/12.

9. REFERENCES

[1] Y. Akbulut and C. S. Cardak. Adaptive educational hypermedia accommodating learning styles: A content analysis of publications from 2000 to 2011. *Computers & Education*, 58(2):835–842, 2012.

[2] H. Ba-Omar, I. Petrounias, and F. Anwar. A framework for using web usage mining to personalise e-learning. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 937–938. IEEE, 2007.

[3] R. S. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, pages 406–415. Springer, 2008.

[4] R. S. Baker, J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger. Why students engage in gaming the system behavior in interactive learning environments. *Journal of Interactive Learning Research*, 19(2):185–224, 2008.

[5] Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. Seaton, and D. Pritchard. Model-based collaborative filtering analysis of student response data: Machine-learning item response theory. In *EDM*, pages 95–102, 2012.

[6] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[7] J. Champaign and R. Cohen. A model for content sequencing in intelligent tutoring systems based on the ecological approach and its validation through simulated students. In *FLAIRS Conference*, 2010.

[8] A. H. Copeland. A reasonable social welfare function. In *University of Michigan Seminar on Applications of Mathematics to the social sciences*, 1951.

[9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[10] M. H. Falakmasir, Z. A. Pardos, G. J. Gordon, and P. Brusilovsky. A spectral learning approach to knowledge tracing. In *EDM*, 2013.

[11] P. C. Fishburn. *The theory of social choice*, volume 264. Princeton University Press Princeton, 1973.

[12] E. Kanoulas and J. A. Aslam. Empirical justification of the gain and discount function for ndcg. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 611–620. ACM, 2009.

[13] K. R. Koedinger, R. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, pages 43–55, 2010.

[14] N. Li, W. W. Cohen, and K. R. Koedinger. Problem order implications for learning transfer. In *Intelligent Tutoring Systems*, pages 185–194. Springer, 2012.

[15] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 83–90. ACM, 2008.

[16] F. Loll and N. Pinkwart. Using collaborative filtering algorithms as elearning tools. In *42nd Hawaii International Conference on Systems Science*, 2009.

[17] H. Nurmi. Voting procedures: a summary analysis. *British Journal of Political Science*, 13(02):181–208, 1983.

[18] Z. A. Pardos and N. T. Heffernan. Determining the significance of item order in randomized problem sets. In *EDM*, 2009.

[19] D. M. Pennock, E. Horvitz, C. L. Giles, et al. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *AAAI/IAAI*, pages 729–734, 2000.

[20] D. Sampson and C. Karagiannidis. Personalised learning: Educational, technological and standardisation perspective. *Interactive Educational Multimedia*, 4:24–39, 2002.

[21] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

[22] F. Schalekamp and A. van Zuylen. Rank aggregation: Together we're strong. In *ALENEX*, pages 38–51, 2009.

[23] S. Schelter and S. Owen. Collaborative filtering with apache mahout. *Proc. of ACM RecSys Challenge*, 2012.

[24] G. Shani and A. Gunawardana. Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 257–297. Springer US, 2011.

[25] A. Toscher and M. Jahrer. Collaborative filtering applied to educational data mining. *KDD Cup*, 2010.

[26] Y. Yao. Measuring retrieval effectiveness based on user preference of documents. *JASIS*, 46(2):133–145, 1995.

[27] E. Yilmaz, J. A. Aslam, and S. Robertson. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 587–594. ACM, 2008.

[28] L. Zhang, X. Liu, and X. Liu. Personalized instructing recommendation system based on web mining. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 2517–2521. IEEE, 2008.

[29] B. Zhou and Y. Yao. Evaluating information retrieval system performance based on user preference. *Journal of Intelligent Information Systems*, 34(3):227–248, 2010.

[30] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.

Exploring Differences in Problem Solving with Data-Driven Approach Maps

Michael Eagle
North Carolina State University
890 Oval Dr, Campus Box 8206
Raleigh, NC 27695-8206
Maiku.Eagle@gmail.com

Tiffany Barnes
North Carolina State University
890 Oval Dr, Campus Box 8206
Raleigh, NC 27695-8206
Tiffany.Barnes@gmail.com

ABSTRACT

Understanding the differences in problem solving behavior between groups of students is quite challenging. We have mined the structure of interaction traces to discover different approaches to solving logic problems. In a prior study, significant differences in performance and tutor retention were found between two groups of students, one group with access to hints, and one without. The Approach Maps we have derived help us discover differences in how students in each group explore the possible solution space for each problem. We summarize our findings across several logic problems, and present in-depth Approach analyses for two logic problems that seem to influence future performance in the tutor for each group. Our results show that the students in the hint group approach the two problems in statistically and practically different ways, when compared to the control group. Our data-driven approach maps offer a novel way to compare behaviors between groups, while providing insight into the ways students solve problems.

Keywords

Approach Maps, Logic Tutor, Data Driven Models

1. INTRODUCTION

Intelligent tutors have been shown to be as effective as human tutors in supporting learning in many domains, in part because of their individualized, immediate feedback, enabled by expert systems that diagnose student's knowledge states [20]. For example, students provided with intelligent feedback in the LISP tutor spent 30% less time and performed 43% better on post-tests when compared to other methods of teaching [1]. Similarly, Eagle, and Barnes showed that students with access to hints in the Deep Thought logic tutor spent 38% less time per problem and completed 19% more problems than the control group [5]. In another study on the same data, Stamper, Eagle, and Barnes showed that students without hints were 3.6 times more likely to drop out and discontinue using the tutor [19].

Procedural problem solving is an important skill in STEM (science, technology, engineering, and math) fields. Open-ended procedural problem solving, where steps are well-defined, but can be combined in many ways, can encourage higher-level learning [2]. However, understanding learning in open-ended problems, particularly when students choose whether or not to perform them, can be challenging. The Deep Thought tutor allows students to use logic rules in different ways and in different orders to solve 13 logic proof problems for homework. In this paper, we analyze the 2009 Deep Thought data set analyzed by Stamper, Eagle, and Barnes to further understand the differences between the hint and control groups.

The rich interaction data saved by transactional tutor logs offers many avenues to explore and understand student problem solving data, particularly for problems with multiple solutions. By mapping Deep Thought transactional data into an interaction network, and applying graph mining to derive regions based on the structure of this network, we develop a new Approach Map that illustrates the approaches that groups of students take in solving logic problems. We built Approach Maps for all 13 problems in the tutor, and illustrate a detailed analysis of two of these maps to explore the differences in problem solving between the hint and control groups.

The Approach Maps for problems 1.4 and 1.5 show that the hint group explored productive regions of the interaction network, while students in the control group were more likely to explore unproductive regions that did not lead to solutions. Problem 1.4 had available hints for the hint group. Even though problem 1.5 has no hints for either group, the Approach Map shows that the two groups still explore the problem space differently, illustrating that prior access to hints had a lasting effect. The Approach Maps help us discover unproductive regions of the problem-solving space, that we believe contributed to lower retention rates for the control group. In these regions, proactive hints could be used to direct students toward more productive approaches.

In section 2, we discuss related work and the prior study with Deep Thought. In section 4, we describe our algorithm for extracting Approach Maps from data. Section 5 presents the results and illustrates two detailed Approach Maps on problems 1.4 and 1.5. Finally, we discuss the results, conclusions, and future directions for this work.

2. RELATED WORK

Although they can be very effective, the construction of intelligent tutors can be costly, requiring content experts and pedagogical experts to work with tutor developers to identify the skills students are applying and the associated feedback to deliver [12]. One way to reduce the costs of building tutoring systems is to build data-driven approaches to generate feedback during tutor problem-solving. Barnes and Stamper built the Hint Factory to use student problem-solving data for automatic hint generation in a propositional logic tutor [17]. Fossati at el. implemented Hint Factory in the iList tutor to teach students about linked lists[7]. Evaluation of the automatically generated hints from Hint Factory showed an increase in student performance and retention [19]; more details about this study are provided in section 3.1.

Although individual differences affect the ways that students solve problems [11], it is difficult to examine the overall approaches that groups of students demonstrate during problem-solving. While pre and posttests are useful for measuring the change in behavior before and after an experimental treatment, we are interested in studying not only whether a student can solve a problem, but how they are solving the problem. In this study, we use interaction networks of student behaviors to investigate how providing hints affects student problem-solving approaches.

Interaction Networks describe sequences of student-tutor interactions [6]. Johnson et al. showed that visualizations of interaction networks in the InVis tool could be used to better understand how students were using the Deep Thought logic tutor [10]. Interaction networks form the basis of the data-driven domain model for automatic step-based hint generation by the Hint Factory. Eagle et al. applyied Girvan-Newman clustering to interaction networks to determine whether the resulting clusters might be useful for more high-level hint generation [6]. Stamper et al. demonstrated the differences in problem solving between the hint and control groups by coloring the edges between Girvan-Newman clusters of interaction networks based on the frequencies between two groups, revealing a qualitative difference in attempt paths [19]. In this paper we expand on these works to develop Approach Maps that concisely illustrate the approaches that students take while solving problems.

The Girvan-Newman algorithm (GN) was developed to cluster social network graphs using edge betweenness to find communities of people [8]. The technique also works in other domains. Wilkinson et al. applied GN in gene networks to find related genes [21]. Gleiser et al. used GN to discover essential ingredients of social interactions between jazz musicians [9]. We are the first to apply GN to interaction networks consisting of problem-solving steps.

In this paper, we mine the interactions from student problem solving data to summarize a large number of student-tutor transaction data into an Approach Map, demonstrating the diverse ways students solve a particular problem. We use Approach Maps to better understand the differences in behavior between two groups, students who were given access to hints, and those who were not, while completing homework in the *Deep Thought* logic proof tutor.

3. THE DEEP THOUGHT LOGIC TUTOR

In Deep Thought propositional logic tutor problems, students apply logic rules to prove a given conclusion using a given set of premises. Deep Thought allows students to work both forward and backwards to solve logic problems [3]. Working backwards allows a student to propose ways the conclusion could be reached. For example, given the conclusion B, the student could propose that B was derived using Modus Ponens (MP) on two new, unjustified (i.e. not yet proven) propositions: $A \to B, A$. This is like a conditional proof in that, if the student can justify $A \to B$ and A, then the proof is solved. At any time, the student can work backwards from any unjustified components (marked with a ?), or forwards from any derived statements or the premises. Figure 1 contains an example of working forwards and backwards with in Deep Thought.

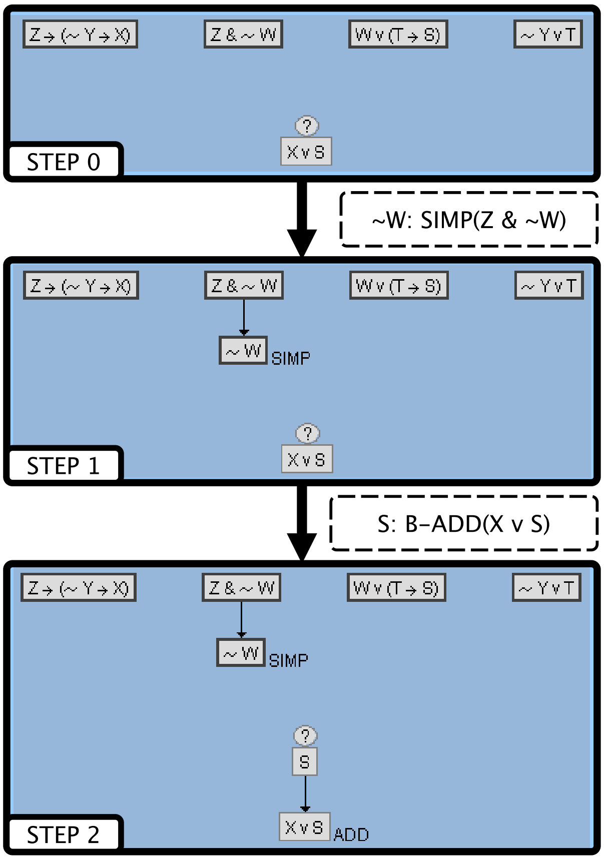


Figure 1: This example shows two steps within the Deep Thought tutor. First, the student has selected $Z \wedge \neg W$ and performed Simplification (SIMP) to derive $\neg W$. Second, the student selects $X \vee S$ and performs backward Addition to derive S.

3.1 Dataset and Prior Results

In 2012, Stamper, Eagle, and Barnes studied the effect of data-driven hints using the Spring and Fall 2009 Deep Thought propositional logic tutor dataset [19]. Data was collected from six 2009 deductive logic courses, taught by three professors. Each instructor taught one class using Deep Thought with automatically-generated hints on half of the problems (hint group, n=105) and one without access to hints on any problems (control, n=98). Students from the 6 sections were assigned 13 logic proofs in Deep Thought as a series of three graded homework assignments, with problems L1: 1.1-1.6, L2: 2.1-2.5, and L3: 3.1-3.2.

Table 1 shows retention information for each group after level L1; a χ^2 test of the relationship between group and dropout produced $\chi^2(1) = 11.05$, which was statistically significant at $p = 0.001$. The hint group completed more problems, with the effect sizes for these differences shown in Table 2. Stamper et al. found that the odds of a student in the control group dropping out of the tutor were 3.6 times more likely when compared to the group provided with automatically generated hints [19].

Table 1: Number of students that continued or dropped out of the tutor after L1

Group	Total	# Continued	# Dropped	% Dropped
Hint	105	95	10	9%
Control	98	71	27	28%
Total	203	166	37	18%

Table 2: The effect sizes of the differences between hint group and control group for completion and attempt rates by level.

	L1	L2	L3
Completed	$d = 0.51$*	$d = 0.64$*	$d = 0.39$*
Attempted	$d = 0.27$	$d = 0.44$*	$d = 0.33$*

Figure 2 charts the attempt and completion rates for hint group and control group for each problem in Deep Thought. Both groups had similar problem attempt rates, shown using solid lines, for L1 (1.1-1.5), but the hint group had significantly higher attempt rates in L2 and L3. The completion rates for each group are shown with dashed lines in Figure 2. Note that, after problem 1.4, the differences in attempt rates and completion rates seem to diverge between the groups.

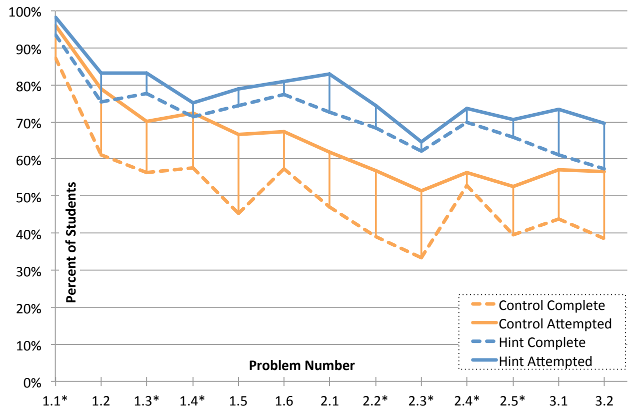


Figure 2: Attempt and complete rates per level, *indicates a problem where the hint group was given access to automatically generated hints.

We have investigated these results further. In another study, we modeled the time spent in the tutor using survival analysis [5]. In this study, we model the approaches students took to solve each problem.

4. METHODS

In Section 4.1, we describe how we use Deep Thought tutor logs to create an interaction network of all the student-tutor interactions within a single problem. We then show how we refine this network into *regions* of densely connected subgraphs (Section 4.2) using the Girvan-Newman (GN) algorithm. Finally, in Section 4.2.1 we define how we construct Approach Maps from the GN regions. For both steps in the process, we use the statistical environment *R* [14], and the complex network research library *iGraph* [4].

4.1 Constructing an Interaction Network

We construct an interaction network using all observed solution attempts to a single problem. Each solution attempt is a sequence of {state, action, resulting-state} interactions from the problem start to the last step a student performs. The *state* represents enough information to regenerate the tutor's interface at each step. An *action* is defined as a step taken, and consists of the name of the rule applied, the statements it was applied to, and the resulting derived statement. For example, Figure 1 displays two Deep Thought interactions. The first interaction works forward from STEP0 to STEP1 with action $SIMP$ (simplification) applied to $(Z \wedge \neg W)$ to derive $\neg W$. The second interaction works backward from STEP1 to STEP2 with action $B - ADD$ (backwards addition) applied to $(X \vee S)$ to derive the new, unjustified statement S.

We use a state matching function to combine identical states, that consist of all the same logic statements, but may have been derived in a different order. This way, the state for a step STEP0, STEP1, or STEP2 in Figure 1 is the set of justified and unjustified statements in each screenshot, regardless of the order that each statement was derived. We use an action matching function to combine actions, and preserve the frequency of each observed application.

If we treat the interactions used to create the networks as *samples* of observed behavior from a population, we could expect that the interaction networks constructed from different populations may have observable differences. However, rather than building two separate interaction networks and attempting to compare them, we construct a single network but keep track of the frequencies of visits by the hint and control groups for each state (vertex) and action (edge).

4.2 Extracting Regions

We partition the interaction network into densely connected subgraphs we call *regions* using Girvan and Newman's edge-betweenness clustering algorithm [8] and modularity score, a measure of the internal verses external connectedness of the regions [13]. We use following algorithm to apply region labels to nodes in a Deep Thought interaction network. First, we remove the problem start state and goal states from the Interaction Network IN to create G_1. Then, we iteratively remove all edges in G_1, in order of edge betweenness. Edge betweenness (EB) for a particular edge e is calculated by computing all shortest paths between all pairs of nodes, and counting the number of shortest paths that contain the edge e. At each GN iteration i and graph G_i, we find the edge with the highest EB, and call this bridge b_i. We remove the bridge b_i from the graph G_i, and compute the modularity score for the resulting graph G_{i+1}. The process is repeated until all edges have been removed. Then, we assign identifiers to all nodes in the disjoint regions in the intermediate graph G_n with the best modularity score. At the end of this process, we use G_n to construct the Approach Map with nodes for the original start and goal states, and a new node for each region in G_n. The Approach Map edges are the edges that connect the start state and goals to the regions, and the bridges between regions that were removed from the interaction network to create G_n.

Regions represent sets of steps that are highly connected to one another. When a solution attempt is within a region, new actions will stay within the region, or take a bridge edge into another region or goal. If an attempt is in a region with no goal bridges, the student must take a bridge to another region to reach a goal. Therefore, paths on the Approach Map can be interpreted as a high-level approaches to solving the problem. We hypothesize that we can use the Approach Map to discover different problem-solving approaches. In the next section, we investigate Approach Maps for two problems in Deep Thought, after which the hint and control groups diverged in performance.

4.2.1 *Approach Map*

Here we provide a more detailed description of the algorithm we use to generate an Approach Map from the interaction network for a problem after its nodes have been labeled with region identifiers. A region A (or action a) *dominates* a region B if every path from the start of the problem to B, must go through A (or a).

1. Combine all nodes with the same region identifier into a single region node labeled with the identifier, and remove all the edges with the same region identifier.
2. Combine all goal states that are dominated by a single region into a single goal node.
3. Calculate chi-squared to find in-edge frequencies that are different than expected between the groups (described in more detail below).
4. Combine parallel bridge edges between two regions into complex edges that represent the combination of the actions.
5. Label each region with the post conditions (derived statements) that result from the most frequent in-edge actions.
6. Provide new region identifiers that indicate the significant regions by the group with larger than expected frequency, with a number indicating the order in which the region was formed. For example, the regions the hint group visits more than expected are H1, H2, ..., the regions the control group visits more than expected are C1, C2, ..., and those that are visited as expected by both groups are labeled N1, N2, etc.

We use a two-tailed chi-squared test to look for differences between the hint and control groups in how they visit regions in the Approach Map. The null hypothesis is that there is no difference in the frequency of entering a particular region between attempts in the hint group and the control group. The alternative hypothesis is that the groups enter regions with different than expected frequency. We use Bonferroni correction [15] to compensate for the number of tests that we run. When the p is less than the Bonferroni-corrected alpha, we label the regions H1, H2, etc., blue for significantly higher than expected participation by the hint group. Regions C1, C2, etc., are bordered in orange and represent regions where the control group was represented more frequently than expected. Regions N1, N2, etc., satisfy the null hypothesis in that both groups visit these regions as expected.

The Approach Map for problem 1.4 is shown in Figure 3. Each region node contains statements derived on the most frequent in-edge. The bridge edges are those actions that most frequently lead into and out of each region. The edges are labeled with the action(s) taken and the number of attempts using these actions. A bridge and its resulting region can be read as, this many students performed the following action(s) to derive the following proposition(s). For clarity we do not draw edges with frequency less than ten, and we delete actions and regions that become disconnected due to these edge removals. The edges on the map are colored on a spectrum based on the ratio between the groups from blue (hint group) to orange (control group.) Paths in the Approach Map can be interpreted as empirically-observed problem solving approaches.

Each approach map is accompanied by a region table which provides more detail about the frequencies of observed solution attempts from each group. The columns Hint and Control are the total frequencies of in-edges by each group, or in other words, the number of solution attempts from each group that visit at least one node in the region. Time refers to the mean time a solution attempt stays in the region before exiting. Goals refers to the sum of the frequencies of out-edges that lead to goal states. The p values are the results of the chi-squared tests to compare group representation to expected values.

5. RESULTS & DISCUSSION

We perform our experiments on the Spring and Fall 2009 Deep Thought propositional logic tutor dataset as analyzed by Stamper, Eagle, and Barnes in 2012[19]. The data set is made up of 4301 student-attempts which contain 85454 student-tutor interactions across 13 problems. The prior study compared the performance between the hint (n=105) and control (n=98) groups, showing that students with available hints on the first 5 problems in L1 were 3.6 times more likely to complete the tutor. In addition, the hint group spent about 12 minutes per problem in the tutor, while the control group took 21 minutes per problem. Although the average total time in tutor between groups was not significantly different, more in-depth analysis of time revealed that this was because many students in the control group dropped out of the tutor, and were less likely to complete problems attempted in levels L2 and L3 [5]. In this section we present the results of applying Approach Maps to 11 problems in this data set, and illustrate the Approach Maps to two problems 1.4 and 1.5, just before the retention gap begins between the hint and control groups.

Table 3 summarizes our results from constructing Approach Maps for 11 of the 13 Deep Thought problems (records for problems 1.6 and 2.1 have not been normalized into our standard format). It is difficult to summarize the information from each map in to a single row in a table, however we have selected a few measures that provide an overview. In Table 3, the Hint and Control columns count the number of problem attempts for each group. Regions refers to the total number of regions in each Approach Map. Sig-H and Sig-C denote the number of regions visited significantly more than expected by the hint and control groups respectively. Sig-G denotes the number of significant regions that were also goal regions. This table shows that most problems in Deep Thought have 10-17 regions. In problems 1.4 and 1.5, more than half of the regions were visited more than expected by the hint or control groups.

Table 3: Summary of Approach Maps for 11 Deep Thought tutor problems. An asterisk (*) indicates problems where the hint group had access to hints.

Prob	Hint	Control	Regions	Sig-H	Sig-C	Sig-G
1.1*	348	447	16	1	7	2
1.2	196	187	16	1	2	1
1.3*	171	152	15	2	3	0
1.4*	138	219	16	5	4	2
1.5	155	218	18	4	6	2
2.2*	150	150	15	4	4	1
2.3*	129	108	14	4	3	1
2.4*	99	80	10	3	1	1
2.5*	112	79	10	3	0	1
3.1	173	114	17	0	1	0
3.2	147	100	12	1	2	0

We present detailed Approach Maps for problems 1.4 and 1.5 for three reasons. First, they occur before a large increase in control group dropout, as shown in Figure 2 in Section 3.1. After these problems, the odds of the control group dropping (no longer logging into the tutor) was 3.6 times that of the hint group [19]. Second, these problems stand out in Table 3, with high goal regions and more than half the extracted regions being significantly different between the groups. Third, in problem 1.4 the hint group had access to hints, however in problem 1.5 neither group received hints. This allows us to look for differences in behavior between the groups when working in the tutor on equal terms. For each of these problems we generated the Approach Map and corresponding reference table and visualization as described in Section 4.2.1.

5.1 Problem 1.4

Problem 1.4: Prove $X \vee S$
Given: $Z \rightarrow (\neg Y \rightarrow X), Z \wedge \neg W, W \vee (T \rightarrow S), \neg Y \vee T$

Problem 1.4 was designed to teach the Constructive Dilemma (CD) rule $[((P \rightarrow Q) \wedge (R \rightarrow S)) \wedge (P \vee R)] \rightarrow (Q \vee S)$. For this problem, students in the hint group had access to hints. Table 4 describes the regions of the Approach Map. Figure 3 shows the Approach Map for problem 1.4. To show differences in more detail, we have provided the most common attempts for each group in figure 4. In particular, this figure shows that the control group has derived an unjustified statement T that cannot be proven.

Hints were available for the hint group on problem 1.4; Table 5 shows the number of hint requests at depths D1 to D4, where students could request up to four consecutive hints while in a single state. In Table 5, R is the region, D1–4 is the depth of the hint, Target Proposition refers to the proposition the student is directed to derive, and Rule is the rule that the student is directed to use. Depth D1 hints direct students to the Hint column, while depth D2 hints direct the students to the Rule column. Depth D3 tells the student the preconditions needed to derive the target proposition. The depth D4 hint is a bottom out hint that directly tells the student what interface elements to click to derive the target step.

Table 4: Detailed information on the regions in the 1.4 Approach Map shown in Figure 3.

Region	Hint	Control	Time	Goals	p
H1	109	65	1.59	2	<0.001
H2	89	43	1.71	81	<0.001
H3	19	3	1.34	22	<0.001
C1	9	106	0.41	0	<0.001
C2	6	68	0.41	0	<0.001
C3	5	62	1.95	0	<0.001
C4	24	134	0.32	0	<0.001
N1	22	51	0.9	0	0.089
N2	9	15	1.41	20	0.811
N3	10	23	1.47	2	0.261
N4	14	38	0.13	0	0.056

Table 5: Number and depth of hints used by the hint group in each region; PS=Problem Start

R	D1	D2	D3	D4	Target Proposition	Rule
PS	50	13	13	4	$\neg W$	SIMP
H1	36	17	11	5	Z	SIMP
H1	32	16	11	5	$T \rightarrow S$	DS
H1	29	17	10	3	$\neg Y \rightarrow X$	MP
H2	36	19	18	2	$(\neg Y \rightarrow X) \wedge (T \rightarrow S)$	CONJ
H2	21	17	12	3	$X \vee S$	CD

There are three obvious paths in the Approach Map in Figure 3, one for the hint group, one for the control, and one with no differences between the groups. Figure 4 shows the most common solution paths for the hint and control group, with the same edges as the Approach Map. The Hint group tends to work forward using simplification (SIMP) (H1 to H2), while the control group was more likely to work backwards with addition (B-ADD) (C4 to C1). This backward addition path is a buggy strategy, that does not lead to any goals. We note that there are no backwards hints given in Deep Thought, so students on this path do not get hints regardless of group. Data on hint usage, shown in Table 5 and the statements derived in the H1-H3 regions suggest that students in the Hint group are being “routed” toward a successful strategy.

The Approach Map in Figure 3 shows that the control group is more likely to visit regions that do not contain successful goals. It seems that the effect of hints is to keep students

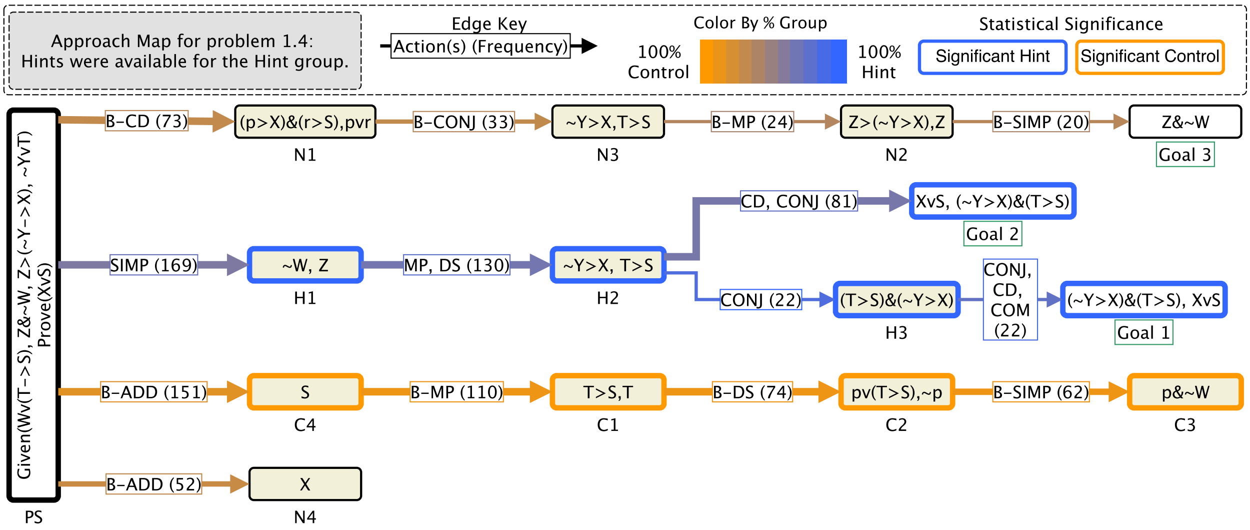


Figure 3: The Approach Map for problem 1.4. Edges and vertexes can be read as the number of students who performed action(s) to derive proposition(s). Three main approaches are revealed, with the hint group strongly preferring to work the problem forwards. The control group often attempts to solve the problem by wards with addition, there are no goals along this path. More detail is given in Table 4.

along a particular solution path, or prevent them from following the unproductive one taken by the control group. As a prior study of this data suggests [18], these students without hints are likely to abandon the tutor altogether. We hypothesize that hints help students achieve small successes and remain in the tutoring environment.

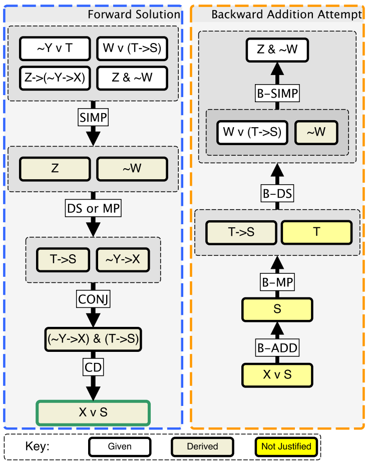


Figure 4: The most common attempt paths for each of the main approaches in the approach map for problem 1.4 (figure 3.) The highlighted nodes represent unjustified propositions.

5.2 Problem 1.5

Problem 1.5: Prove $A \vee \neg C$, given: $B \to (A \to E), B \vee (A \to \neg C), D \wedge \neg(A \to \neg C), E \to \neg C$.

Problem 1.5 was designed to teach the Hypothetical Syllogism (HS) axiom $[(P \to Q) \wedge (Q \to R)] \to (P \vee R)$. Problem 1.5 is interesting, as this problem had no hints, but still has large differences between the groups. The Approach Map is shown in Figure 5, and additional information on the regions is available in Table 6.

Table 6: Detailed information on the regions in the 1.5 Approach Map shown in Figure 5.

Region	Hint	Control	Time	Goals	p
H1	53	39	0.42	82	0.002
H2	89	58	1.16	26	<0.001
C1	17	55	0.69	0	0.002
C2	36	106	0.19	0	<0.001
C3	24	65	2.41	0	0.005
C4	16	51	0.72	0	0.003
C5	30	81	1.2	0	0.002
C6	3	19	0.22	0	0.007
N1	7	14	2.27	0	0.434
N2	7	8	1.16	11	0.700
N3	2	12	3.38	0	0.037
N4	8	15	0.26	0	0.498

The Hint group approaches problem 1.5 by working forward using simplification (SIMP) on $D \wedge B$ to derive the separate statements D and B; this could be a result of the forward directed hints they received in the earlier problems. The hints may have helped students develop a preference to working forwards, as doing so allowed them to request help if they became stuck. This preference carried over to the problems where hints were not available.

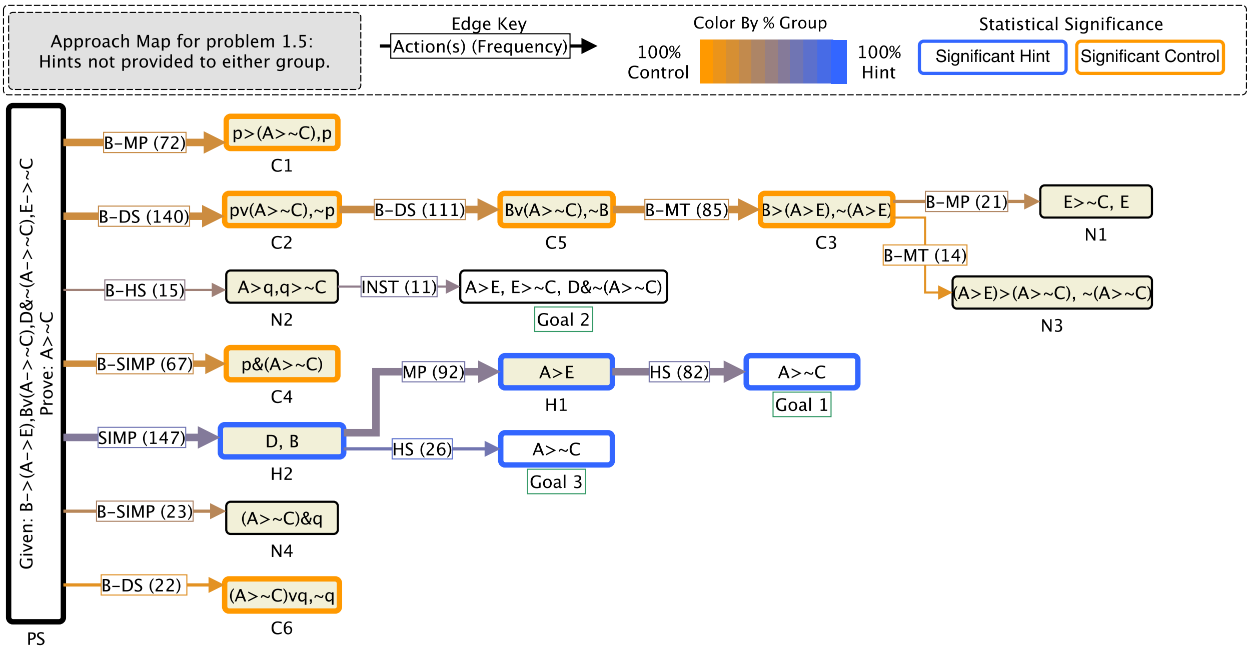


Figure 5: Even in the absence of the automatically generated hints, the hint group still prefers a forward solution. The control group explores regions that do not lead to goals. Details are given in Table 6.

When working problem 1.5, the control group systematically derives statements that do not lead to goals. The most common attempt is to work backwards with disjunctive syllogism (B-DS) (region C2) to derive $B \vee (A \rightarrow \neg C), \neg B$ from the conclusion $A \rightarrow \neg C$. This is likely because connecting with the premise $B \vee (A \rightarrow \neg C)$ seems like a promising direction. However, it is not possible to justify the proposed proposition $\neg B$ in this problem. This discovery is important as interventions can be added to warn away from regions that do not lead to goals. For example, we could offer a message that warns them that most students who attempt the same type of proof are not successful. Fossati et al. showed that human tutors helping students with the iList tutor, suggest that students delete unproductive steps [7].

5.3 Working Backwards and Trailblazing

Although working backwards seems be unproductive for the control group, we note that there are productive approaches that work backwards, for example N1-N4 regions in problem 1.4 explored evenly by both groups. There are some advantages to working backwards in *Deep Thought*. When a student works backwards, *Deep Thought* asks whether they would like to target the premises (extraction) or construct their own hypothesized statement from the conclusion. Then, the student clicks on one of just a few rules that can be used backwards, limiting the search space for the next step. Next, students are prompted to fill in the blanks in statements derivable from the chosen rule.

Region N1 in Figure 3, shows variables p and r that students can set to any proposition. Should the newly derived statements seem to match the patterns of existing premises, students keep them; otherwise they delete and try again. *Deep Thought* will sometimes warn students when they try to work backwards with something that is not justifiable. However, this may lead students to think that the tutor can always determine when working backwards is a viable strategy. In this case, students might mistakenly suppose that if there is no error message, they are closer to the solution. This is not the case, as *Deep Thought* has no built-in measures to determine closeness to completion. Rather, a few buggy rule applications are included in Deep Thought's automated error detection.

5.3.1 *Trailblazing Effect*

Barnes and Stamper proposed that hints might limit the breadth of student approaches to problems, causing a hint 'trailblazing' effect that might bias students toward expert solutions when originally building the Hint Factory[16]. In this analysis, we see some evidence of this effect. The difference in solution breadth between the two groups seems to be significant on several problems. The hints provided were limited to working forward, and the hint group demonstrated a strong preference for working forward. It remains to be seen whether providing hints for working backward will allow for more breadth of the search space. In any case, our results suggest that hints can cause a trailblazing effect, even when no hints are provided. Therefore, hints should be carefully constructed to include the diversity that a tutor designer wishes to promote in the tutor.

5.4 Conclusions and Future Work

In this paper, we have presented Approach Maps, a novel representation of student-tutor interaction data that allows for the comparison of problem-solving approaches on open-ended logic problems. The Approach Map visualization results in a significant reduction in the space needed to describe a large amount of student-tutor data. It does this by reducing the student attempts into regions that we can consider as higher-level approaches to problem-solving. Deep Thought problems each had an average of 330 solution attempts, which were made up of about 6.5 thousand interactions. Using our Approach Maps, we partition problems into about 15 regions each (including 2–3 goal regions, as shown in Table 3).

We have shown that we can use Approach Maps annotated with frequencies of visits by two groups to identify regions where a particular study group was over-represented. This allowed us to examine the approaches each group took to solving each proof. As we predicted, the automatically generated hints seemed to direct the students in the hint group down a common path, and we were able to detect this with the Approach Maps. Interestingly, even in problem 1.5, where neither group had hints, the hint group still showed a preference for working forwards, providing some evidence for a persistent effect of the hints. Analyzing Approach Maps also facilitated another important discovery that control group tended enter and remain in unproductive (or buggy) regions. These observed differences help explain how the automatically-generated hints produced the difference in tutor performance and retention in the 2009 Deep Thought study. Our investigations suggest that the patterns of behavior exhibited by students do result in meaningful regions of the solution attempt search space. We believe that, since the algorithms we applied to derive Approach Maps work on general graphs, we may be able to apply Approach Maps to understand problem-solving in domains where students solve open-ended problems in a procedural way.

In our future work, we plan to use Approach Maps to provide students with hints towards target sub-goals rather than simple step-based hints. We could also combine this with expert-created subgoals. We hypothesize that these more abstract hints will help encourage student planning. We also plan to use Approach Maps to provide proactive feedback to students when they enter unproductive regions. We will also apply Approach Maps to other open-ended problems to investigate their generalizability to other STEM fields.

6. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0845997. The authors wish to thank John Stamper for providing the 2009 Deep Thought dataset for this analysis.

7. REFERENCES

[1] J. R. Anderson and B. J. Reiser. The lisp tutor. *Byte*, 10(4):159–175, 1985.

[2] B. S. Bloom. *Taxonomy of Educational Objectives: The Classification of Educational Goals.* Taxonomy of educational objectives: the classification of educational goals. Longman Group, New York, 1956.

[3] M. J. Croy. Problem solving, working backwards, and graphic proof representation. *Teaching Philosophy*, 23:169–188, 2000.

[4] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[5] M. Eagle and T. Barnes. Survival analysis on duration data in intelligent tutors (under review). In *Intelligent Tutoring Systems*, Honolulu, Hawaii, 2014.

[6] M. Eagle, M. Johnson, and T. Barnes. Interaction Networks: Generating High Level Hints Based on Network Community Clustering. *educationaldatamining.org*, pages 1–4.

[7] D. Fossati, B. Di Eugenio, S. Ohlsson, C. Brown, L. Chen, and D. Cosejo. I learn from you, you learn from me: How to make ilist learn from students, 2009.

[8] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. National Academy of Sciences*, 99(12):7821–7826, June 2002.

[9] P. Gleiser and L. Danon. Community structure in jazz. *arXiv preprint cond-mat/0307434*, 2003.

[10] M. Johnson, M. Eagle, and T. Barnes. Invis: An interactive visualization tool for exploring interaction networks. In *Educational Data Mining (EDM2013)*, pages 82–89, 2013.

[11] D. Jonassen. Toward a design theory of problem solving. *Educational Technology Research and Development*, 48(4):63–85, 2000.

[12] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.

[13] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[14] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[15] J. P. Shaffer. Multiple hypothesis testing. *Annual review of psychology*, 46(1):561–584, 1995.

[16] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. *Intelligent Tutoring Systems YRT*, pages 71–78, 2008.

[17] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. *Educational Data Mining (EDM 2008)*, pages 197–201, 2008.

[18] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. In *Artificial Intelligence in Education*, AIED'11, pages 345–352, Berlin, Heidelberg, 2011. Springer-Verlag.

[19] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education (IJAIED)*, 22(1):3–18, 2012.

[20] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

[21] D. M. Wilkinson and B. A. Huberman. A method for finding communities of related genes. *Proc. National Academy of Sciences*, 101(Suppl 1):5241–5248, 2004.

General Features in Knowledge Tracing: Applications to Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge

* José González-Brenes[†], * Yun Huang[♯]
[†]Digital Data, Analytics & Adaptive Learning
Pearson Research & Innovation Network
Philadelphia, PA, USA
jose.gonzalez-brenes@pearson.com

Peter Brusilovsky[♯]
[♯]Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA, USA
{yuh43, peterb}@pitt.edu

ABSTRACT

Knowledge Tracing is the de-facto standard for inferring student knowledge from performance data. Unfortunately, it does not allow modeling the feature-rich data that is now possible to collect in modern digital learning environments. Because of this, many ad hoc Knowledge Tracing variants have been proposed to model a specific feature of interest. For example, variants have studied the effect of students' individual characteristics, the effect of help in a tutor, and subskills. These ad hoc models are successful for their own specific purpose, but are specified to only model a single specific feature.

We present FAST (Feature Aware Student knowledge Tracing), an efficient, novel method that allows integrating general features into Knowledge Tracing. We demonstrate FAST's flexibility with three examples of feature sets that are relevant to a wide audience. We use features in FAST to model (i) multiple subskill tracing, (ii) a temporal Item Response Model implementation, and (iii) expert knowledge. We present empirical results using data collected from an Intelligent Tutoring System. We report that using features can improve up to 25% in classification performance of the task of predicting student performance. Moreover, for fitting and inferencing, FAST can be 300 times faster than models created in BNT-SM, a toolkit that facilitates the creation of ad hoc Knowledge Tracing variants.

Keywords

knowledge tracing, feature engineering, IRT, subskills

1. INTRODUCTION

Various kinds of e-learning systems, such as Massively Open Online Courses and intelligent tutoring systems, are now producing large amounts of feature-rich data from students solving items at different levels of proficiency over time. To analyze such data, researchers often use Knowledge Tracing [7], a 20-year old method that has become the de-facto standard for inferring student knowledge. Unfortunately, Knowledge Tracing uses only longitudinal performance data and does not permit feature engineering to take advantage of the data that is collected in modern e-learning systems, such as student or item differences. Prior work has focused on adhoc modifications to Knowledge Tracing to enable modeling a specific feature of interest. This has led to a plethora of different Knowledge Tracing reformulations for very specific purposes. For example, variants have studied measuring the effect of students' individual characteristics [15, 18, 21, 29], assessing the effect of help in a tutor system [3, 25], controlling for item difficulty [10, 20, 26], and measuring the effect of subskills [28]. Although these ad hoc models are successful for their own specific purpose, they are single-purpose and require considerable effort to build.

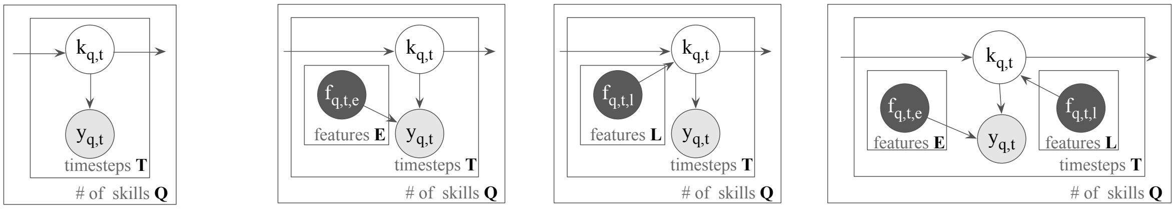
We propose *Feature-Aware Student knowledge Tracing* (FAST), a novel method that allows efficient general features into Knowledge Tracing. We propose FAST as a general model that can use features collected from digital learning environments. The rest of this paper is organized as follows: Section 2 describes the scope of the features FAST is able to model; Section 3 describes the FAST algorithm; Section 4 reports examples of using features with FAST; Section 5 compares FAST's execution time with models created by BNT-SM; Section 6 relates to prior work; Section 7 concludes.

2. KNOWLEDGE TRACING FAMILY

In this section we define a group of models that we call the Knowledge Tracing Family. We argue that a significant amount of prior work has reinvented models in the Knowledge Tracing Family for very diverse uses, yet their structures when represented as a graphical model are very similar. As we will see, by design, FAST is able to represent all the models in the Knowledge Tracing Family.

Figure 1 uses plate notation to describe the graphical models for the Knowledge Tracing Family of models. In plate notation, the clear nodes represent latent variables; the light gray nodes represent variables that are observed only in training; dark nodes represent variables that are both visible in train-

[*] Both authors contributed equally to the paper.



(a) Original formulation (b) Emission features (c) Transition features (d) Emission and Transition features

Figure 1: Plate diagrams of the Knowledge Tracing Family models.

Table 1: Variants of the Knowledge Tracing model

Feature	Emission	Transition	Both
Student ability		[21]	[18, 29]
Item difficulty	[10, 20]		[26]
Subskills		[28]	
Help		[25]	[3]

ing and testing; plates represent repetition of variables.

Figure 1a describes the original Knowledge Tracing formulation. Knowledge Tracing uses Hidden Markov Models [23] to model students' knowledge as latent variables. The binary observation variable ($y_{q,t}$) represents whether the student gets a question correct at the t^{th} learning opportunity of skill q. The binary latent variable ($k_{q,t}$) represents whether the student has learned the skill q at the t^{th} learning opportunity. In the context of Knowledge Tracing, the *transition probabilities* between latent states are often referred to as learning and forgetting probabilities. The *emission probabilities* are commonly referred to as guess and slip probabilities. Figures 1b and 1c describe two common modifications of Knowledge Tracing: adding features to parametrize the emission probabilities ($f_{q,t,e}$), and adding features to parametrize the transition probabilities ($f_{q,t,l}$). In this context, the features nodes are discrete or continuous variables that affect the performance of students or their learning. It is also possible to parametrize both the emission and the transition features. Table 1 summarizes some prior work that has reinvented the same graphical model with a different interpretation of the feature nodes, and are in fact part of the Knowledge Tracing Family.

To assist with the creation of models of the Knowledge Tracing Family, previous research has proposed a dynamic Bayesian network toolkit for student modeling [6]. Unfortunately, extending Knowledge Tracing using dynamic Bayesian networks is tractable only for the simplest models – exact inference on dynamic Bayesian networks is exponential in the number of parents a node has [19]. More specifically, as the number of features increases (**E** in Figure 1b and **L** in Figure 1c), the time and space complexity of the model grows exponentially. We believe that this exponential cost is the reason that although there is a plethora of Knowledge Tracing variants, they are only used for a single purpose. In the next section we describe FAST, a method that is able to generalize all models of the Knowledge Tracing family using a large number of features, but with a complexity that only grows linearly to the number of features.

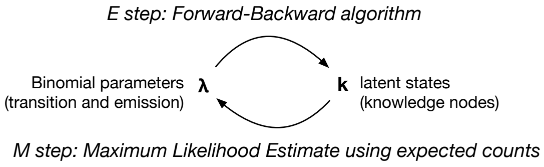

E step: Forward-Backward algorithm
Binomial parameters
(transition and emission)
λ
k
latent states
(knowledge nodes)
M step: Maximum Likelihood Estimate using expected counts

Figure 2: EM algorithm.

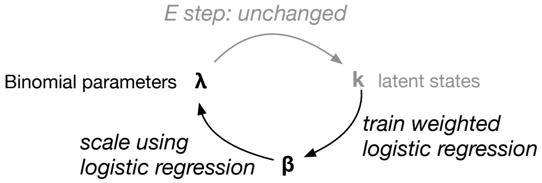


Figure 3: EM with Features algorithm [4].

3. FEATURE-AWARE STUDENT KNOWLEDGE TRACING

FAST extends Knowledge Tracing to allow features in the emissions and transitions using the graphical model structure in Figure 1d. Unlike conventional Knowledge Tracing that uses conditional probability tables for the guess, slip and learning probabilities, FAST uses logistic regression parameters. Conditional probability tables make inference exponential in the number of features, while FAST's performance is only linear in the number of features.

For parameter learning, FAST uses the *Expectation Maximization with Features* algorithm [4] – a recent modification of the original Expectation Maximization (EM) algorithm that is used in Knowledge Tracing. For simplicity, in this paper we focus on only emission features. In preliminary experiments we discovered that emission features outperform transition features, and using both did not yield a statistically significant improvement.

The rest of this section discusses parameter learning. Section 3.1 reviews the EM algorithm used in the Knowledge Tracing Family, and Section 3.2 describes the EM with Features algorithm.

3.1 Expectation Maximization

The EM algorithm is a popular approach to estimate the parameters of Knowledge Tracing. Figure 2 shows the two steps of the algorithm. The "E step", uses the current parameter estimates (λ) for the transition and emission probabilities to infer the probability the student has mastered the skill at each practice opportunity. Inferring mastery can be efficiently computed with the Forward-backward algorithm. The "M step", recomputes the parameter estimates (λ) given the estimated probabilities of mastery computed in the E step. For example, the estimate of the emission parameter of answering y' at latent state k' can be estimated as:

$$\lambda^{y',k'} = p(y = y'|k = k') \quad (1)$$

$$= \frac{\text{expected counts}\,(y = y' \wedge k = k')}{\text{expected counts}\,(k = k')} \quad (2)$$

3.2 Expectation Maximization with Features

The EM with Features algorithm was recently suggested for computational linguistics problems [4]. It uses logistic regression instead of probability tables to model features, which can be discrete or continuous, and are observed during both training and testing. Figure 3 shows the EM with Features algorithm. The E step is unchanged from the original EM algorithm, which gives the probability that the student has mastered the skill at each practice opportunity. However, the M step changes substantially: the parameters λ are now a function of weights $\boldsymbol{\beta}$ and features $\mathbf{f}(t)$. The feature extraction function $\mathbf{f}$ constructs the feature vector $\mathbf{f}(\mathbf{t})$ from the observations (rather than student responses) at the t^{th} time step For example, the emission probability from Equation 1 now is represented with a logistic function:

$$\lambda(\boldsymbol{\beta})^{y',k'} = p(y = y'|k = k'; \boldsymbol{\beta}) \quad (3)$$

$$= \frac{1}{1 + \exp(-\boldsymbol{\beta}^T \cdot \mathbf{f}(t))} \quad (4)$$

We learn $\boldsymbol{\beta}$ from data by training a weighted regularized logistic regression using a gradient-based search algorithm, called LBFGS. Training logistic regression requires a design matrix (a matrix with the explanatory variables). Figure 4 visualizes the design matrix we use. Depending on how features are encoded in the design matrix, FAST allows three different types of features: (i) features that are active only when the student has mastered the skill, (ii) features that are active only when the student has not mastered the skill, or (iii) features that are always be active. The number of features in each type (m_1, m_2, m_3 in Figure 4) can vary. By design, FAST is able to represent the models in the Knowledge Tracing Family. For example, when FAST uses only intercept terms as features for the two levels of mastery, it is equivalent to Knowledge Tracing.

To train the logistic regression, we weight each observation proportionally to the likelihood of the observation being generated from the latent states. Therefore each observation is duplicated during training: the first appearance is weighed by the probability of mastering at current observation; the second appearance is weighted by the probability of not mastering at current observation. This likelihood is calculated during the E step using the forward backward probabilities. More formally, the instance weight is :

$$w_{y',k'} = p(k = k'|\mathbf{Y}; \boldsymbol{\beta}) \quad (5)$$

Then, the Maximum Likelihood estimate $\boldsymbol{\beta}^*$ is:

$$\boldsymbol{\beta}^* = \arg\max_{\boldsymbol{\beta}} \underbrace{\sum_{y,k} w_{y,k} \cdot \log \lambda(\boldsymbol{\beta})^{y,k}}_{\text{data fit}} - \underbrace{\kappa||\boldsymbol{\beta}||_2^2}_{\text{regularization}} \quad (6)$$

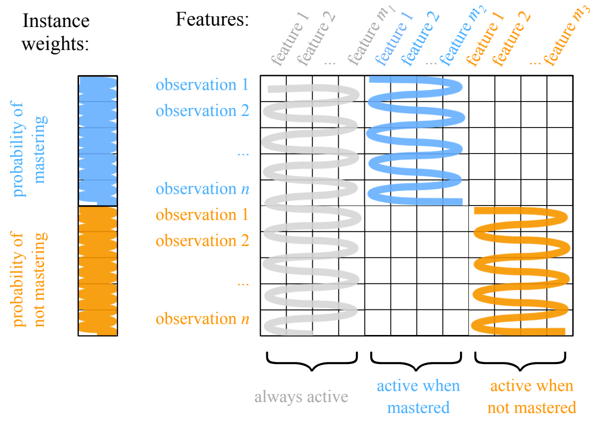


Figure 4: Feature design matrix and instance weights of FAST. During training, observations are duplicated.

where κ is a regularization hyper-parameter to penalize overfitting.

4. EXAMPLES

In this section we describe three case studies that demonstrate FAST's versatility. The rest of this section is organized as follows: Section 4.1 describes our experimental setup; Section 4.2 describes how to model subskills using FAST; Section 4.3 describes how to implement a Temporal Item Response Model using FAST; Section 4.4 describes how to use expert knowledge to improve classification performance.

4.1 Experimental Setup

We used student data collected by an online Java programming learning system called JavaGuide [12]. JavaGuide asks students to answer the value of a variable or the printed output of a parameterized Java program after they have executed the code in their mind. JavaGuide automatically assesses each result as correct or incorrect. The Java programs are instantiated randomly from a template on every attempt. Students can make multiple attempts until they master the template or give up. In total there are 95 different templates.

Experts identified skills and subskills from the templates aided by a Java programming language ontology [11]. Each item is mapped to one of 19 skills, and may use one to eight different subskills. Our dataset was collected during three semesters (Spring 2012 to Spring 2013). It consists of 20,808 observations (from which 6,549 represent the first attempt of answering an item) from 110 students. The dataset is very unbalanced since 70% of attempts are correct (60% of the first attempts are correct).

We evaluated FAST using a popular machine learning metric, the *Area Under the Curve* (AUC) of the Receiver Operating Characteristic (ROC) curve. The AUC is an overall summary of diagnostic accuracy. AUC equals 0.5 when the ROC curve corresponds to random chance and 1.0 for perfect accuracy. We reported two ways of calculating the AUC: (i) overall AUC across all data points of all skills, and (ii) average AUC of the skills as:

$$\text{average AUC} = \sum_s \frac{\text{AUC(skill } s)}{\#\text{ of skills}}$$

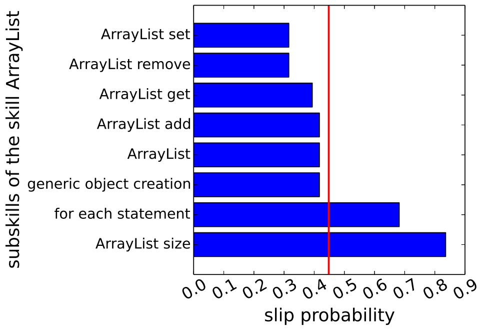


Figure 5: Subskill slip probabilities of the skill ArrayList estimated by FAST. Original Knowledge Tracing estimates the slip probability as 0.45 for the skill.

For the overall AUC, we reported the 95% confidence intervals with an implementation of the bootstrap hypothesis test method[1], a method that corrects for the non-independence of the points of the ROC. To train our classifiers, unless explicitly noted, we modeled each skill independently, and thus we have different model parameters for each of the 19 skills. In the rest of this section we discuss different feature sets we used in FAST.

4.2 Multiple Subskills

The original Knowledge Tracing formulation is designed for fitting a single skill with no subskills. A thorough survey of how prior ad hoc variants of Knowledge Tracing have accounted for multiple subskills can be found elsewhere [28].

To model subskills in FAST we just have to define binary subskill indicator features. In this way, FAST is able to estimate slip and guess probabilities as a function of the subskills present in the practice opportunity. Figure 5 compares the slip probabilities of FAST and Knowledge Tracing for the subskills that are used in the skill ArrayList. We calculate the subskill's slip probability by activating the subskill indicator and intercept in the logistic regression (using Equation 4). The original Knowledge Tracing formulation does not account for differences in subskills, and therefore estimates a single skill slip probability as 0.45. We now evaluate how this improves forecasting performance.

Table 2 compares FAST with different models previously used in the literature. For these experiments, we use a training set of a random sample of 80% of the students, and the rest of the students are used to evaluate the models. The training and testing set do not have overlapping students. We use data on all of the attempts students had to solve an item. We make predictions on all observations of the students in the test set, and evaluate them using overall AUC and mean AUC (when defined). The models we compare are:

- **FAST:** FAST using subskill binary indicator features. We allow FAST to learn different coefficients for guess and slip for each subskill.
- **PFA:** Performance Factors Analysis [22] has been shown to be effective in modeling multiple subskills [8].
- **LR-DBN:** A Knowledge Tracing Family member that uses binary subskill indicators as transition features [28].

Table 2: Overall AUC for multiple subskills experiments.

Model	Overall AUC
FAST	.74 ± .01
PFA	.73 ± .01
LR-DBN	.71 ± .01
KT(single skill)	.71 ± .01
KT(weakest)	.69 ± .01
KT(multiply)	.62 ± .02

- **KT:** We evaluate different Knowledge Tracing variants:
  - **single skill:** We fit each skill independently (original formulation with no subskills).
  - **weakest:** We fit each subskill independently, and then take the minimum of each subskill's predicted probability of success as the final prediction. We update the knowledge of this weakest subskill by the actual response evidence while we update other subskills by the correct response [8].
  - **multiply:** We fit each subskill independently, and then multiply each subskill's predicted probability of success as the final prediction. We then update the knowledge of each subskill by the same actual response evidence [8].

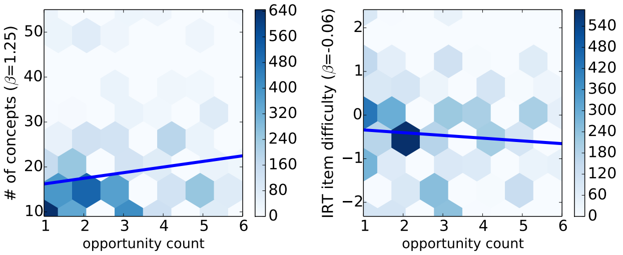
FAST significantly outperforms all the above Knowledge Tracing variants. In particular, FAST improves the overall AUC of KT(multiply) by about 19% (significant at $p \ll .01$), and outperforms KT(weakest) by over 7% (significant at $p \ll .01$). We hypothesize that FAST's better performance comes from estimating each subskill's responsibility using a logistic regression. This avoids Knowledge Tracing variants' crude assumption that each subskill accounts equally for a correct or incorrect response during the parameter learning. FAST also outperforms the LR-DBN by 4% (significant at $p < .003$), which may indicate parameterizing emission probabilities is better than parameterizing transitions in this dataset. Improving over the original Knowledge Tracing formulation (significant at $p < .002$) suggests that modeling subskills is important. The fact that LR-DBN does not improve over Knowledge Tracing questions its usefulness. We do not find statistically significant differences between FAST and PFA using this feature set in our dataset.

4.3 Temporal Item Response Theory

Knowledge Tracing and classical psychometric paradigms, such as *Item Response Theory* (IRT), treat item difficulty in different ways. The Knowledge Tracing paradigm assumes that all items for practicing a skill have the same difficulty [17]. For example, when a student struggles on some items, the paradigm explains it by assuming there is some subskill(s) that the student has yet to acquire. This paradigm requires a very careful definition of skills and subskills, which may be a very expensive requirement – skill definitions are often done manually by an expert. A cheaper alternative is to discover the skill definitions using data, but such methods are still a relatively new technology [9].

On the other hand, IRT explicitly models item difficulty and student ability. For example, the Rasch model [24], the simplest IRT model, assumes that the correctness of a student's response on an item depends on the student ability and the item difficulty. Unfortunately, IRT models are static, and

[1] http://www.subcortex.net/research/code/



(a) Number of concepts (b) Item difficulty

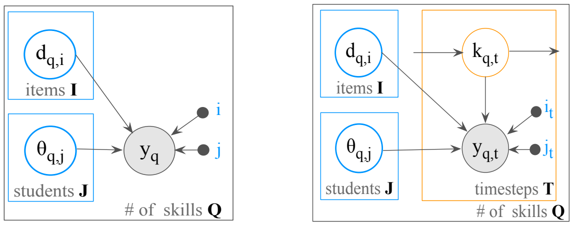
Figure 6: Although experts consider the item complexity increases in latter items, IRT estimates items become easier. Hexagon colors indicate the number of points that fall within the region. Binning is necessary to see overlapping points.

therefore, unlike Knowledge Tracing, do not account for student learning.

Prior Knowledge Tracing variants have been proposed to bridge between the Knowledge Tracing and IRT paradigm. For example, Table 1 summarizes different models that try to account for different student abilities or item difficulties in a learning environment. In addition, Latent-Factor Knowledge Tracing (LFKT) [15], a recent single-purpose specialized graphical model, bridges between both paradigms. FAST takes an alternative approach and models Item Response Theory using feature engineering. Although Rasch Model is typically formulated with latent variables for item and student differences, it can also be estimated using logistic regression with binary variables indicators (sometimes called dummy variables) for each student and each item [**?**, **?**].

In Figures 6a and 6b we show binned scatter plots of the item complexity and the estimated IRT difficulty, respectively. The complexity of an item is defined objectively by experts counting the number of Java concepts used in a question.The item difficulty is estimated by a Rasch model. A higher number of concepts and a higher value of item complexity represent harder items. We run two univariate linear regressions to fit item complexity and difficulty as a function of number of practice opportunities. Experts consider that items practiced later in the tutor are more complex (β=1.25, p<.0005), while IRT estimates that items become easier (β=-0.06, p<.0005). The mismatch between IRT difficulty and item complexity happens because learning is getting confounded with item difficulty. Even though items are becoming harder, students are learning, and thus getting better at them, resulting in the underestimation of item difficulty.

We believe we are the first to show this confounding. It is unclear if prior work is able to deliver the promise of accounting for student learning in the presence of items with different difficulty, particularly because prior work only evaluates on classification accuracy. Aware of this caveat, we use FAST using IRT features. As in previous work, we estimated item difficulties using longitudinal data from students answering items in almost the same order. However, we suggest that future work should calibrate the item difficulty in a controlled experiment first. This would be an easy modification for FAST, as it would only require using a continuous feature (item difficulty) instead of discrete variables (item indicators). This would not be easy in previous work – a new ad hoc model would be required.



(a) Rasch model of IRT (b) FAST using IRT features

Figure 7: Static IRT (Rasch) and temporal IRT models.

Table 3: Overall and average AUC for IRT experiments.

features	static		temporal	
	all	avg.	all	avg
none	.65 ± .03	.50	.67 ± .03	.56
+student	.64 ± .03	.59	.67 ± .03	.60
+item	.73 ± .03	.63	.73 ± .03	.63
+IRT	.76 ± .03	.70	.76 ± .03	.70

Figures 7a and 7b show the plate diagram of the Rasch model and temporal IRT as a Knowledge Tracing Family member. We do not have to change anything in FAST: our temporal IRT definition uses student and item binary indicator features. The probability of getting a correct response y' of student j for item i on skill q, at the t^{th} time step is:

$$p(y'_{q,t}|\mathbf{Y}) = \sum_{\substack{l\in\{\text{mastered},\\ \text{not mastered}\}}} p(k_{q,t} = l|\mathbf{Y}) \cdot \text{Rasch}(d_{q,i_t}, \theta_{q,j_t}, c_{q,l}) \quad (7)$$

where **Y** is the corresponding observed sequence. Here, the Rasch function is parametrized with item difficulty d, student ability θ and and a bias c that is specific to whether or not the student has mastered the skill. Both Knowledge Tracing and IRT can be recovered from the combined model with different choices of parameter values. For example, when abilities and difficulties are zero, the combined model is equivalent to Knowledge Tracing. When bias terms are the same (i.e., $c_{q,\text{not mastered}} = c_{q,\text{mastered}}$), we get IRT.

Table 3 evaluates FAST using IRT features. To get a better estimate of item difficulty, for these experiments we only use data on the first attempt of a student solving an item. The training set is a random sample of 50% of the students. For students in the test set, we observe the first half of their practice opportunities and make predictions in the second half of their practice opportunities. We compare static models that assume no learning using logistic regression. For the temporal models we use Knowledge Tracing (with no features) or FAST (with features). We experiment with the following feature sets:

- **none**. No features are considered for classifiers.
- **student**. We only use student indicator features.
- **item**. We only use item indicator features.
- **IRT**. We use both item and student indicator features.

FAST using IRT features outperforms the overall AUC of Knowledge Tracing by over 13% (significant at p≪.01) and the mean AUC of Knowledge Tracing by 25%. Using item features in FAST improves the overall AUC of Knowledge Tracing by about 9% (significant at p≪.01) and the mean

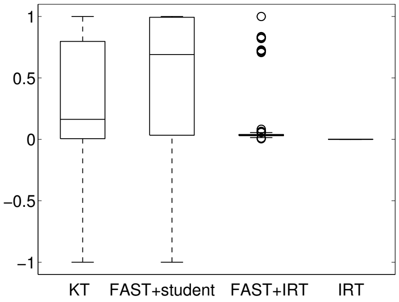


Figure 8: Boxplot of Knowledge Tracing, FAST and IRT's estimates of student learning. The whiskers indicate minimum and maximum values, the lines in the box indicate quartiles. IRT always estimates zero learning.

AUC by 13%. The Follow-up work [?] confirms in other datasets that FAST with IRT features is able to significantly outperform Knowledge Tracing.The overall AUC of Knowledge Tracing is not significantly different from that of logistic regression with no features (p>.2). The difference between the mean and overall AUC of these models with no features should be accounted by the expert knowledge introduced by the expert's skill definition. These results coincide with previous work: adding item difficulty and student ability features can greatly increase the performance of Knowledge Tracing. However we do not find any significant differences between the temporal and static models. This may be because the item difficulty is confounded with learning.

Even if Knowledge Tracing with IRT features does not outperform IRT, such a model could be preferable to IRT if it allows modeling student learning. We estimate student learning by subtracting the probability of mastery at the first practice opportunity from the probability of mastery at the last practice opportunity for each student-skill sequence:

$$\text{learning} \equiv p(k_{q,T} = \text{mastered}|\boldsymbol{Y}) - p(k_{q,0} = \text{mastered}|\boldsymbol{Y})$$

Figure 8 shows the box plot of average learning of students across all skills for the models. Both Knowledge Tracing and FAST+student are able to capture student learning from data much better than models that account for item difficulty. In our literature review from Table 1, none of the methods reported both a comparison with IRT and this analysis of learning. We do not know if our results are typical, but they suggest that item difficulty is confounded with learning. Future work should study using FAST with item difficulties calibrated in a controlled experiment to avoid confounding with item difficulties.

4.4 Expert Knowledge

In this section we perform feature engineering to improve student performance prediction. We use features that previous work has found to be useful to predict student performance [10, 20, 22, 26]. More concretely, we demonstrate FAST with three types of features: continuous features that indicate the number of prior practice opportunities where the student answered (i) correctly, and (ii) incorrectly the item template, and (iii) item indicator features. The item indicator features correspond to a binary indicator per item (95 indicator features in total). We use the same experimental setup of Section 4.2, which is a non-overlapping set of students for training and testing, and data from all attempts. Table 4 compares the following models:

Table 4: Overall and average AUC for item practice feature experiments

Model	all	avg.
FAST+item+practice	$.77 \pm .01$	.73
FAST+item	$.75 \pm .01$	.68
FAST+practice	$.72 \pm .01$	.67
PFA	$.70 \pm .01$	.60
KT	$.71 \pm .01$	.58

- **FAST+item+practice** uses item indicator features and the numbers of prior correct and incorrect practices features for each item. The coefficients of item practice performance features can be interpreted as learning rates from correct and incorrect practices of an item.
- **FAST+item** uses item indicator features.
- **FAST+practice** uses only the numbers of prior correct and incorrect practices as features.
- **PFA** (Perfomance Factors Analysis) model uses skill indicator, the numbers of prior correct and incorrect practices feature of the skill as features.
- **KT** is the original Knowledge Tracing without features.

The most predictive model is FAST using item difficulty and prior practice features, which outperforms the overall AUC of KT over 8% (significant at $p \ll .01$) and outperforms PFA by 10% ($p \ll .01$). Its mean AUC also beats KT by 26% and PFA by 22%. The best model outperforms FAST with only item indicator features with an improvement of 3% of overall AUC (significant at $p < .005$) and 7% of mean AUC, indicating that adding item practice features can provide significant gain over just using item difficulty parameters. When FAST uses only practice features we do not find a significant difference from PFA and Knowledge Tracing in terms of overall AUC (p>.1), but it shows improvement using the mean AUC metric – improving PFA by 12% and Knowledge Tracing by 16%. Future research should investigate whether this discrepancy between the overall AUC and mean AUC is because of the distribution of common and rare skills or because of the quality of the item to skill mapping. Our results suggest that feature engineering can improve the forecasting quality in Knowledge Tracing.

5. EXECUTION TIME

We now study the execution time of learning the parameters and making predictions. As a comparison, we use a popular tool that facilitates the creation of ad hoc Knowledge Tracing variants called BNT-SM [6]. We conduct the experiments on a contemporary laptop (1.8 GHz Intel Core i5 CPU and 4GB RAM). We compare FAST and BNT-SM under two settings: (i) tracing single skills as the standard Knowledge Tracing and (ii) tracing multiple subskills. For the Knowledge Tracing experiment, Figure 9 shows the execution time of both algorithms varying the dataset size, and FAST is about 300 times faster. For the multiple subskill experiment, we compare with LR-DBN, a recent method [28] implemented on BNT-SM. We use the authors' implementation of LR-DBN. LR-DBN takes about 250 minutes while FAST only takes about 44 seconds on 15,500 datapoints. We didn't report results varying dataset since LR-DBN requires much time. The execution time of LR-DBN is comparable to the one reported by LR-DBN authors.

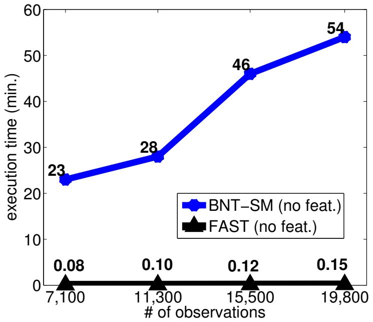


Figure 9: Execution time (in minutes) of FAST and BNT-SM with different sizes of dataset on single skill experiment

In both experiments, FAST's parameter fitting time can be up to 300 times faster — while keeping the same or better performance in terms of overall AUC (significant at p<.05). Our implementation of FAST is in Java, while BNT-SM is implemented in Matlab. It is contentious to measure the effect of the programming language in the experiment. However, some informal benchmarks [2] suggest that Matlab is in fact faster for scientific computations.

6. RELATION TO PRIOR WORK

The likelihood functions of FAST and Knowledge Tracing are non-convex. Therefore, parameters discovered might only be local optima, not a global solution. In this paper we only experiment with the Expectation Maximization algorithm, but future work may compare multiple fitting procedures to avoid local solutions [8]. Prior work [2] has also used regression as a post-processing step after Knowledge Tracing. This is different from FAST's approach of jointly training (logistic) regression and Knowledge Tracing. We leave for future work the analysis and comparison of FAST and the post-hoc analysis.

Table 5 compares FAST with models from the literature whether they allow general features, slip and guess probabilities, time, and multiple subskills. For the time dimension we consider whether the models consider recency (R), ordering (O) and learning (L) or none. The Rasch model [24] and Performance Factor Analysis (PFA) [22] use logistic regression and may model arbitrary features. However, Rasch can not account for student learning or multiple subskills. Although PFA is able to fulfill these two cases, it does not consider recency (a correct response following an incorrect response is modeled in the same way as an incorrect response following a correct response), or ordering (a question that was answered incorrectly recently is modeled in the same way as if it were answered incorrectly two weeks ago). Furthermore, PFA does not model slip and guess probabilities.

Knowledge Tracing [7] has a robust mechanism to model time, but lacks the ability to allow arbitrary features and multiple subskills. LR-DBN is a variant of Knowledge Tracing that uses logistic regression [28], yet it is proposed for modeling subskills but not general features. Moreover, in Section 4 we report experiments in which FAST has better predictive performance than LR-DBN. Unlike prior work, FAST is a general method that is able to fulfill a wide range of use cases.

Table 5: Model Comparison

Model	features	slip / guess	time	multiple subskills
FAST	✓	✓	R,O,L	✓
LR-DBN		✓	R,O,L	✓
KT		✓	R,O,L	
PFA	✓		L	✓
Rasch	✓			

7. CONCLUSION

In this paper we identified a family of models, the Knowledge Tracing Family, that have similar graphical model structures. The graphical model structures of the Knowledge Tracing Family have been reinvented multiple times for different applications. We presented FAST as a flexible and efficient method that allows representing all of the models in the Knowledge Tracing Family. FAST uses logistic regression to model general features in Knowledge Tracing. Previous student modeling frameworks [6] for Knowledge Tracing are inefficient because their time and space complexity is exponential in the number of features. FAST is very efficient, and its complexity only grows linearly to the number of features.

Although theoretically FAST should be very similar to the conventional Knowledge Tracing Family implementations, future work may run a more detailed comparison. A limitation of this study is that we did not compare against all of the previous implementations of the Knowledge Tracing Family.

A secondary contribution of this paper is that we identified a problem of prior published work of learning item difficulty from within Knowledge Tracing. The resulting item difficulty estimates are confounded with learning. FAST is also susceptible to this problem, and in future work we will use FAST with item difficulty estimates calibrated with a controlled study. Additionally, future work may study alternative ways of training FAST [4] and discovering an item to skill mapping.

We demonstrated FAST's generality with three use cases that have been shown to be important in prior work: (i) modelling subskills, (ii) incorporating IRT features in Knowledge Tracing, and (iii) using features designed by experts. In our experiments we see improvements of FAST over Knowledge Tracing by up to 13% in mean AUC of skills, and 25% in the overall AUC. When and how feature engineering can assist in student modeling depends on the characteristics of the data and the experience of domain experts. FAST provides high flexibility in utilizing features, and as our studies show, even with simple general features, FAST presents much improvement over Knowledge Tracing. We expect more thorough feature engineering for FAST in the future should provide greater improvement. Moreover, FAST is efficient for model fitting and inferencing — FAST can be 300 times faster than models created in other general purpose student modeling toolkits while keeping the same or better classifier performance.

[2] https://modelingguru.nasa.gov/docs/DOC-1762

Acknowledgements

This research is supported by Pearson[3] and the Advanced Distributed Learning Initiative[4]. We also thank the 2013 LearnLab Summer School for enabling the cooperation.

8. REFERENCES

[1] A. Agresti. Computing conditional maximum likelihood estimates for generalized rasch models using simple loglinear models with diagonals parameters. *Scandinavian Journal of Statistics*, pages 63–71, 1993.

[2] R. Baker, A. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In B. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, editors, *ITS*, volume 5091 of *Lecture Notes in Computer Science*, pages 406–415. Springer, 2008.

[3] J. E. Beck, K.-m. Chang, J. Mostow, and A. Corbett. Does help help? introducing the bayesian evaluation and assessment methodology. In *Intelligent Tutoring Systems*, pages 383–394. Springer, 2008.

[4] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, 2010.

[5] R. D. Bock and M. Aitkin. Marginal maximum likelihood estimation of item parameters: Application of an em algorithm. *Psychometrika*, 46(4):443–459, 1981.

[6] K.-m. Chang, J. Beck, J. Mostow, and A. Corbett. A bayes net toolkit for student modeling in intelligent tutoring systems. In *Intelligent Tutoring Systems*, pages 104–113. Springer, 2006.

[7] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[8] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*, pages 35–44. Springer, 2010.

[9] J. P. González-Brenes and J. Mostow. What and When do Students Learn? Fully Data-Driven Joint Estimation of Cognitive and Student Models. In A. Olney, P. Pavlik, and A. Graesser, editors, *EDM*, pages 236–240, Memphis, TN, 2013.

[10] S. M. Gowda, J. P. Rowe, R. S. J. de Baker, M. Chi, and K. R. Koedinger. Improving models of slipping, guessing, and moment-by-moment learning with estimates of skill difficulty. In *EDM*, pages 199–208, 2011.

[11] R. Hosseini and P. Brusilovsky. Javaparser: A fine-grain concept indexing tool for java problems. In *The First Workshop on AI-supported Education for Computer Science*, page 60, 2013.

[12] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. *Journal of Computer Assisted Learning*, 26(4):270–283, 2010.

[13] Y. Huang, Y. Xu, and P. Brusilovsky. Doing more with less: Student modeling and performance prediction with reduced content models. In *the 22nd Conference on User Modeling, Adaptation and Personalization*, 2014.

[14] M. Khajah, Y. Huang, J. P. González-Brenes, M. C. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *In submission*, 2014.

[15] M. Khajah, R. M. Wing, R. V. Lindsey, and M. C. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *In submission*, 2014.

[16] K. R. Koedinger, R. S. J. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. In *Handbook of Educational Data Mining*, pages 43–55.

[17] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[18] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the 5th International Conference on Educational Data Mining*, pages 118–125, 2012.

[19] K. Murphy et al. The bayes net toolbox for matlab. *Computing science and statistics*, 33(2):1024–1034, 2001.

[20] Z. Pardos and N. Heffernan. $kt-idem$: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*. 2011.

[21] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *Proceedings of the 18th international conference on User Modeling, Adaptation, and Personalization.*

[22] P. Pavlik, H. Cen, and K. Koedinger. Performance Factors Analysis–A New Alternative to Knowledge Tracing. In *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 531–538. IOS Press, 2009.

[23] L. Rabiner and B. Juang. An introduction to Hidden Markov Models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

[24] G. Rasch. Probabilistic models for some intelligence and attainment tests. In *Paedagogike Institut, Copenhagen.*, 1960.

[25] M. A. Sao Pedro, R. S. Baker, and J. D. Gobert. Incorporating scaffolding and tutor context into bayesian knowledge tracing to predict inquiry skill acquisition. In *EDM*, pages 185–192, 2013.

[26] S. Schultz and T. Tabor. Revisiting and extending the item difficulty effect model. In *In Proceedings of the 1st Workshop on Massive Open Online Courses at the 16th Annual Conference on AIED*, pages 33–40, Memphis, TN., 2013.

[27] B. D. Wright and G. A. Douglas. Best procedures for sample-free item analysis. *Applied Psychological Measurement*, 1(2):281–295, 1977.

[28] Y. Xu and J. Mostow. Comparison of methods to trace multiple subskills: Is LR-DBN best? In *EDM*, pages 41–48, 2012.

[29] M. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *AIED*, pages 171–180, Memphis, TN, 2013. Springer.

[3] `http://researchnetwork.pearson.com/`

[4] `http://www.adlnet.gov/`

Generating Hints for Programming Problems Using Intermediate Output

Barry Peddycord III
North Carolina State University
890 Oval Drive
Raleigh, NC, 27695
bwpeddyc@ncsu.edu

Andrew Hicks
North Carolina State University
890 Oval Drive
Raleigh, NC, 27695
aghicks3@ncsu.edu

Tiffany Barnes
North Carolina State University
890 Oval Drive
Raleigh, NC, 27695
tmbarnes@ncsu.edu

ABSTRACT

In this work, we compare two representations of student interactions within the context of a simple programming game. We refer to these representations as Worldstates and Codestates. Worldstates, which are representations of the output of the program, are generalizations of Codestates, snapshots of the source code taken when the program is run. Our goal is to incorporate intelligent data-driven feedback into a system, such as generating hints to guide students through problems. Using Worldstates simplifies this task by making it easier to compare student approaches, even for previously unseen problems, without requiring expert analysis. In the context of the educational programming game, BOTS, we find that worldstates require less prior data to generate hints in a majority of cases, without sacrificing quality or interpretability.

Keywords

Hint Generation, Programming Tutor, Educational Game

1. INTRODUCTION

One key benefit of Intelligent Tutoring Systems over other computer-aided instruction is the ability to provide intelligent adaptive feedback. A popular way of providing this feedback is through the generation of hints. Hints can help students who are stuggling by suggesting a next step or providing a clue about what the next step might be. While some of the earliest work in this area focuses on building models of the learner [7], recent work shows that quality hints can be generated in certain domains using data-driven methods, informed by the type and frequency of actions taken by students in the system [1].

Programming has been a domain of interest for tutoring systems as far back as the Lisp Tutor [3]. There has been recent interest in trying to apply hint generation techniques such as Stamper and Barnes' Hint Factory [9] to programming languages [6, 8], but this still remains an open problem given the complexity associated with learning programming languages. One of the challenges associated with handling programming tutors comes from the diversity of possible programs that a student can write.

2. PROBLEM STATEMENT

Our work is an effort to add techniques from Intelligent Tutoring Systems to an educational game called BOTS. BOTS is an educational programming game designed to teach middle school students the principles of programming, and also allows students to create their own puzzles for other students to solve. Due to the rapid creation of new puzzles, it is necessary that hints can be generated with relatively little data, since expert authoring is infeasible.

Like Rivers, our work is based on Hint Factory, but rather than attempting to analyze the student source code, our work looks entirely at the *output* of the programs. We hypthesize that using the output of programs for hint generation allows us to deal with the challenge of source code diversity and generate more hints with less data than using source code alone, without diminishing the quality of hints.

Though the hints have not yet been integrated into the game, this work shows that our technique is promising and could feasibly be integrated into the game for future studies.

3. PRIOR WORK

One popular technique for automatic hint generation that has enjoyed success is Stamper and Barnes' Hint Factory [9]. Hint Factory can be applied in any context where student interactions with a tutor can be defined as a graph of transitions between states [2]. In the Deep Thought propositional logic tutor, students are asked to solve logic proofs by deriving a conclusion from a set of premises [1]. Each time a student applies a logical rule, a snapshot of the current state of the proof is recorded in a graph called the *interaction network* as a node, with an edge following the states along the path they follow to get to their proof. At any point during the proof, a student can ask for a hint, and Hint Factory selects the edge from the state they are currently in that takes them closest to the solution.

The iList Linked List tutor is another example where a Hint Factory-based approach to generating feedback has been successful [4]. In Fossati's work on the iList Linked List tutor, the developers used a graph representation of the program's state as the input to their system. In order to give hints for semantically equivalent states, rather than for only precisely identical states, the developers searched for isomorphisms between the program state and the previously observed states. This is a non-trivial comparison, and requires significant knowledge of the domain in order to assess the best match when multiple such relations exist.

Rivers et al also use Hint Factory to generate hints specifically for programming tutors. Due to the complexity of programming problems, a simple snapshot of the code will not suffice [8]. There are many varied approaches that can be taken to programming problems, and if a direct comparison is being used, it is rare that a student will independently have a precisely identical solution to another student. Therefore, some way of reducing the size of the state space is needed. During a programming problem, each time a student saves his or her work, their code is put through a process of canonicalization to normalize the naming conventions, whitespace, and other surface-level programming variations that differ widely across students. When two programs have the same canonical form, they are considered the same state in the interaction network. In both cases, the generation of a hint comes from selecting the appropriate edge and then articulating to the student some way of getting from their current state to the state connecting by that edge.

In Jin's work on programming tutors [6] a similar approach is taken. Rather than using canonical forms generated using abstract syntax trees, the authors generate "Linkage Graphs" which define the relationships between variables in the program, then condense those graphs by removing intermediate variables, creating abstract variables that represent multiple approaches. To take a very simple example, if the goal is to multiply A by B and somehow output the result, a programmer may write `A = A * B` or `C = A * B`. After building the CLG, these approaches would be the same. If the condensed linkage graphs (CLGs) derived from two different programs are isomorphic to eachother, then those programs are said to be similar. Additional analysis is then needed to identify which concepts the abstracted variables in the CLGs represent; in this case, k-means clustering was used to find the most similar abstracted variable, then choosing the most common naming convention for that variable.

We propose that another way of canonicalizing student code is to use the intermediate output of programs. In a study of 30,000 student source code submissions to an assignment in the Stanford Massive Open Online Course on Machine Learning, it was found that there were only 200 unique outputs to the instructor test cases [5]. In the Machine Learning class, there was only one correct output, but despite there being an infinite number of ways to get the problem wrong, Huang et al observed a long tail effect where most students who got the exercises wrong had the same kinds of errors.

4. CONTEXT

4.1 BOTS

BOTS is an educational programming game that teaches the basic concepts of programming through block-moving puzzles. Each problem in BOTS, called a "puzzle", contains buttons, boxes, and a robot. The goal for the player is to write a program for the robot, using it to move boxes so that all of the buttons are held down. The BOTS programming language is quite simple, having elementary robot motion commands ("forward", "turn", "pickup/ putdown") and basic programming constructs (loops, functions, variables). Most programs are fairly small, and students who solve puzzles using the fewest number of blocks overall (fewer "lines of code") are shown on the leaderboard.

One important element of BOTS is that there is a built-in level editor for students to develop their own puzzles. There is a tutorial sequence that contains puzzles only authored by the developers of the system, but students are also free to attempt peer-authored puzzles which may only be played by a few students. Due to the constant addition of new puzzles, expert generation of hints is infeasible, and due to the limited number of times levels will be played, it must be possible to generate hints with very little data.

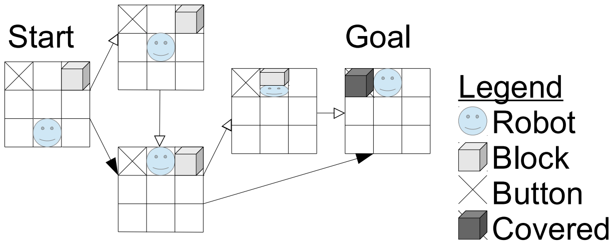
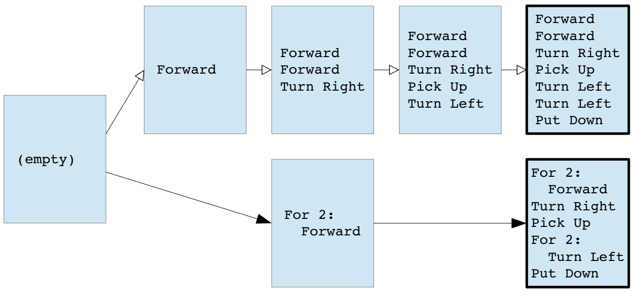


Figure 1: An example of the interaction network of a BOTS puzzle solved two ways by two students. One student (empty arrow) wrote four programs to arrive at the solution, and the other (filled arrow) only wrote two.



```
(empty)
```

```
Forward
```

```
Forward
Forward
Turn Right
```

```
Forward
Forward
Turn Right
Pick Up
Turn Left
```

```
Forward
Forward
Turn Right
Pick Up
Turn Left
Turn Left
Put Down
```

```
For 2:
  Forward
```

```
For 2:
  Forward
Turn Right
Pick Up
For 2:
  Turn Left
Put Down
```

Figure 2: This figure shows the same two students with the same solutions. Notice how in this example, these two paths that get to the same goal have no overlapping states - including the goal state.

We use a data structure called an *interaction network* to represent the student interactions with the game [2]. An interaction network is a graph where the nodes represent the states that a student is in, and the edges represent transitions between states. In this research, we compare the effects of using two different types of states in the interaction network, which we refer to as *codestates* and *worldstates*.

Codestates are snapshots of the source code of a student's program, ignoring user generated elements like function and variable names. Worldstates, inspired by Jin and Rivers, use the configuration of the entities in the puzzle as the definition of the state. In other words, codestates represent student source code while worldstates represent the output of student source code. In both cases, student interactions can be encoded as an interaction network, which enables the application of techniques like Hint Factory. Figures 1 and 2 demonstrate how the interaction networks differ between codestates and worldstates.

4.2 Hint Generation

When generating a hint, the goal is to help a student who is currently in one state transition to another state that is closer to the goal, without necessarily giving them the answer. In this work, rather than attempting to identify these states *a priori* using expert knowledge, we instead use the solutions from other students to estimate the distance to a solution. It is a fair assumption that students who solve the same problem will use similar approaches to solve it [1]. This assumption implies that there will be a small number of states that many students visit when they solve a problem. If several students take the same path from one state to another, it is a candidate for a hint. Hint Factory formalizes this process with an algorithm.

"Generating" a hint is a two-part problem. First, we must devise a *hint policy* to select one of the edges from the user's current state in the interaction network. Then, we must *articulate* the resultant state into a student-readable hint. For example, if we applied hint factory to Figure 1 when the student is in the start state, the edge selected would be the one going down to the bottom-most worldstate. The "hint" might be articulated to the student as "write a program that moves the robot north by two spaces". An example is provided in Figure 3

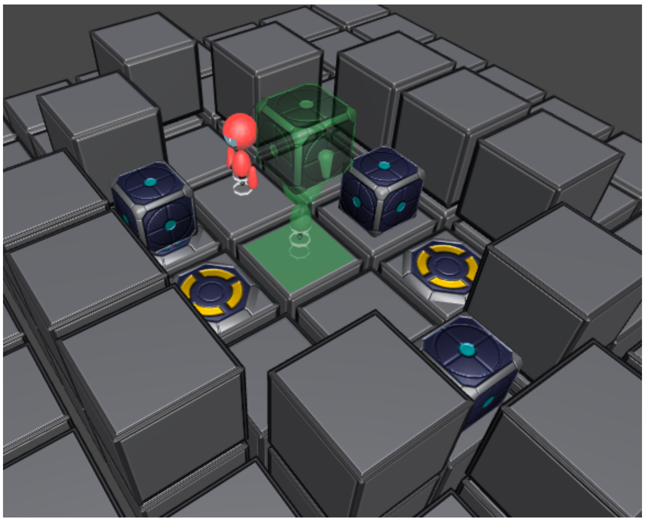


Figure 3: A mock-up of how a high-level hint might be presented in BOTS. The green "hologram" of the robot indicates the next worldstate the player should attempt to recreate.

In order to give a student a relevant hint, another student must have reached the solution from the same state that the one requesting a hint is currently in. If no other student has ever been in that state before, we can not generate an exact hint. Using source code to determine the state is challenging, since students can write the same program in many different ways, which makes the number of states across all students highly sparse, reducing the chance that a match (and by extension, a hint) will be available. Previous work attempts to find ways to canonicalize student program [8], but even then the variability is still very high. Using worldstates, however, serves to "canonicalize" student submissions without having to do complicated source code analysis.

When it comes to the articulation of hints, we can use whatever information is available in the interaction network to provide a hint to the student. Using codestates, a diff between the source code of the current state and the hint state can be used. For worldstates, an animation of the path of the robot can be played to show what the hint state's configuration would look like. In tutoring systems like Deep Thought, hints are given at multiple levels, with the first hint being very high-level, and the last essentially spelling out the answer to the student (a "bottom-out" hint) [1]. In BOTS, we can progressively reveal more information as necessary - hints about the worldstate help a student see how to solve the puzzle without telling them exactly how to code it. If a student is having trouble with the code itself, then lower-level hints might suggest which kinds of operations to use or show them snippets of code that other students used to solve the puzzle.

It is important to note that while worldstates are a generalization, we do not necessarily lose any information when using them for hint generation. BOTS programs are deterministic, and there is no situation where the same code in the same puzzle produces a different output. Therefore, using worldstates not only allows us to articulate high-level hints, it also provides a fallback when a student's source code snapshot is not yet in the interaction network.

5. METHODS

5.1 Data set

The data for this study comes from the 16-level tutorial sequence of BOTS. These levels are divided into three categories: demos, tutorials, and challenges. Demos are puzzles where the solution is pre-coded for the student. Tutorials are puzzles where instructions are provided to solve the puzzle, but the students build the program themselves. Finally, challenges require the student to solve a puzzle without any assistance. Challenge levels are interspersed throughout the tutorial sequence, and students must complete the tutorial puzzles in order - they can not skip puzzles.

For the purposes of this evaluation, we exclude demos from our results, and run our analysis on the remaining 8 tutorials and 5 challenges. The data comes from a total of 125 students, coming from technology-related camps for middle school students as well as an introductory CS course for non-majors. Not all students complete the entire tutorial sequence, as only 34 of the students attempted to solve the final challenge in the sequence. A total of 2917 unique code submissions are collected over the 13 puzzles, though it is important to note that the number of submissions spike when students are presented with the first challenge puzzle. This information is summarized for each puzzle in Table 1.

Table 1: A breakdown of the tutorial puzzles in BOTS, listed in the order that students have to complete them. Hint states are states that are an ancestor of a goal state in the interaction network, and are the states from which hints can be generated.

		Codestates		Worldstatates	
Name	#Students	Hint	All	Hint	All
Tutorial 1	125	89	162	22	25
Tutorial 2	118	36	50	12	14
Tutorial 3	117	130	210	22	24
Tutorial 4	114	137	225	33	41
Tutorial 5	109	75	106	25	29
Challenge 1	107	348	560	143	191
Challenge 2	98	201	431	86	133
Tutorial 6	90	107	143	33	36
Challenge 3	89	192	278	28	30
Challenge 4	86	137	208	40	45
Tutorial 7	76	206	383	43	57
Tutorial 8	68	112	134	29	30
Challenge 5	34	17	27	13	17

In order to demonstrate the effectiveness of using worldstates to generate hints, we apply a technique similar to the one used to evaluate the "cold start" problem used in Barnes and Stamper's work with Deep Thought [1]. The cold start problem is an attempt to model the situation when a new problem is used in a tutor with no historical data from which to draw hints. The evaluation method described in Barnes and Stamper's previous work uses existing data to simulate the process of students using the tutor, and provides an estimate as to how much data is necessary before hints can be generated reliably. As such, it is an appropriate method for determining how much earlier - if at all - worldstates generate hints as opposed to codestates.

We break the student data for each puzzle into a training and a validation set, and iteratively train the Hint Factory by adding one student at a time. We chart how the number of hints available to the students in the validation set grows as a function of how much data is in the interaction network, and average the results over 1000 folds to avoid ordering effects. The specific algorithm is as follows:

Step 1 Let the Validation set = 10 random students, and the training set = the n-10 remaining students

Step 2 Randomly select a single student attempt from the training set

Step 3 Add states from the student to the interaction network and recalculate the Hint Factory MDP

Step 4 Determine the number of hints that can be generated for the validation set

Step 5 While the training set is not empty, repeat from step 2

Step 6 Repeat from step 1 for 1000 folds and average the results

This approach simulates the a cohort of students asking for hints at the same states as a function of how much data is already in the system, and provides a rough estimate as to how many students need to solve a puzzle before hints can be generated reliably. However, this approach is still highly vulnerable to ordering effects, so to verify a hypothesis that n students are sufficient to generate hints reliably, we do cross validation with training sets of n students to further establish confidence in the hint generation.

6. RESULTS

6.1 State-space reduction

At a glance, Table 1 shows how using worldstates as opposed to codestates reduces the state space in the interaction network. Challenge 1, for example, has 560 unique code submissions across the 107 students who attempted the puzzle. These 560 code submissions can be generalized to 191 unique outputs.

We see a significant reduction in the number of states containing only a single student. Intuitively, students who get correct answers will overlap more in terms of their solutions, while the infinite numbers of ways to get things wrong will result in several code submissions that only a single student will ever encounter. In the Table 2, we look at the number of frequency-one states that students encounter for challenge puzzles 1 through 4.

When using worldstates, in all three cases, more than half of the states in the interaction network are only observed one time. In particular, Challenge 3 is particularly interesting, considering that out of 278 unique code submissions from 89 students, only 8 of the 30 worldstates are observed more than once. The sheer degree of overlap demonstrates that worldstates in particular meet the assumptions necessary to apply hint factory.

Table 2: This table highlights the number of code and worldstates that are only ever visited one time over the course of a problem solution.

		Codestates		Worldstatates	
Name	#Students	All	Freq1	All	Freq1
Challenge 1	107	560	146	191	112
Challenge 2	98	431	127	133	84
Challenge 3	89	278	91	30	22
Challenge 4	86	208	65	45	36

6.2 Cold Start

The graphs in Figure 4 show the result of the cold start evaluation for all 13 of the non-demo puzzles. Looking at the two graphs side-by-side, the worldstates have more area under the curves, demonstrating the ability to generate hints earlier than codestates. The effect is particularly pronounced when looking at the challenge puzzles (represented with the solid blue line). In tutorials, code snippets are given to the students, so there is more uniformity in the code being written. When the guidance is taken away, the code becomes more variable, and this is where the worldstates show a demonstrable improvement.

It is important to note that because of the way worldstates are defined, the ability to generate more hints is trivially guaranteed. The contribution is the *scale* at which the new hints are generated. For the same amount of student data in the interaction network, the percentage of hints available is anywhere from two to four times larger when using worldstates than codestates.

Figure 5 summarizes these results by averaging the hints available for the first four challenge puzzles. Challenge puzzles are chosen as they represent a puzzle being solved without any prompts on how to solve it, and would be the environment where these automatically generated hints are deployed in practice. Challenge 5 was only attempted by 34 students, so it was left out of the average.

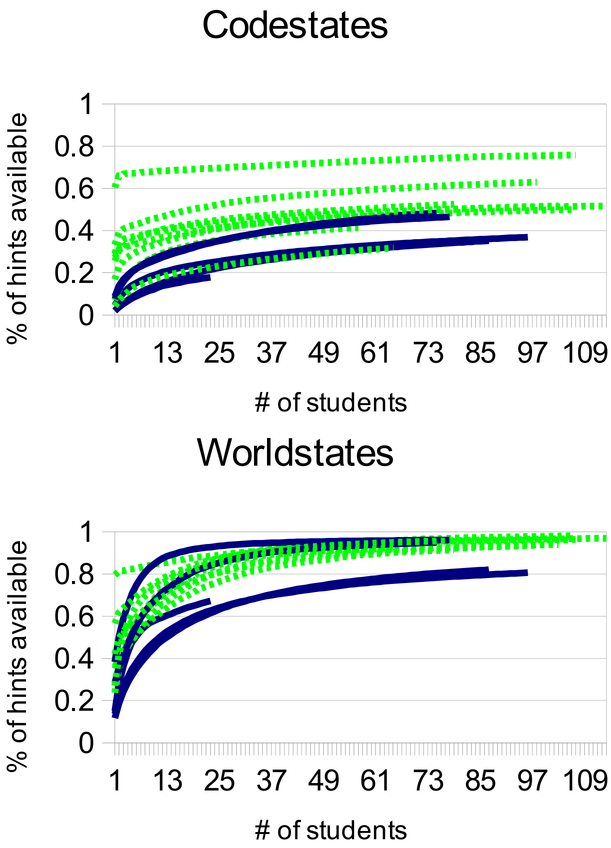


Figure 4: These graphs show the overall performance of hint generation for codestates and worldstates. The solid blue lines are the challenge puzzles, and the dotted light-green lines are the tutorial puzzles.

6.3 Validation

Table 3: This table shows the percentage of hints available using worldstates when 30 students worth of data are in the interaction network.

Name	Average	Median	Min	Max
Challenge 1	0.67	0.66	0.48	0.81
Challenge 2	0.67	0.66	0.44	0.84
Challenge 3	0.94	0.93	0.83	0.99
Challenge 4	0.88	0.87	0.69	0.96

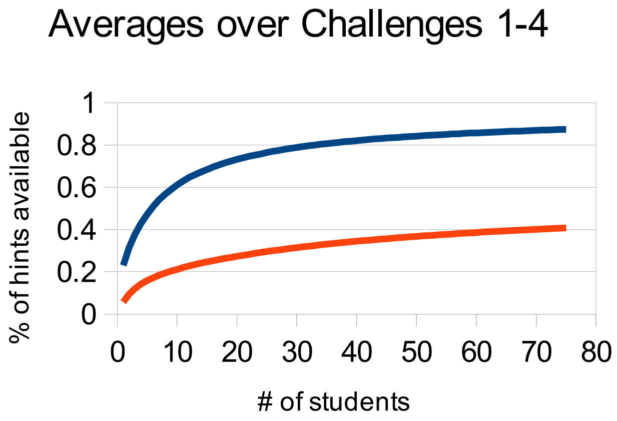


Figure 5: This graph summarizes Figure 4 by averaging the results of the analysis over the first four challenge puzzles. For these challenge levels (where students do not have guidance from the system) we are able to consistently generate hints with less ata when using world states rather than code states.

Figure 5, suggests that 30 students of data should be able to generate hints using worldstates about 80% of the time. To validate this hypothesis, we do cross-validation with training sets of 30 students and validate on the remaining students for challenge puzzles one through four, once again, because they are an appropriate model for how hints would be deployed in practice. We find the average, median, maximum, and minimum of the results over another 1000 trials, and summarize them in Table 3. We find that for challenges 1 and 2, hint generation is below the 80% mark, but for challenges 3 and 4, it is well over.

7. DISCUSSION

Our results indicate that when using worldstates we are able to generate hints using less data than when using code states. The reduction in the state space and the number of hints available after only a few dozen students solve the puzzle is highly encouraging. We only test our results on instructor-authored puzzles for this study, but these results potentially make hint generation for user-generated levels feasible as well.

While on average, the number of hints available using worldstates is very high, it is interesting to look at numbers on a per level basis. For example, in the summary in Table 3, there are less hints available after 30 students have been through the puzzle than the second two, which are presumably harder. This could be an averaging effect due to the more advanced students remaining in the more advanced levels, but there are some structural differences to the levels that may also have an effect. Figures 6 and 7 show the puzzles for Challenges 1 and 4.

Challenge 1 is a much smaller puzzle, but can be solved many different ways. Any of the blocks can be put on any of the buttons, and the robot can also hold down a button, meaning that there are several goal states, especially considering that the superflous block can be located anywhere. Challenge 4 is substantially more difficult, but has much less variation in how it can be completed. In this puzzle, the student has to place blocks so that the robot can scale the tower, and because of the way the tower is shaped, there are not nearly as many different ways to approach the problem as there are in Challenge 1.

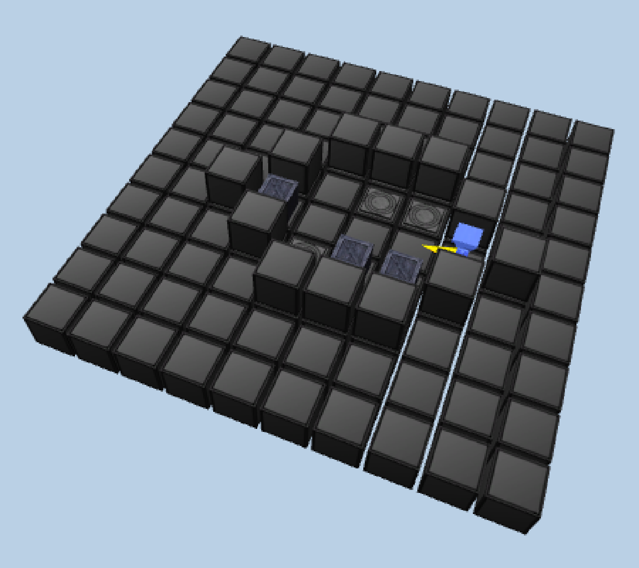


Figure 6: A screenshot of challenge 1. There are more blocks than necessary to press all the buttons, since the robot can press a button too.

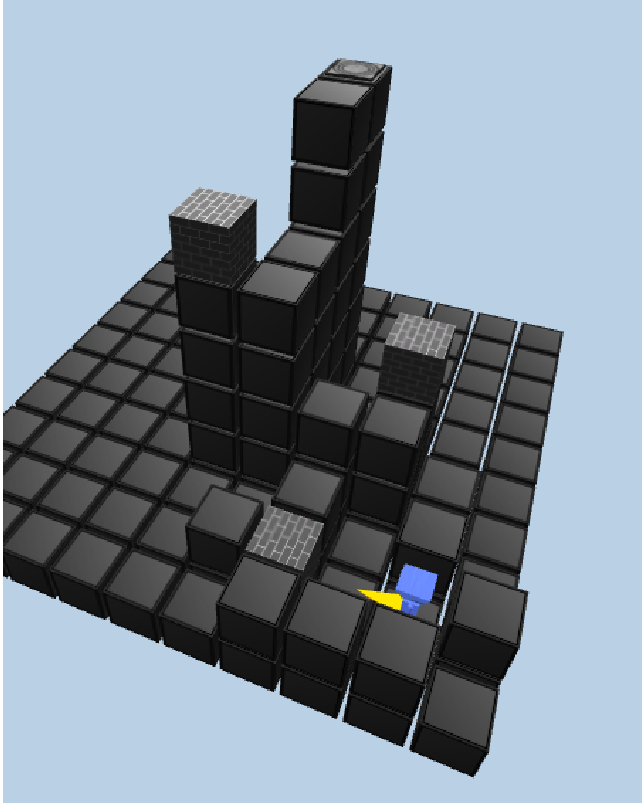


Figure 7: A screenshot of challenge 4. The puzzle is more complex, yet has a more linear solution.

The results for the earlier stages are more interesting for interpreting these results, because it shows how well worldstates manage the variability even in the wake of open-ended problems. While we emphasize the ability for worldstates to generate hints, it is important to note that codestates still have utility. A hint can be generated from any of the data in an interaction network, and using a world-based state representation does not restrict us from comparing other collected data. When enough data is collected for the codestates to match, more specific, low-level hints can be generated as well, meaning that more data does not just mean more hints, but also more detailed hints that can be applied at the source code level.

8. CONCLUSIONS AND FUTURE WORK

In this work, we describe our how we added intelligent tutoring system techniques to an educational game. Rather than using knowledge engineering, we instead use the approach used in Deep Thought, a logic tutor built around Hint Factory, where we provide hints to students by drawing from what worked for other students [1]. In order to deal with the fact that student code submissions can be highly diverse, with many different inputs resulting in the same output, we use the output of the student code to represent a student's position in the problem solving process. In doing so, we generate hints much more quickly than if we had only analyzed the source code alone.

In future work, we will identify the ability to generate hints on student-authored puzzles and test the effectiveness of these hints implemented in actual gameplay. We predict that by including hints, we can improve the completion rate of the tutorial and - if our results transfer to student-authored puzzles - improve performance on puzzles generated by other students. We will also explore how well these techniques transfer to contexts beyond our programming game.

9. ACKNOWLEDGEMENTS

Thanks to the additional developers who have worked on this project or helped with our outreach activities so far including Aaron Quidley, Veronica Catete, Trevor Brennan, Irena Rindos, Vincent Bugica, Victoria Cooper, Dustin Culler, Shaun Pickford, Antoine Campbell, and Javier Olaya. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 0900860 and Grant No. 1252376.

10. REFERENCES

[1] T. Barnes and J. C. Stamper. Automatic hint generation for logic proof tutoring using historical data. *Educational Technology & Society*, 13(1):3–12, 2010.

[2] M. Eagle, M. Johnson, and T. Barnes. Interaction networks: Generating high level hints based on network community clusterings. In *EDM*, pages 164–167, 2012.

[3] R. G. Farrell, J. R. Anderson, and B. J. Reiser. An interactive computer-based tutor for lisp. In *AAAI*, pages 106–109, 1984.

[4] D. Fossati, B. Di Eugenio, S. Ohlsson, C. W. Brown, L. Chen, and D. G. Cosejo. I learn from you, you learn

from me: How to make ilist learn from students. In *AIED*, pages 491–498, 2009.

[5] J. Huang, C. Piech, A. Nguyen, and L. Guibas. Syntactic and functional variability of a million code submissions in a machine learning mooc. In *AIED 2013 Workshops Proceedings Volume*, page 25, 2013.

[6] W. Jin, T. Barnes, J. Stamper, M. J. Eagle, M. W. Johnson, and L. Lehmann. Program representation for automatic hint generation for a data-driven novice programming tutor. In *Intelligent Tutoring Systems*, pages 304–309. Springer, 2012.

[7] B. J. Reiser, J. R. Anderson, and R. G. Farrell. Dynamic student modelling in an intelligent tutor for lisp programming. In *IJCAI*, pages 8–14, 1985.

[8] K. Rivers and K. R. Koedinger. Automatic generation of programming feedback: A data-driven approach. In *The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)*, page 50, 2013.

[9] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems Young Researchers Track*, pages 71–78, 2008.

Integrating Latent-Factor and Knowledge-Tracing Models to Predict Individual Differences in Learning

Mohammad M. Khajah, Rowan M. Wing, Robert V. Lindsey, Michael C. Mozer
Department of Computer Science and Institute of Cognitive Science
University of Colorado, Boulder, CO 80309-0430
{mohammad.khajah,rowan.wing,robert.lindsey,mozer}@colorado.edu

ABSTRACT

An effective tutor—human or digital—must determine what a student does and does not know. Inferring a student's knowledge state is challenging because behavioral observations (e.g., correct vs. incorrect problem solution) provide only weak evidence. Two classes of models have been proposed to address the challenge. Latent-factor models employ a collaborative filtering approach in which data from a population of students solving a population of problems is used to predict the performance of an individual student on a specific problem. Knowledge-tracing models exploit a student's sequence of problem-solving attempts to determine the point at which a skill is mastered. Although these two approaches are complementary, only preliminary, informal steps have been taken to integrate them. We propose a principled synthesis of the two approaches in a hierarchical Bayesian model that predicts student performance by integrating a theory of the temporal dynamics of learning with a theory of individual differences among students and problems. We present results from three data sets from the DataShop repository indicating that the integrated architecture outperforms either alone. We find significant predictive value in considering the difficulty of specific problems (within a skill), a source of information that has rarely been exploited.

Keywords

Bayesian knowledge tracing, cognitive modeling, collaborative filtering, latent factor models, hierarchical Bayesian models

1. INTRODUCTION

Intelligent tutoring systems (ITS) employ cognitive models to track and assess student knowledge. Beliefs about what a student knows and doesn't know allow an ITS to dynamically adapt its feedback and instruction to optimize the depth and efficiency of learning. A student's knowledge *state* can be described by the specific concepts and operations that have been mastered in the domain of study. These atomic elements are often referred to as *knowledge components* or *skills*. (We use the latter term.) For example, in a geometry curriculum, the *parallelogram-area* skill involves being able to compute the area of a parallelogram given the base and height [6]. Solving any problem typically requires breaking the problem into a series of *steps*, each requiring the application of one or more skills. For example, solving for x in $3(x+2) = 15$ might be broken down into two steps: (1) *eliminate-parentheses*, which transforms $3(x+2) = 15$ to $x+2 = 5$, and (2) *remove-constant*, which simplifies $x+2 = 5$ to $x = 3$ [14]. Because the terminology 'problem step' is cumbersome, we shorten it to 'problem' in the rest of this paper.

A key challenge in student modeling is predicting a student's success or failure on each problem. Following a common practice in the literature, we focus on modeling performance on individual skills. Formally, for a particular skill, the data consist of a set of binary random variables indicating the correctness of response on the i'th problem attempted by a student s, $\{X_{si}\}$. The data also include the problem labels, $\{Y_{si}\}$, which provide a unique index to each problem in the ITS. Recent work has considered secondary data, including the student's utilization of hints, response time, and characteristics of the specific problem and the student's particular history with the problem [2, 27]. Although such data improve predictions, the bulk of research in this area has focused on the primary success/failure data, and a sensible research strategy is to determine the best model based on the primary data, and then to determine how to incorporate secondary data.

2. EXISTING MODELS OF STUDENT LEARNING AND PERFORMANCE

The challenge inherent in predicting student performance is that knowledge state is a hidden variable and must be inferred from patterns of student behavior. Due to the intrinsic uncertainty associated with the inference problem, past approaches have been probabilistic in nature. Two broad classes of approaches have been explored, which we'll refer to as *latent-factor models* and *Bayesian knowledge tracing*, and some preliminary efforts have been made to synthesize the two. In this paper, we present a principled Bayesian unification of the two classes of models. We begin, however, with a summary of past work.

2.1 Latent-factor model

Traditional psychometric methods such as item-response theory [11] use data from a population of students solving a common set of problems to infer the latent ability of each student and the latent difficulty of each problem. These methods can be used to predict student performance. The simplest such model supposes that the log odds of a correct response by student s on trial i is given by logit$[P(X_{sy} = 1|Y_{si} = y)] = \alpha_s - \delta_y$, where, as before, Y_{si} denotes the problem index, α_s denotes the student's ability and δ_y denotes the problem's difficulty. We refer to this model class as *latent-factor models* or *LFMs*. The left panel of Figure 2 summarizes a Bayesian LFM in graphical model form, with priors on the abilities and difficulties (details to follow shortly), and with $G \equiv P(X_{sy} = 1|Y_{si} = y)$.

Latent-factor models have been used within the ITS community to characterize student performance and predict the consequences of instructional interventions. Examples include performance factors analysis [23], learning factors analysis [6, 5], and instructional factors analysis [8]. Although these models incorporate a wide range of factors, only a few papers have considered what has historically been at the core of latent-factor models, the difficulty of a specific problem. Consequently, a *remove-constant* problem step that simplifies $x + 1 = 3$ is typically considered to to be equivalent to problem step that simplifies $x + 8 = 11$.

2.2 Bayesian knowledge tracing

Bayesian knowledge tracing (*BKT*) [9] is based on a theory of all-or-none human learning [1], which postulates that the knowledge state of student s following trial i, K_{si}, is binary: 1 if the skill has been mastered, 0 otherwise. BKT, often conceptualized as a hidden Markov model, infers K_{si} from the sequence of observed responses on trials $1 \ldots i$, $\{X_{s1}, X_{s2}, \ldots, X_{si}\}$. Table 1 presents the model's four free parameters.

Because BKT is typically used in modeling practice over brief intervals, the model assumes no forgetting, i.e., K cannot transition from 1 to 0. This assumption greatly constrains the time-varying knowledge state: it must make at most one transition from $K = 0$ to $K = 1$ over the sequence of trials. Denoting the trial following which the transition is made as τ, the generative model specifies:

$$P(\tau = i) = \begin{cases} L_0 & \text{if } i = 0 \\ (1 - L_0)T(1 - T)^{i-1} & \text{if } i > 0 \end{cases}$$

$$P(X_{si} = 1|G, S, \tau) = \begin{cases} G & \text{if } i \le \tau \\ 1 - S & \text{otherwise} \end{cases}$$

The middle panel of Figure 2 shows a graphical model depiction of BKT with the knowledge-state transition sequence represented by τ. With this representation, marginalization over τ is linear in the number of trials, permitting the efficient computation of the posterior predictive distribution, $P(X_{s,i+1} \mid X_{s1}, \ldots, X_{si})$.

2.3 Prior efforts to unify latent-factor and knowledge-tracing models

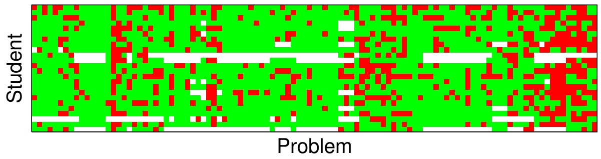


Figure 1: Student × problem matrix for the *Geometry Area* data, obtained from the PSLC DataShop [17]. Correct and incorrect responses are green and red, respectively; white indicates missing data. Students who attempted few problems have been omitted.

Latent-factor and knowledge-tracing models have complementary strengths and weaknesses. LFM addresses individual differences among students and problems. However, because it does not consider the order in which problems are solved, it ignores the likely possibility that performance improves over practice. BKT characterizes the temporal dynamics of learning. However, because it makes no distinction among students or problems, it ignores confounding factors on performance. A natural extension of the models is to formulate some type of combination that yields a more robust representation of knowledge state.

Interesting extensions have been proposed to each model to move it toward the other. Starting with LFM, the latent factors have been augmented with non-latent factors that represent facets of study history such as the amount and success of past practice and the type of instructional intervention [5, 6, 7, 8, 19, 23]. However, these approaches reduce the specific sequential ordering of problems to a few summary statistics, which may not be sufficient to encode the relevant history of past experience.

Many proposals have been put forth to adapt parameters of BKT to individual students. The original BKT paper [9] included heuristic parameter adjustments based on the initial trials in the problem sequence. Another heuristic approach involves the contextualization of guess and slip probabilities based on a range of features such as help requests, response time history, and ITS history [2, 10]. The initial mastery parameter L_0 has been individualized to students, based both on their performance on other skills [20] and on an inferred latent ability parameter [26]. Rather than adapting parameters to individual students, [22] clustered students based on their ITS usage patterns and fit separate parameters for each cluster. The latter two methods require previous history with a particular student, though placing Dirichlet priors on guess and slip rates [3, 4] has been used not only to individuate the parameters for a particular student but to allow for generalization to new students.

Most applications of BKT fit model parameters independently for each skill. There are only a few examples of modulating parameters based on the specific problem being solved. In [13], problem difficulty is represented by using the average number of correct responses on a problem as a feature in the contextualization model of [2]. In the KT-IDEM model [21]—the work closest to our own—the guess and slip parameters are fit individually for each problem within a skill.

Table 1: Free parameters of BKT

L_0	$P(K_{s0} = 1)$	probability that student has mastered skill prior to solving the first problem
T	$P(K_{s,i+1} = 1 \mid K_{si} = 0)$	transition probability from the not-mastered to mastered state
G	$P(X_{si} = 1 \mid K_{si} = 0)$	probability of correctly *guessing* the answer prior to skill mastery
S	$P(X_{si} = 0 \mid K_{si} = 1)$	probability of answering incorrectly due to a *slip* following skill mastery

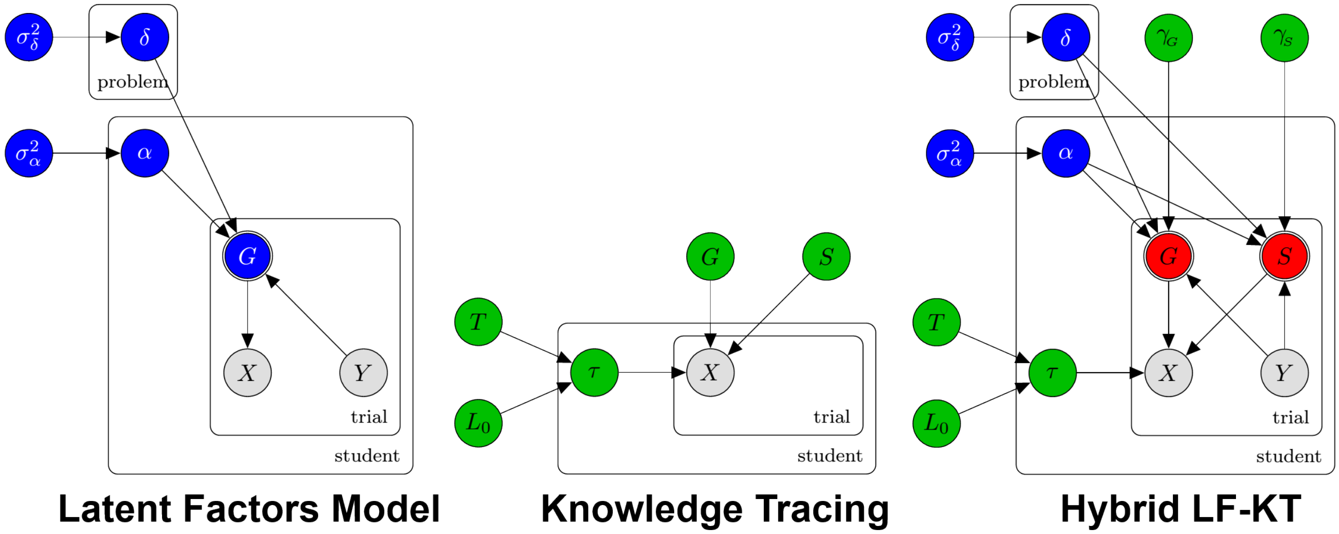


Figure 2: Graphical model depiction of the latent-factor model (left), Bayesian knowledge tracing (middle), and our hybrid LFKT model (right). Following standard notation, shaded nodes are observations, with X denoting the response of a student when problem Y is presented. Double circles denote deterministic nodes. A node's color represents the model that contributed the node with blue, green and red indicating LFM, BKT and LFKT nodes, respectively.

Figure 1 provides an intuition for the value of both student- and problem-specific factors influencing performance. The Figure shows a student × problem matrix, with a cell colored to indicate whether the student solved the problem. As variation in the columns indicate, some problems are more challenging than others. (However, because problem selection and order are partially confounded, one must be cautious in attributing the accuracy effects to intrinsic difficulty of the problem. Regardless of the source of the effect, the presence of the effect is indisputable.)

In the next section, we propose a synthesis of latent-factor and knowledge-tracing models. The synthesis is a natural extension and integration of past work. Indeed, the synthesis is so natural that another paper accepted at EDM 2014 also made this same proposal [12]. We address this highly related work in the discussion section at the end of this paper. A poster at EDM 2013 [26] also explicitly proposed combining latent-factor and knowledge-tracing models. However, their synthesis focused on individuating BKT's initial mastery probability, whereas our effort focuses on individuating guess and slip probabilities.

3. LFKT: A SYNTHESIS OF LATENT-FACTOR AND KNOWLEDGE-TRACING MODELS

In Figure 2, LFM and BKT are depicted in a manner that allows the two models to be superimposed to obtain a synthesis, which we'll refer to as LFKT, depicted in the rightmost panel of the Figure. LFKT personalizes the guess and slip probabilities based on student ability and problem difficulty:

$$\mathrm{logit}(G_{si}|Y_{is} = y) = \alpha_s - \delta_y + \gamma_G \quad \text{and}$$
$$\mathrm{logit}(S_{si}|Y_{is} = y) = \delta_y - \alpha_s + \gamma_S \ .$$

For simplicity, we assume that the effects of ability and difficulty are symmetric on guessing and slipping, though scaling parameters could be incorporated to permit asymmetry. Due to the offsets γ_G and γ_S, we can constrain the expectations $E[\alpha_s] = 0$ and $E[\delta_y] = 0$ with no loss of generality. Specifically, we assume $\alpha_s \sim \mathcal{N}(0, \sigma_\alpha^2)$ and $\delta_y \sim \mathcal{N}(0, \sigma_\delta^2)$, where σ_α^2 and σ_δ^2 are variances drawn from an Inverse-Gamma-distributed conjugate prior.

LFKT can be specialized to the LFM simply by fixing $T = 0$ and $L_0 = 0$. LFKT can be specialized to BKT at the limit of $\sigma_\alpha^2, \sigma_\delta^2 \rightarrow 0$.

Table 2: Dataset columns identifying students, problems, skills and correct responses

	Columns
Student	anonymous student ID
Problem	problem hierarchy + problem name + step name
Skill	problem hierarchy + knowledge component
Correct	first attempt

The LFKT model allows for the simultaneous determination of parameters of BKT and LFM. Alternative approaches might include training one model first, freezing its parameters, and then training the other model; or training the two models independently and then using them as an ensemble for prediction. However, simultaneous training allows each component to be informed by the other. Thus, by considering the difficulty of problems, the transition in the knowledge state may become sharper, and by considering the transition in the knowledge state, a better measure of problem difficulty and student ability may be obtained.

4. METHODOLOGY

4.1 Data and prediction task

Our simulation experiments were conducted using three corpora from the PSLC DataShop [17]: *Geometry Area (1996-97)*, from the Geometry Cognitive Tutor [16], *USNA Physics Fall 2006*, from the Andes Tutor [25] and *OLI Engineering Statics Fall 2011* [24]. The Geometry corpus contains 5,104 trials from 59 students on 18 skills, the Physics corpus contains 110,041 trials from 66 students on 652 skills and the Statics corpus contains 189,297 trials from 333 students on 156 skills. Each corpus was divided into skill-specific data sets consisting of the sequence of trials for each student involving problems that require a particular skill. In this paper, we refer to these sequences as student-skill sequences. If multiple skills are associated with a problem, we treat the combination of skills as one unique skill. Trial sequences had mean length 8.0 for Geometry, 4.5 for Physics and 6.0 for Statics.

For reference, Table 2 shows the dataset columns used to identify students, skills, problems and correct responses. The PSLC datashop exports datasets in a common format, which allows us to refer to the same column names for all datasets. The plus sign indicates that the contents of the columns are concatenated together. We attach the problem hierarchy to the skill column following the same practice in [22, 21]. Effectively, this breaks up trial sequences into shorter sequences, which alleviates the problem of students forgetting learned skills over a long time period.

To validate model implementation and parameter settings, we also explored a synthetic dataset obtained by running LFKT in generative mode with the same weak priors used for inference in real datasets. The synthetic dataset contains 50 students and 50 skills. Each skill contains 50 problems and a student may practice a skill for a maximum of 50 trials.

In the literature on student modeling, a variety of measures have been used to evaluate model performance. It seems common to train a model on the entire data set, and to use an AIC- or BIC-penalized measure of fit to estimate performance. We prefer the more conventional approach of partitioning a data set into training and test trials. One way to partition is between the early and late trials in each student's trial sequence. Using this partition, one can predict the future performance for a current student. Another way to partition is by placing some students in the training set and some in the test set. Using this partition, one can predict the performance of the model on previously unseen students. We conduct a separate set of simulation studies for each partitioning.

Model predictions, $\hat{P}$, were evaluated using the log likelihood of the complete test data, i.e.,

$$l = \sum_s \sum_{i=1}^{N_s} \ln \hat{P}(X_{si}|X_{s1}, \ldots, X_{s,i-1}),$$

which can be interepreted as a measure of sequential prediction accuracy for each test trial conditioned on preceding trials in the student-skill sequence.

4.2 Models and implementation

We conducted simulations using the three models in Figure 2—LFM, BKT, and LFKT—in addition to a baseline model. The baseline model gave a fixed prediction equal to the mean response accuracy on each skill in the training set, and was thus independent of trial, problem, and student. To get a better handle on the contribution of student abilities and problem difficulties to model performance, we also tested variants of LFKT that included only abilities or only difficulties. We refer to these variants as BKT+A and BKT+D, where BKT+AD is equivalent to LFKT.

Models containing student ability parameters (LFM, LFKT and BKT+A) were fitted across skills. Thus, a model may use the performance of a student on one skill to infer the student's performance on another skill. This contrasts with most work on modeling with BKT, where models are independently trained on each skill.

LFM and BKT were implemented as special cases of LFKT, in order to use the same code and algorithms for each model. Each model was evaluated in a two-phase process. In the first phase, using the training data $(\{X_{si}\}, \{Y_{si}\})$ and MCMC sampling, a set of posterior samples were obtained on the variables γ_G, γ_S, $\{\delta_y\}$, $\{\alpha_s\}$, L_0, and T. The conditional data likelihood for each student, $P(\mathbf{X}_s|\mathbf{G}_s, \mathbf{S}_s, L_0, T)$ was computed exactly, and therefore sampling of τ was not required. For the other variables, slice sampling was used for a total of 100 iterations after a burn-in of 10 iterations. (These small numbers were sufficient due to the efficiency of slice sampling.) In the second phase, the training samples were used to formulate predictions for the test set. Due to the conjugate prior on the α's, the posterior predictive distribution on test student ability could be determined analytically, i.e., $P(\alpha_{s'}|\{\alpha_s\})$, where s' indexes a test student, and $\{\alpha_s\}$ are the sampled abilities of the training students. A similar predictive distribution could be obtained for $\delta_{y'}$, the difficulty level for a problem y' found in the test set but not the training set, via $P(\delta_{y'}|\{\delta_y\})$.

Weak priors were specified for six variables in the LFKT model: $\gamma_G, \gamma_S \sim \text{Uniform}(-3, 3)$, $L_0, T \sim \text{Uniform}(0, 1)$, $\sigma_\alpha^2, \sigma_\delta^2 \sim \text{Inverse-Gamma}(1, 2)$.

5. EXPERIMENTS

We conducted two experiments to evaluate the models. The first experiment evaluates model performance on the final trials of current students whilst the second evaluates model performance on students held out from training in a particular skill. The two experimental setups are depicted in Figure 3 and are explained in the next two sections.

5.1 Experiment 1: Predicting Performance of Current Students

In this experiment, we ask the question: given the initial responses of a student practicing some skill, how well does the model predict performance on the remaining trials? To answer this question, we grouped trials by skill and student to obtain a list of student-skill sequences. The last 20% of trials from each sequence were placed in the testing set. This design ensures that the models do not have to generalize to new students.

The top row of Figure 4 shows the mean negative log likelihood on the test data. Each graph is for a different data set. Each bar represents the performance score for a given model, with the models arranged left-to-right from simplest to most complex, i.e., from fewest to most free parameters. Smaller scores indicate better performance. The results are consistent across the four data sets: (1) BKT outperforms the baseline model. BKT assumes the student can be in one of two knowledge states, whereas the baseline model assumes a single knowledge state (and a constant probability of correct response across trials for a given skill). (2) When BKT is modulated by latent student ability (BKT+A) or problem difficulty (BKT+D), it outperforms off-the-shelf BKT, with the possible exception of BKT+D in the Geometry data set. (3) LFKT, which incorporates both student abilities and problem difficulties, outperforms BKT as well as the variants that incorporate one latent factor or another. (4) LFKT also outperforms off-the-shelf LFM, indicating that the temporal dynamics of learning incorporated into BKT are helpful for prediction. Thus, we observe clear evidence that the combination of latent factors and knowledge tracing yields a model with greater predictive power than models that have one component or the other.

5.2 Experiment 2: Predicting Performance of New Students

In this experiment, we ask the question: Given a model trained on some students for a given skill, how well does it predict performance of a new student on that skill?

For this experiment, we chose a random subset of students to hold out from each skill. Fifty train/test splits were generated this way using 10 replications of 5-fold cross validation (with an 80%/20% data split). Results were averaged across the 50 test sets.

Test performance in Experiment 2 is shown in the second row of Figure 4, respectively. The pattern of results we observe is identical to that in Experiment 1, indicating that the superiority of LFKT over BKT and LFM does not depend on the specific manner of evaluating the model. (The error bars are somewhat smaller in Experiment 2 than in Experiment 1 due to the fact that the nature of the experiment allowed for more data to be included in the test set.)

We note that Experiment 2 is not purely student stratified because each student had data included in both the training and test sets, albeit for different skills. We conducted a third experiment in which the models were trained not on all skills simultaneously, but on one skill at a time. This training procedure ensures that the models are truly naive to a given student in the test set, which impacts the performance of BKT+A, LFM, and LFKT. Nonetheless, the training data still constrains the student ability distribution, and as a result, the pattern of results still shows LFKT outperforming LFM and BKT.

5.3 Visualization of the Posterior Marginals

One advantage of using a Bayesian modeling approach is that we obtain posterior distributions over model parameters, rather than just point estimates, which allows us to directly quantify model uncertainity about those parameters. In a Bayesian model, we can estimate the joint posterior distribution over the parameters conditioned on the training data. From the joint distribution, which is challenging to visualize, we can compute marginals for each parameter. The marginals are easier to interpret. Because we are using an MCMC sampler, we obtain multiple samples of each parameter setting. The estimated marginal posterior for a parameter is then just histogram of those samples.

To calculate the marginals, we trained LFKT on the entire statics dataset and obtained 1000 samples from the posterior. Figure 5 shows visualizations of the resulting marginal distributions for each parameter. The x-axis in each plot is an index over either students, problems, or skills and each vertical slice of a plot provides the probability distribution over the parameter's value. Probability density is indicated by the color. The targets on the x-axis are sorted by the mean value of the corresponding parameter.

The top two plots in Figure 5 give us a clue about why problem difficulties have a larger effect on prediction performance than student abilities for the Statics data set. The posterior on student abilities are smaller in magnitude than the posterior on problem difficulties. Hence, when abilities are removed (i.e., set to 0) in LFKT to obtain the BKT+D model, the model does not lose much during testing. The model appears to be more certain about student and problem parameters (top row of Figure 5) than skill parameters (the bottom two rows of Figure 5). This difference is reflected in the fact that LFM, which uses the student and problem parameters, outperforms BKT, which uses the skill-specific parameters.

5.4 Execution Time

Even though LFKT combines BKT and LFM, its execution time is longer than the sum of the execution times of the two component models. Under LFM, a modification to a problem's difficulty requires re-evaluating the likelihood of the trials involving that problem. However, modifying a problem's difficulty under LFKT requires re-evaluating the

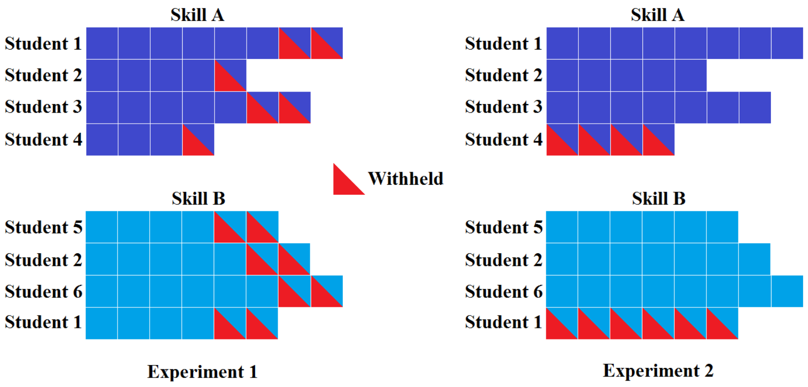


Figure 3: Data split for Experiments 1 and 2 (left and right columns, respectively). The squares represent individual trials and the red triangles represent trials withheld for testing. Squares with different colors belong to different skills.

Table 3: Execution times (seconds)

Dataset	BKT	LFM	LFKT
Synthetic	68.4	108.1	404.0
Geometry	8.0	2.0	14.5
Physics	211.3	412.4	712.4
Statics	175.3	81.0	865.2

likelihood of all the *student-skill sequences* that contain the problem. Table 3 presents the execution time in seconds for each model when training on the entire dataset. The runtime of LFKT is superadditive for all but the Physics data set, which is an anomaly because of (a) the large number of skills which results in short student-skill sequences and (b) the large number of problems which results in a sparse collection of student-skill sequences containing any particular problem. We note that we have made little effort to optimize run times, and alternative approaches (e.g., maximum likelihood parameter estimation) are likely to be significantly faster. Further, run time should not be nearly as important a consideration as model accuracy, so long as run times are tractable, which they clearly are in our simulations.

6. CONCLUSIONS

Within the intelligent tutoring community, there are two common approaches to modeling the performance of a student: Bayesian Knowledge Tracing (BKT) and Latent Factors Models (LFM). BKT is a two state model that attempts to characterize the temporal dynamics of student learning. LFM is a logistic regression model that infers latent factors associated with students, skills, and problems. Two approaches are complimentary, allowing us to synthesize the two into a single model. In this work, we presented LFKT, which integrates BKT and LFM in a mathematically principled manner, and we showed that the synthesis outperforms both BKT and LFM.

We investigated the contribution of individual components and factors within LFKT. Overall, our results indicate that the most important contribution to predicting performance comes from considering problem effects (difficulties), followed by student effects (abilities), followed by skill-specific learning effects (BKT). This ordering holds regardless of whether we are predicting performance on later trials of current students or on complete trial sequences of new students.

One important contribution of the work is the discovery that problem instances drawing on the same skill can systematically vary in difficulty, and inferring the latent difficulty of a problem and incorporating it in a predictive model can significantly bolster prediction accuracy. Although all problems that tax a given skill are equivalent in a formal sense, students are sensitive to the specific instantiation of the skill in a problem. We are aware of three variants of BKT that incorporate this useful fact. The KT-IDEM model [21] incorporates problem difficulties into BKT by fitting separate guess and slip probabilities for each a problem in a skill.

The FAST model [12] provides a general framework for characterizing guess and slip probabilities as a sigmoid function of a weighted linear combination of features. Given student and problem features, FAST discovers weights that are equivalent to the latent ability and difficulty factors in LFKT. However, in FAST, these factors are assumed to be independent for guess and slip probabilities. Thus, both KT-IDEM and FAST have *two* free parameters associated with problem difficulty, whereas LFKT has one one, which is assumed to be symmetric for guess and slip probabilities. This restriction may benefit LFKT in reducing overfitting. Another key difference is that both KT-IDEM and FAST are fit using maxmimum likelihood, whereas LFKT uses MCMC sampling to estimate Bayesian posteriors. The Bayesian approach allows LFKT to generalize to new problems and students in a principled manner. In a recent collaboration with the authors of FAST, we have performed a comparison of LFKT and FAST using the same datasets and evaluation metrics [15].

Another recent development that is complementary to LFKT is a variant of BKT in which the probability of initially knowing a skill (L_0) and the transition probability (T) are individualized to a student [28]. Individualization occurs by

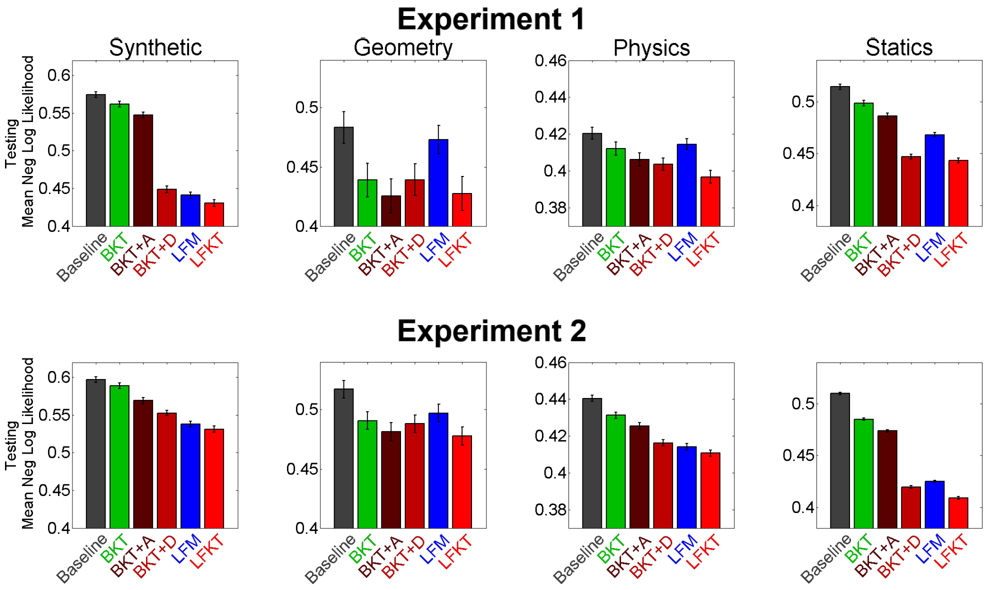


Figure 4: The mean *testing* performance on four data sets (columns) in Experiments 1 and 2 (top and bottom rows, respectively). Each graph shows the negative log likelihood score, averaged across trials, for each of six models. A lower value indicates better performance. BKT+A and BKT+D correspond to LFKT with difficulties set to zero or abilities set to zero, respectively. All trials are weighted equally across skills. Error bars indicate standard errors.

splitting each BKT parameter into skill-specific and student-specific components which are summed and passed through a logistic transform, yielding the BKT parameter value. Although this work mostly parallels ours but focusing on different BKT parameters, our discovery of problem-specific effects makes the intriguing suggestion that one might wish to consider problem difficulty on the transition probability; that is, the probability of learning a skill on a trial may be problem dependendent as well as success dependent.

By understanding the relationship between LFKT and other innovative variants of BKT, we are starting to delineate the space of models of student performance and the critical dimensions along which they vary. This understanding should lead to the emergence of a principled, unified theory that is sensitive to differences among individuals and differences due to the specific content. Such a theory should yield not only improved predictions of student performance but also more effective tutoring systems [18].

Acknowledgments

The research was supported by NSF grants SBE-0542013 and SMA-1041755 and an NSF Graduate Research Fellowship to RL.

7. REFERENCES

[1] R.C. Atkinson. Optimizing the learning of a second-language vocabulary. In *Journal of Experimental Psychology*, volume 96, pages 124–129, 1972.

[2] Ryan S. Baker, Albert T. Corbett, and Vincent Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In *Proceedings of the 9th international conference on Intelligent Tutoring Systems*, pages 406–415, Berlin, Heidelberg, 2008. Springer-Verlag.

[3] Joseph E. Beck. Difficulties in inferring student knowledge from observations (and why you should care). In *Proceedings of AIED2007 Workshop on Educational Data Mining (EDM'07)*, pages 21–30, 2007.

[4] Joseph E. Beck and Kai-min Chang. Identifiability: A fundamental problem of student modeling. In Cristina Conati, Kathleen F. McCoy, and Georgios Paliouras, editors, *User Modeling*, volume 4511 of *Lecture Notes in Computer Science*, pages 137–146. Springer, 2007.

[5] Hao Cen. *Generalized learning factors analysis: improving cognitive models with machine learning.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2009.

[6] Hao Cen, Kenneth R. Koedinger, and Brian Junker. Learning factors analysis - a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*, pages 164–175, 2006.

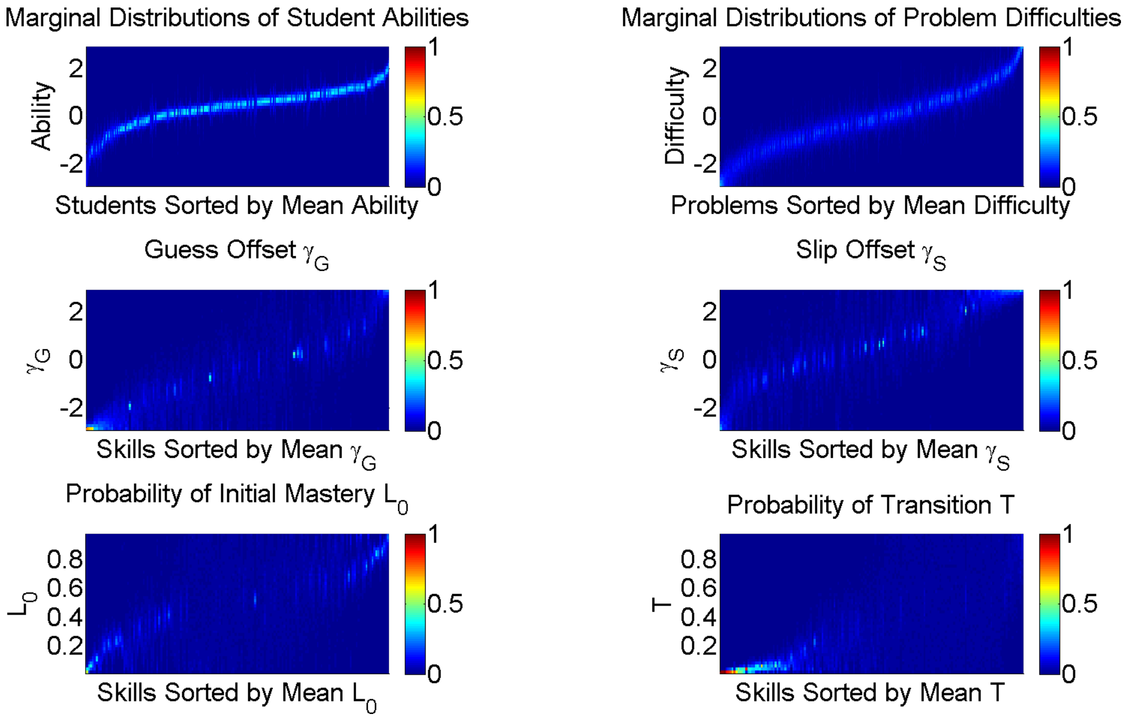
[7] Hao Cen, Kenneth R. Koedinger, and Brian Junker. Comparing two IRT models for conjunctive skills. In Beverly Park Woolf, Esma AÃŕmeur, Roger Nkambou, and Susanne P. Lajoie, editors, *Intelligent Tutoring Systems*, volume 5091 of *Lecture Notes in Computer Science*, pages 796–798. Springer, 2008.

[8] Min Chi, Kenneth R. Koedinger, Geoffrey J. Gordon, Pamela W. Jordan, and Kurt VanLehn. Instructional factors analysis: A cognitive model for multiple instructional interventions. In *EDM*, pages 61–70, 2011.

[9] Albert T. Corbett and John R. Anderson. Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.*, 4(4):253–278, 1995.

[10] Ryan Shaun Joazeiro de Baker, Albert T. Corbett, Sujith M. Gowda, Angela Z. Wagner, Benjamin A. MacLaren, Linda R. Kauffman, Aaron P. Mitchell, and Stephen Giguere. Contextual slip and prediction of student performance after use of an intelligent tutor. In *UMAP*, pages 52–63, 2010.

[11] P. De Boeck and M. Wilson. *Explanatory Item Response Models: a Generalized Linear and Nonlinear Approach.* Springer-Verlag, New York, NY, 2004.


Marginal Distributions of Student Abilities

Marginal Distributions of Problem Difficulties

Guess Offset γ_G

Slip Offset γ_S

Probability of Initial Mastery L_0

Probability of Transition T

Figure 5: Marginal posterior distributions for each parameter of the model. The x-axis represents the parameter's target (students, problems or skills), the y-axis is the value of the parameter and the color represents the probability of the parameter value for a particular target.

[12] J.P. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *To appear in Proceedings of the Seventh International Conference on Educational Data Mining*, 2014.

[13] Sujith M Gowda and Jonathan P Rowe. Improving models of slipping, guessing, and moment-by-moment learning with estimates of skill difficulty.

[14] KDD cup, 2010. Algebra 1 2005-2006 data set.

[15] Mohammad M Khajah, Yun Huang, José P González-Brenes, Michael C Mozer, and Peter Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *Personalization Approaches in Learning Environments*, 2014.

[16] Ken Koedinger. Geometry area (1996-97), February 2014.

[17] K.R. Koedinger, R.S.J.d. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data respository for the EDM community: The pslc datashop. In C. Romero, S. Ventura, M. Pechenizkiy, and R.S.J.d. Baker, editors, *Handbook of Educational Data Mining*, 2010. http://pslcdatashop.org.

[18] Jung In Lee and Emma Brunskill. The impact of individualizing student models on necessary practice opportunities. In *Educational Data Mining 2012*, pages 118–125. educationaldatamining.org, 2012.

[19] R.V. Lindsey, J.D. Shroyer, H. Pashler, and M.C. Mozer. Improving student's long-term knowledge retention with personalized review. *Psychological Science*, 25:639–47, 2014.

[20] Zachary A. Pardos and Neil T. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In Paul De Bra, Alfred Kobsa, and David N. Chin, editors, *UMAP*, volume 6075 of *Lecture Notes in Computer Science*, pages 255–266. Springer, 2010.

[21] Zachary A Pardos and Neil T Heffernan. KT-IDEM: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization*, pages 243–254. Springer, 2011.

[22] Zachary A. Pardos, Shubhendu Trivedi, Neil T. Heffernan, and Gábor N. Sárközy. Clustered knowledge tracing. In Stefano A. Cerri, William J. Clancey, Giorgos Papadourakis, and Kitty Panourgia, editors, *ITS*, volume 7315 of *Lecture Notes in Computer Science*, pages 405–410. Springer, 2012.

[23] Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 531–538, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.

[24] Paul Steif and Norman Bier. Oli engineering statics - fall 2011, February 2014.

[25] Kurt VanLehn. USNA physics fall 2006, February 2014.

[26] Y. Xu and J. Mostow. Using item response theory to refine knowledge tracing. In S. K. D'Mello, R. A. Calvo, and A. Olney, editors, *Proceedings of the Sixth International Conference on Educational Data Mining*, pages 356–7, 2013.

[27] Hsiang-Fu Yu and Others. Feature engineering and classifier ensemble for KDD cup 2010. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2010.

[28] Michael V Yudelson, Kenneth R Koedinger, and Geoffrey J Gordon. Individualized Bayesian knowledge tracing models. In *Artificial Intelligence in Education*, pages 171–180. Springer, 2013.

Interpreting Model Discovery and Testing Generalization to a New Dataset

Ran Liu
Psychology Department
Carnegie Mellon University
ranliu@cmu.edu

Kenneth R. Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
koedinger@cmu.edu

Elizabeth A. McLaughlin
Human-Computer Interaction Institute
Carnegie Mellon University
mimim@cs.cmu.edu

ABSTRACT

Automated techniques have proven useful for improving models of student learning even beyond the best human-generated models. There has been concern among the EDM community about whether small prediction improvements matter. We argue that they can be quite significant when they are interpretable and actionable, but the importance of generating meaningful, validated, and generalizable interpretations from machine-model discoveries has been under-emphasized in educational data mining. Here, we interpret a Learning Factors Analysis model discovery from a geometry dataset to suggest that students experienced difficulty applying the square root operation in circle-area backward problem steps. We then sought to validate and generalize this interpretation in the context of a completely novel dataset. Results indicated that our interpretation of the small, automated prediction improvement not only held up in the context of a novel dataset but also generalized to new types of problems that didn't exist in the original dataset. We argue that identifying cognitive interpretations of automated model discoveries and assessing the generalizability of such interpretations are critical to translating those model discoveries to concrete improvements in instructional design.

Keywords

Cognitive model discovery, model interpretation, generalization across datasets, learning factors analysis.

1. INTRODUCTION

Much Educational Data Mining (EDM) has focused on new data mining methods for improving within-dataset predictions. There has been interest in the community concerning whether small prediction improvements matter. Although we cannot provide a firm answer, we argue that they do when the improvements are interpretable and actionable. We have shown, in past experimental results, that genuine learning improvements can result from automated discoveries of small prediction differences [16]. Further, we argue that there should be more emphasis in EDM on whether predictions are clearly interpretable from a theoretical or cognitive perspective and whether the interpretation has external validity (e.g. generalizes beyond the dataset in which it was discovered).

Here, we present one of the first attempts at taking an interpretation of an automated cognitive model (or Q matrix [1, 8, 19]) discovery and generalizing that interpretation to a novel dataset, different from the one used to make the discovery. We focused on a discovery by the Learning Factors Analysis (LFA) algorithm [4] from a geometry dataset that improved predictions beyond the best available human-generated cognitive model. Even though the prediction improvement was small within this original dataset, with the addition of some exploratory data analysis, we interpreted the discovery within the context of a cognitive skill model [15].

Our intention was not to apply the improved model directly to new data (e.g., as in [11]) nor to run an exact replication of the study but, rather, to test whether the interpretation itself held up within the context of a new dataset with direct relevance to the interpretation but whose structure and properties may differ from those of the original dataset.

2. BACKGROUND

Cognitive models are an important basis for the instructional design of automated tutors and are important for accurate assessment of learning. Improvements to cognitive models, when combined with an appropriate theoretical interpretation, can yield better instruction and improved learning. More accurate skill diagnosis leads to better predictions of what a student knows, thus resulting in improved assessment and more efficient learning overall. Cognitive Task Analysis [5, 6, 17] is currently the best strategy for creating cognitive models of learning, but the method has its limitations. For example, it involves many subjective decisions and requires large amounts of human time and effort, as well as a high level of psychological expertise.

Educational data mining and machine learning techniques can be used to improve cognitive models in an automated fashion. These methods involve using data and statistical inference to create or modify a cognitive model involving continuous parameters over latent variables that can be linked to observed student performance variables. In addition to saving time and effort, machine models have the potential to discover cognitive model improvements that may not otherwise be considered via human-generated methods.

In order to use techniques of automated cognitive model improvement effectively towards the primary goal of bettering instructional design and assessment, it is important to properly interpret machine discoveries in the context of a cognitive skill model. Furthermore, it is critical to demonstrate the external validity of the interpretation beyond the dataset from which the discoveries were made. There exist good techniques (e.g., various methods of cross-validation) for ensuring internal validity of automated discoveries, but there have been few demonstrations of generalization beyond the samples in which discoveries are made.

Here, we discuss an example of an automated model discovery that improved a Knowledge Component (KC) Model, a specific type of cognitive skill model, beyond the best existing human-generated model. Knowledge Components represent units of knowledge, concepts, or skills that students need to solve problems. A KC Model is composed of a set of KCs mapped to a set of instructional tasks (e.g. problem steps). The LFA algorithm [4] automates the search process across hypothesized knowledge components (KCs) across a number of possible models. A tool such as the LFA algorithm not only reduces human effort and error by providing an automated method for discovering and evaluating cognitive models, but it outputs a most predictive Q matrix [19], thus producing a statistical version of a symbolic model. As such, LFA eases the burden of interpretation, but it does not in itself accomplish interpretation.

We applied the LFA search process across 11 datasets using different domains and technologies (available from DataShop at http://pslcdatashop.org; [13]). This automated process improved models, by cross-validation measures, across all of the datasets beyond the best manual models available [15]. However, the improvements in root mean square error (RMSE) were quite small. We questioned whether such miniscule changes in measurement are interpretable, generalizable, and—most importantly—actionable.

To investigate these questions, we focused on a particular dataset called Geometry Area 1996-1997, which is available to the public, has been analyzed for several other studies and shown to be reliable, and has produced findings we can test for generalization [12]. These data included 5,104 student steps completed by 59 students. Within this dataset, we compared the best LFA-discovered model (according to item-stratified cross validation) against two human generated models—the original model and the best hand-generated model (according to item-stratified cross validation). The LFA algorithm split circle-area problem steps into those that use a forward strategy (find area, given radius) and those that use a backward strategy (find radius, given area). It did not split any other area formulas for the backward-forward distinction. Thus, LFA essentially discovered an unforeseen "new" knowledge component (i.e., circle-area backward) for this dataset. As mentioned, the cross-validation results provided evidence of the internal validity of the discovered cognitive model improvement.

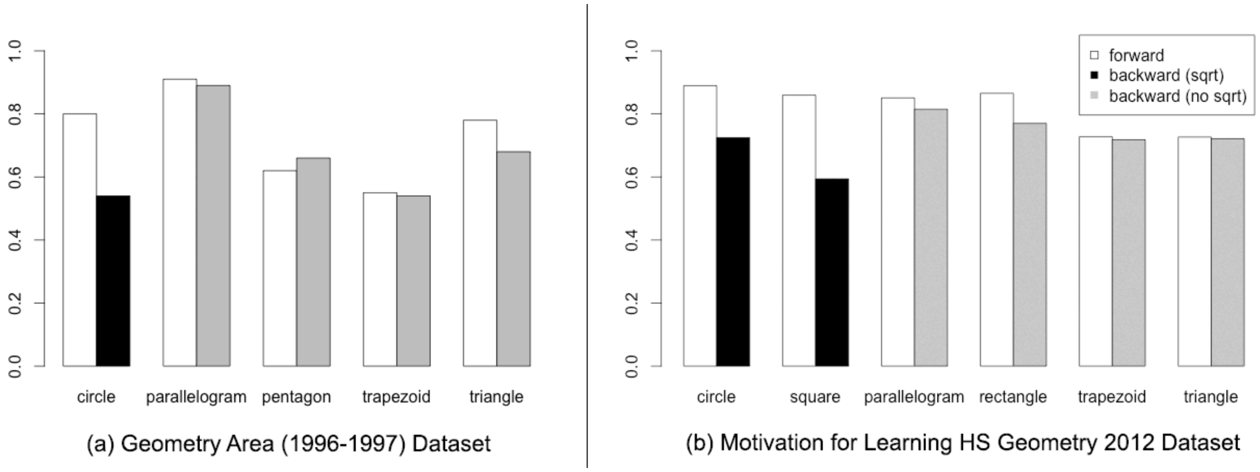
In the current paper, we aim to assess the external validity of this result in a novel dataset whose structure and properties are different from the original dataset in which the discovery was made. Since it was not possible to test the original LFA-discovered model directly on a new dataset due to its differing structure and problem types, it was critical that we generated a cognitive *interpretation* of the finding. The interpretation makes it possible to generate predictions and models that are appropriate to the novel dataset in which we aim to test the validity and generalization of our findings.

3. INTERPRETING MACHINE-DRIVEN MODEL IMPROVEMENTS

The LFA discovery within the Geometry Area 1996-1997 dataset yielded a result that we interpreted by combining information from the algorithm split and other relevant exploratory measures from the dataset itself. Analysis of the automated model revealed a forward-backward split only predictive for circle area (i.e., not for the other geometric shapes in the dataset nor for other circle formulas such as find diameter or radius given circumference). Data on student performance corroborated this finding. Circle-area backward problems were substantially more difficult for students than circle-area forward problems (54% vs. 80%), but performance on the other shapes exhibited small or negligible differences in forward vs. backward steps (Figure 1a). The circle-area split illustrates an important factor discovered by the LFA algorithm that had not been anticipated by human analysts.

Delving into the problem steps associated with circle-area backward computations revealed the necessity of a square root operation ($r = \sqrt{(A/\pi)}$) that was not a requirement in any of the other backward formulas. Given the unique feature of square root operation in the context of this dataset and the absence of a forward-backward model split or performance discrepancy on all other shapes' area calculations and all other circle formula calculations, we hypothesized that the automated model improvement was more about the difficulty knowing when and how to apply a square root operation than about the difficulty applying a backward strategy more generally.

Although data mining techniques helped discover the split, it took



(a) Geometry Area (1996-1997) Dataset

(b) Motivation for Learning HS Geometry 2012 Dataset

Figure 1. Average proportion correct on first attempts at geometry area problem steps, grouped by shape and color-coded based on whether the problem step requires a forward strategy, a backward strategy that requires a square root calculation, or a backward strategy that does not require a square root calculation. Panel (a) reflects the Geometry Area 1996-1997 dataset, where LFA discovered that merging forward and backward for all shapes but circle yielded the best predictions. Our interpretation was that this split reflected a difficulty applying (or knowing to apply) the square root, which only affects the circle-area backward computations. Based on this interpretation, we predicted a split between forward and backward problem steps for circles *and* squares but not other shapes. Panel (b) shows that performance in the Motivation for Learning HS Geometry 2012 dataset confirms this predicted split.

a rational cognitive analysis to identify an underlying cognitive process (e.g., square root operation) from the information obtained via the LFA output. To move from data analysis to data interpretation requires domain knowledge and cognitive psychology expertise beyond just methodological skills in EDM techniques.

4. VALIDATING AND GENERALIZING THE INTERPRETATION

Before using interpretations from machine-model discoveries to redesign instructional principles, it is often important to assess the external validity of the interpretations themselves. For example, the tutor unit for the Geometry Area 1996-1997 dataset had only three unique problem steps associated with the circle-area backward (i.e., find circle radius given area) calculation. Furthermore, it had no problem steps associated with a square-area backward (i.e., find square side length given area) calculation. Due to the limited task variety available in the Geometry Area 1996-1997 dataset, it remains unclear from that dataset alone whether our interpretation of difficulty applying the square root operation will generalize to data containing a broader set of tasks.

Thus, we sought to validate this interpretation of a machine-driven cognitive model discovery in an independent dataset containing substantially more circle-area backward problem steps as well as the existence of square-area backward problem steps, which were entirely absent from the original dataset. To this end, we investigated the geometry portion of a much more recent dataset, Motivation for Learning HS Geometry 2012 (geo-pa) [3]. This dataset is an excerpt from regular classroom use of a Geometry Cognitive Tutor [18] by 82 HS students (10th graders) with a total of 72,404 student steps. It contains similar types of shape-area modules and questions as the original dataset but has many more (49) unique circle-area backward problem steps. It also contains many (57) unique square-area backward problem steps. This makes it possible to validate (i.e. by investigating circle-area and other shape-area forward and backward performance) and generalize (i.e. by investigating square-area forward and backward performance) our interpretation of the original LFA-based discovery.

A first-pass exploratory analysis of the 2012 dataset reveals a substantially higher proportion of correct first attempts at forward, compared to backward, circle- and square-area problem steps (Figure 1b). In order to validate the specificity of the square root operation interpretation, we also investigated performance on backward vs. forward steps for all other shapes' area formulas. These data confirm that the performance differences between forward and backward area KCs are substantially smaller for the other shapes that don't require a square root operation in their backward steps (parallelogram backward=81%, forward=85%; rectangle backward=77%[1], forward=86%; trapezoid backward=72%, forward=73%; triangle backward=72%, forward=73%).

Beyond these performance data, we compared the performance of a KC model that aligns with our square root interpretation against KC models representing alternative hypotheses. Our hypothesis-driven KC model distinguishes backward-area steps from forward-area steps for circles and squares (since the backward steps require a square root operation) but does not make this forward-backward distinction for any other shapes. We compared this to a KC model that makes no forward-backward distinctions for any shapes (merges F-B across all shapes) as well as a KC model that makes all forward-backward distinctions for all shapes.

Since this dataset contained both circle-area and square-area problems, we also asked whether there might have been transfer between circle- and square-area backward problem steps on the basis that both require application of the square root operation. If there were full transfer, we would expect that a KC model merging square- and circle-area backward steps into a single skill should outperform a KC model that distinguishes square- from circle-area backward steps. To test this question of transfer, we created the former KC model and included it in our model comparison.

We compared performance across these four hypothesis-driven single-skilled[2] KC models:

1. SQRT SKILL CIR-SQ DISTINCT (58 KCs): Forward-backward steps coded as distinct for circle and square area problems; forward-backward steps merged (into a single "area" KC) for each of the other shapes. This KC model is structured based on our interpretation that backward steps requiring a square root operation should be coded as separate skills.
2. ALL SHAPES F-B MERGED (56 KCs): No forward-backward distinction for any shapes' areas (a single "area" KC is coded for each shape). This KC model is analogous to the original hand model for the Geometry Area 1996-1997 dataset from which LFA discovered the circle forward-backward area split on.
3. ALL SHAPES F-B DISTINCT (66 KCs): Forward-backward steps are coded as distinct[3] for all shapes' area problems. The comparison of our interpretation-based model (SQRT SKILL CIR-SQ DISTINCT) against this one is important for establishing the specificity of a square root operation hypothesis and rules out the possibility that the best split should, more generally, be forward vs. backward area steps across all shapes.
4. SQRT SKILL CIR-SQ BACKWARD (57 KCs): Forward steps coded as distinct for circle and square area problems; backward circle- and square-area steps merged into a single skill; forward-backward steps merged (into a single "area" KC) for each of the other shapes. The comparison of our interpretation-based model (SQRT SKILL CIR-SQ DISTINCT) against this one will inform us as to whether there was full transfer between backward circle- and square-area skills.

[1] Adjusted value reflecting the omission of 7 problem steps for which there was an error in the problem text. The pre-adjustment value is 0.70.

[2] The original KC model from which we constructed these four single-skilled models was a multi-skilled model. To convert the multi-skilled into a single-skilled model, we selected single skills corresponding with the LFA results on the Geometry Area 1996-1997 dataset.

[3] This model codes forward vs. backward steps with the finest-grain distinction possible: some shapes have multiple backward steps that are coded as distinct from each other (e.g., for parallelograms, "find height given area" and "find base given area" are coded as separate KCs).

Model Name	**KCs**	**AIC**	**BIC**	**RMSE:** ***Item-Stratified*** **Cross Validation (Average of 20 runs)**	**RMSE:** ***Student-Stratified*** **Cross Validation (Average of 20 runs)**
ALL SHAPES: F-B MERGED	56	20,992	22,652	0.28208	0.28702
ALL SHAPES: F-B DISTINCT	66	**20,839**	22,670	0.28104	*0.28588*
SQRT SKILL: CIR-SQ DISTINCT	58	20,857	**22,551**	**0.28087***	**0.28584**
SQRT SKILL: CIR-SQ BACKWARD	57	20,883	22,560	0.28113	0.28621

Table 1. Comparison between prediction accuracies of the four hypothesis-driven KC models, evaluated using AIC, BIC, and both item-stratified and student-stratified 10-fold cross validation (CV). Cross validation results are reported as the average root mean-square error (RMSE) values across twenty runs of 10-fold CV. The best performing model, by each of the measures, is bolded. *Significant at the p<0.001 level in t-tests comparing model performance against all other models, except the one italics entry, over the twenty runs of cross validation.

The models were evaluated using Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and 10-fold cross validation (CV). Due to the random nature of the folding process in cross validation, we repeated each type of 10-fold CV (item-stratified and student-stratified) 20 times and calculated the RMSE on each run, as has been done in previous work to handle this variabiltiy in CV [16].

In Table 1, we report the average root mean-square error (RMSE) values across 20 runs each of 10-fold item-stratified and 10-fold student-stratified CV. The SQRT SKILL: CIR-SQ DISTINCT model performs best, on average, by both item-stratified and student-stratified CV measures.

To ensure that it performed better than the next best model (ALL SHAPES: F-B DISTINCT) consistently, as opposed to by chance (due to random selection of folds), we compared the RMSEs from the 20 runs of item-stratified CV and student-stratified CV between the two models using a paired t-test. For *item-stratified* CV, the SQRT SKILL: CIR-SQ DISTINCT model had consistently lower RMSEs than the ALL SHAPES: F-B DISTINCT model across every one of the 20 runs, and this pattern was significant based on a paired t-test ($t = -10.249$, $df = 19$, $p < 0.0001$). For *student-stratified* CV, the SQRT SKILL: CIR-SQ DISTINCT model had lower RMSEs than the ALL SHAPES: F-B DISTINCT model on 14 of 20 runs, which was not statistically significant by a paired t-test.

Consistent with our previous work comparing machine-discovered models to baseline models [15], we focus on item-stratified cross validation as the primary metric, because we are concerned with improving cognitive tutors. Item stratified cross validation corresponds most closely with a key tutor decision of selecting the next problem type. Furthermore, the BIC measure concurs with the item-stratified cross validation results in suggesting that the SQRT SKILL CIR-SQ DISTINCT model is the best-performing model.

The superior performance of SQRT SKILL CIR-SQ DISTINCT over ALL SHAPES F-B MERGED (on all measures) supports and extends the original LFA finding that splitting F-B on circles and squares is better than leaving F-B merged. Notably, SQRT SKILL CIR-SQ DISTINCT even performs better, by item-stratified CV and BIC measures, than the ALL SHAPES F-B DISTINCT, the KC model that contains the same F-B distinctions for circle and square but even more fine-grained distinctions in the form of F-B separation for other shapes. This validates the specificity of the square root operation hypothesis and rules out the possibility that the major split should be for general forward vs. backward strategies among all shapes' area problems.

Thus, there is good evidence from KC model comparisons that distinguishing forward from backward steps specifically for circle- and square-area problems but not other shape-area problems predicts student learning best. This validates and generalizes our original interpretation that knowing when and how to apply the square root operation is the basis for the cognitive model improvements.

We did not observe full skill transfer between backward circle- and square-area steps, since the SQRT SKILL CIR-SQ BACKWARD model performed consistently worse than the SQRT SKILL CIR-SQ DISTINCT model by all measures. To investigate whether this may have been due to a lack of variability in the order that students complete circle-area backward vs. square-area problem steps, we examined the relative ordering of the two shapes' backward area steps. We discovered that each individual student completed all square-area backward steps before any circle-area backward steps. These data show a lack of variability in the relative ordering of the opportunities for the two shapes' backward-area practice, which suggest the combined model may only reflect partial transfer. This interpretation is supported by the observation that the end of the square-area backward learning curve (Figure 2, middle panel) does not align well with the beginning of the circle-area backward learning curve; rather, there is an increase in error rate (computed by taking the inverse logit of model values) from the end of square-area backward (22.2%) to the start of circle-area backward (47.3%).

We investigated learning curve prediction improvements yielded by our hypothesis-driven models (SQRT SKILL CIR-SQ

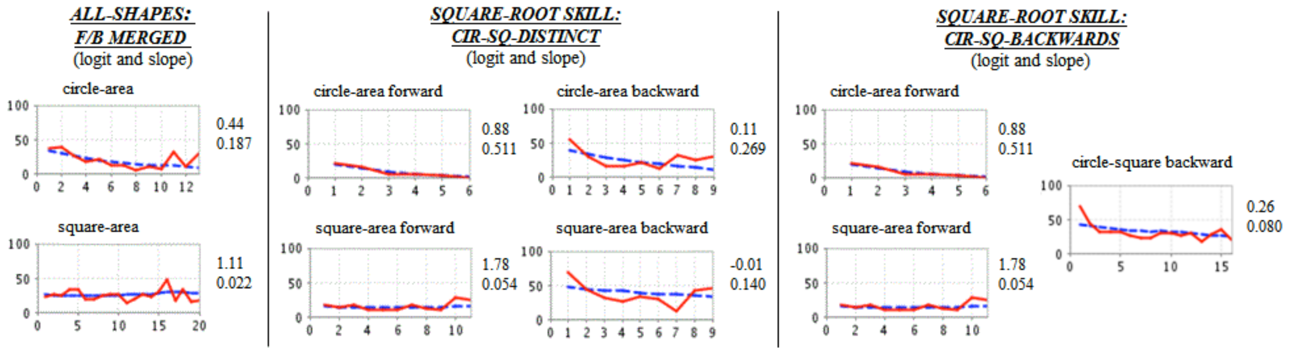


Figure 2. Learning curve prediction improvements (from the new 2012 dataset) yielded by comparing the square root KC models (middle and right panels) based on our interpretation of the LFA discovery against one that reflects what the KC model would have been (ALL-SHAPES: F-B MERGED, left panel) without the LFA discovery/interpretation. The x-axis reflects the opportunity number. Each data point was required to have at least 10 observations. The interpretation-based KC models that yielded better prediction results also exhibited higher learning slope values (bottom number to the right of each graph). This finding is consistent with what we observed using the LFA-discovered model in the original dataset.

DISTINCT and SQRT SKILL CIR-SQ BACKWARDS) compared to the baseline KC model (ALL SHAPES F-B MERGED). Figure 2 shows these learning curve predictions as well as their AFM model values (logit and slope). Our hypothesis-driven KC models, both of which consistently performed better than the baseline KC model, exhibit higher learning slope values. This finding is consistent with our general LFA results [15] that showed that models with better prediction results had higher learning slope values.

5. RELATED WORK

Beck & Xiong [2] rightfully raised concerns about the fact that many promising modeling approaches have produced only "negligible gains in accuracy, with differences in the thousandths place on RMSE." That paper focused on differences in statistical modeling approaches, such as Bayesian Knowledge Tracing and Performance Factors Assessment, whereas our focus is on cognitive model improvements. Beck & Xiong do make a similar comment about how cognitive model (they use the phrase "transfer model") modifications produce only "slight improvement in accuracy". In our case, we argue that even slight improvements can yield meaningful and valid interpretations that generalize to new contexts within the same domain and can be used to produce significant differences in student learning.

We completely agree with Beck & Xiong's suggestion that "higher predictive accuracy is not sufficient" and with their emphasis on interpretability, "is there any interpretable component relating to student knowledge?" We share a desire to connect results to student learning and address questions like "can we use this model to predict whether an intervention will lead to more learning?" However, we emphasize using interpretation of models not only to predict the impact of an intervention, but also as a *guide to design* such interventions. As we discuss below, cognitive model improvements, even ones with small impact on prediction accuracy, can be used to guide new instructional designs and high plausibility for impact in improving student learning. We need to "close the loop" and test whether designs based on cognitive model insights do improve learning, as has been done in past experiments [14, 16].

Learning Factors Analysis (LFA) requires human intervention to propose factors that may (or may not) account for task difficulty or transfer of learning from one task to another (e.g., backward application of a formula). This human intervention can be considered a downside of LFA relative to other cognitive model or q-matrix discovery algorithms [e.g., 1, 7-9, 19] that automatically produce new factors (e.g., as clusters of tasks with similar factor loadings). The results of these models, however, must be interpreted and post-hoc factor labeling is, in our experience, extremely difficult. It is quite hard to make sense of discovered factors or the task clusters they imply. We suspect that such interpretation difficulty is the reason that, to our knowledge, none of these methods have been used to produce new cognitive model explanations of task difficulty or transfer. More importantly, to our knowledge, none of them have been used to redesign instruction that can be tested in close-the-loop experiments. Thus, while LFA does require upfront human intervention to propose factors, this upfront investment appears to pay off in that LFA output affords more effective interpretation of model results on the backend.

At the other extreme, traditional methods of Cognitive Task Analysis such as structured interviews of experts [5, 6, 17] or think alouds [10] puts great emphasis on logical interpretation. They draw on qualitative data and are quite time-consuming or expensive to implement. LFA offers a quantitative alternative that may be easier to implement.

Other work besides ours has tested models produced using one dataset on another. For example, it was demonstrated that the structure and parameterization of a model using ASSISTment (www.assistments.org) system interaction data to predict state test scores in one year also works well in predicting state test scores from data in another year [11]. Here, we focused on transferring not only the specific structure of the model (e.g., the Q-matrix) but the cognitive insights from interpreting the model. The latter allowed us to make predictions on a kind of task (i.e., square-area backward) that was not even present in the original data or in the original Q-matrix. Making predictions of student performance on unseen tasks is something that a purely statistical model cannot do. We need to extend such models with logical or structural interpretations that have both explanatory power (i.e., they help us make sense of student learning) and generative power (i.e., they guide the design of better instruction).

6. CONCLUSIONS & FUTURE DIRECTIONS

Although the reduction in overall error (RMSE) was rather small in the original LFA model discovery on dataset Geometry Area 1996-1997, we demonstrated that the theoretical interpretation of this discovery was not only validated in an independent dataset but also generalized to new problem types that were not part of the original dataset (i.e., square-area backward). Error reductions can be small as a consequence of most of the model being essentially the same as the original but can still indicate a few isolated changes that are highly practically significant for tutor redesign. In a recent close-the-loop study [16], we demonstrated how using a cognitive model discovery to redesign a tutor unit led to both much more efficient and more effective learning than the original tutor. In that case, the discovered model had a statistically significantly lower RMSE on item-stratified cross validation (0.403) than the existing human-created model (0.406). The actionable interpretation of this small difference, only 0.003 in RMSE, was demonstrated to be practically important.

Some other automated techniques discover models that are difficult or impossible to understand (e.g., matrix factorization [7, 9]), either toward deriving insights into student learning or making practical improvements in instruction. The output of LFA is more interpretable and convertible to tutor changes than these alternative methods that may produce latent variable representations without the consistent application of human-derived codes or without code labels at all.

Here, we aimed specifically to assess the generalizability of our cognitive interpretation of an LFA model discovery. We showed that our interpretation held up within the context of a new dataset with domain relevance but whose structure and properties differed from those of the original dataset. Validation and generalization were confirmed, in the 2012 geometry dataset, based on (1) performance measures and (2) superior prediction of learning by a KC model constructed based specifically on our interpretation.

These findings move beyond simply replicating the original LFA model discovery. Since the novel dataset had a different structure from the original dataset, including differences relevant to our interpretation (i.e., existence of square-area backward problem steps), it would not have been viable to directly test the discovered automated model on this new dataset. Thus, the interpretation of automated model discoveries is actually *necessary* in order to test the generalizability of such discoveries across contexts with non-identical structures. Furthermore, interpretations help anchor all subsequent data exploration and analyses to something meaningful that can then be translated into concrete improvements to instructional design.

Testing the generalization of our interpretation not only confirmed the robustness of the idea but also yielded further details about the scope of the interpretation that have relevant implications for modifying instruction. For example, the original automated discovery may have suggested that we should treat circle-area backward problems as a separate skill, but the generalization of our interpretation suggests we should treat all backward area problems involving application of the square root operation—including square area—as distinct from their forward area counterparts. Further, the demonstrated validity of our interpretation has potential implications for instructional design beyond the cognitive tutors used to generate the datasets we worked with here. For example, the Khan Academy (www.khanacademy.org) geometry area units treat all circle-area problems as one skill and all square-area problems as one skill, with no forward-backward distinction, in their practice sets. Our findings suggest, at the very least, that it may be worth investigating whether our discovered interpretation also generalizes to student performance in very different instructional contexts such as that in the Khan Academy. If so, it would suggest potential instructional improvements there as well.

By isolating improvement in an interpretable component of student learning, elements of instructional design can be modified to more efficiently address student learning. An improved cognitive model can be used in multiple possible ways to redesign a tutor [16]. These include resequencing (positioning problems requiring fewer KCs before ones needing more), knowledge tracing (adding or deleting skill bars), creating new tasks, and adding/changing feedback or hint messages.

From the cognitive model improvement demonstrated here, we suggest adding new skills to the tutor that differentiate backward circle- and square-area problem steps from their forward counterparts. For other shapes, in contrast, we suggest that the skills for forward and backward area problem steps be merged. These skill bar changes would lead to changes in knowledge tracing as well as the creation of new tasks. In particular, students would receive increased practice on circle-area and square-area backward problems and decreased practice on some forward and backward steps for other shapes' area formulas. Finally, we suggest that new tasks or hint messages might be added to the backward circle- and square-area practice problems. For example, we might include additional questions, or hints, that simply ask "What do you need to do to 50 in x^2 = 50 to find the value of x?" We expect that the combination of increased practice on newly discovered skill difficulties and new tasks/hints that scaffold the difficulty would significantly improve overall student learning. In future work, we aim to "close the loop" on this finding by implementing these suggested instructional design changes and testing whether a redesigned tutor yields improvements in student learning above those achieved by the current tutor.

More generally, this work contributes to a broader set of evidence that a deep understanding of the cognitive processes of a domain through Cognitive Task Analysis (CTA) can lead to instructional designs that produce much better learning than typical instruction created through the self-reflections of a domain expert [5, 6, 17]. Prior work on CTA involves time-consuming expert interviews and subjective qualitative analysis. We find great promise in using data mining as a form of quantitative CTA that can more automatically and efficiently produce actionable discoveries. Nevertheless, the analysis process still involves human expertise in cognitive science to interpret model output and hypothesize cognitive interpretations that can be used to generalize across datasets and make effective instructional design decisions.

7. ACKNOWLEDGMENTS

Thanks to Carnegie Learning, Inc. for providing the Cognitive Tutor data supporting this analysis, to DataShop (http://pslcdatashop.org) for dataset access and to the National Science Foundation (#SBE-0836012) for funding of the Pittsburgh Science of Learning Center.

8. REFERENCES

[1] Barnes, T. (2005). The q-matrix method: Mining student response data for knowledge. In Proceedings of the American Association for Artificial Intelligence 2005 Educational Data Mining Workshop, pp. 1-8. Pittsburgh, PA.

[2] Beck, J. E., & Xiong, X. (2013). Limits to accuracy: How well can we do at student modeling. Proceedings of the 6th International Conference on Educational Data Mining.

[3] Bernacki, M., & Ritter, S. (2012). Motivation for Learning HS Geometry 2012 (geo-pa). Dataset 748 in DataShop. https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=748

[4] Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K.D. Ashley, T.-W. Chan (Eds.) Proceedings of the 8th International Conference on Intelligent Tutoring Systems, pp. 164-175. Berlin: Springer-Verlag.

[5] Clark, R. E. (2014). Cognitive Task Analysis for Expert-Based Instruction in Healthcare. In J. Michael Spector, J. M. Merrill, M. D. Elen, J. and Bishop, M. J. (eds.). Handbook of Research on Educational Communications and Technology, 4th Edition, pp. 541-551. Springer: New York.

[6] Clark, R. E., Feldon, D., van Merriënboer, J., Yates, K., & Early, S. (2008). Cognitive task analysis. In Spector, J. M., Merrill, M.D., van Merriënboer, J., & Driscoll, M.P. (Eds.) Handbook of research on educational communications and technology (3rd ed.). Mahwah: Lawrence Erlbaum.

[7] Desmarais, M. C. (2011). Mapping question items to skills with non-negative matrix factorization. ACM KDD-Explorations, 13(2), pp. 30-36.

[8] Desmarais, M. C., Behzad B., and Naceur, R. (2012). Item to skills mapping: deriving a conjunctive q-matrix from data. In Intelligent Tutoring Systems, pp. 454-463. Springer: Berlin-Heidelberg.

[9] Desmarais, M. C. and Naceur, R. (2013). A Matrix Factorization Method for Mapping Items to Skills and for Enhancing Expert-Based Q-matrices. In Proceedings of the 16th Conference on Artificial Intelligence in Education (AIED2013), pp. 441-450. Memphis, TN.

[10] Ericsson, K. A., & Simon, H. A. (1984). Verbal reports as data. Cambridge, MA: MIT Press.

[11] Feng, M., Heffernan, N. T., & Koedinger, K. R. (2009). Addressing the assessment challenge in an online system that tutors as it assesses. User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI), 19(3), pp. 243-266.

[12] Koedinger, K. R. (2006). Geometry Area 1996-1997. Dataset 76 in DataShop. https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=76

[13] Koedinger, K. R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. (2010). A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press.

[14] Koedinger, K. R. & McLaughlin, E. A. (2010). Seeing language learning inside the math: Cognitive analysis yields transfer. In: Ohlsson, S., Catrambone, R. (eds.) Proceedings of the 32nd Annual Conference of the Cognitive Science Society, pp. 471–476. Austin, TX.

[15] Koedinger, K. R., McLaughlin, E. A., & Stamper, J. C. (2012). Automated cognitive model improvement. Yacef, K., Zaïane, O., Hershkovitz, H., Yudelson, M., and Stamper, J. (eds.) Proceedings of the 5th International Conference on Educational Data Mining, pp. 17-24. Chania, Greece.

[16] Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. (2013). Using data-driven discovery of better cognitive models to improve student learning. Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED2013).

[17] Lee, R. L. (2003). Cognitive task analysis: A meta-analysis of comparative studies. Unpublished doctoral dissertation, University of Southern California, Los Angeles.

[18] Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. (2007). The Cognitive Tutor: Applied research in mathematics education. Psychonomics Bulletin & Review, 14(2), pp. 249-255.

[19] Tatsuoka, K.K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. Journal of Educational Measurement, 20, 345-354.

Learning Individual Behavior in an Educational Game: A Data-Driven Approach

Seong Jae Lee, Yun-En Liu, and Zoran Popović
Certer for Game Science, Computer Science and Engineering
University of Washington
{seongjae, yunliu, zoran}@cs.washington.edu

ABSTRACT

In recent years, open-ended interactive educational tools such as games have been gained popularity due to their ability to make learning more enjoyable and engaging. Modeling and predicting individual behavior in such interactive environments is crucial to better understand the learning process and improve the tools in the future. A model-based approach is a standard way to learn student behavior in highly-structured systems such as intelligent tutors. However, defining such a model relies on expert domain knowledge. The same approach is often extremely difficult in educational games because open-ended nature of these systems creates an enormous space of actions. To ease this burden, we propose a data-driven approach to learn individual behavior given a user's interaction history. This model does not heavily rely on expert domain knowledge. We use our framework to predict player movements in two educational puzzle games, demonstrating that our behavior model performs significantly better than a baseline on both games. This indicates that our framework can generalize without requiring extensive expert knowledge specific to each domain. Finally, we show that the learned model can give new insights into understanding player behavior.

Keywords

Educational Games, Data-driven Learning, Supervised Learning, Logistic Regression, Learning from Demonstration

1. INTRODUCTION

Open-ended educational environments, especially educational games and game-based learning, have been gaining popularity as more and more evidence suggests that they can help enhance student learning while making the process enjoyable [21, 19]. The interactive nature of this media enables us to track user inputs and provide an immediate personalized response to make learning more effective and efficient. Therefore, modeling and predicting user behavior lies at the heart of improving engagement and mastery for every learner. For example, we can detect if a student is struggling with a certain concept by simulating the learned student behavior model on a related task; if the student model performs poorly, we can then give more tasks related to that concept. The ability to further predict which error the student is going to make can be used to infer how the learner misunderstood related concepts and provide a targeted instruction focusing on that specific misconception.

There has been active research on learning individual behavior in both the education and game community. Most of this research focuses on inferring a meaningful latent structure, such as knowledge or skill, often simplifying the behavior space into a small number of parameters. Hence the choice of a model significantly affects the quality of the learned behavior, forcing researchers to spend time on experimenting with different models and refining various features [6, 20]. Moreover, unlike highly-structured systems such as intelligent tutors, it is difficult to define a behavioral model describing movements in an educational game. This is especially true when the player is given a large number of available moves, resulting in a large scale multi-class prediction problem. For example, predicting the exact moves someone will make while solving a puzzle is more challenging than predicting whether a student will solve a problem correctly.

Therefore, a purely data-driven approach is both a suitable and preferable alternative for learning user behavior in highly open-ended environment, such as educational games. It needs less expert authoring, and it can capture various low-level user movements—mistakes and errors, exploration habits, and adapting a strategy while playing—as they are even without a specific model describing such moves. Also, we can further analyze learned policies to give more interpretable insights into user behavior. For example, we could analyze erroneous movements to figure out their misconceptions. This knowledge can be used to construct more sophisticated cognitive models that will give us a deeper and more accurate understanding of user behavior.

In this paper, we propose a data-driven framework that learns individual movements in a sequential decision-making process. It uses a supervised classification method to predict the next movement of a user based on past gameplay data. For each game state, our framework transforms user play logs into a high-dimensional feature vector, and learns a classifier that predicts the next movement based on this feature vector. To construct this set of features, we start from a massive set of default features defined in a domain-independent way over a state-action graph, and then pick a small set of relevant ones using a univariate feature selection technique. We apply this framework to two very different educational games DragonBox Adaptive and Refraction, and evaluate the learned behavior by predicting the actions of held-out users.

Our contribution is threefold. First, we propose a data-driven individual behavior learning framework, which does not rely on heavy domain-dependent expert authoring. Second, we apply our framework on two different games and demonstrate our framework improves the prediction quality substantially over a previously proposed algorithm. Finally, combined with a robust feature selection, we show our framework learns an efficient yet powerful set of features, which further gives new insight into understanding player behavior in the game.

2. RELATED WORK

Learning user behavior has been actively researched in the education and game community. One of the most widely used models is the Bayesian network model. Knowledge tracing (KT) [9] and its variations [17, 27] are probably the most widely used student models in the field of educational data mining. This model estimates the user's knowledge as latent variables, or knowledge components (KCs). These KCs represent student mastery over each concept and predict whether a student will be able to solve a task. There are also other approaches using Bayesian networks, such as predicting whether a student can solve a problem correctly without requesting help in an interactive learning environment [16], predicting a user's next action and goal in a multiuser dungeon adventure game [5], or predicting a build tree in a real-time strategy game [22]. Nevertheless, these approaches usually learn the knowledge of users directly from carefully designed network structures, which often need expert authoring to define a task-specific and system-specific network model.

Another widely used approach to learn individual behavior is using factor analysis. Item response theory (IRT) models have been extensively used in psychometrics and educational testing domains [11, 8]. They represent an individual's score as a function of two latent factors: individual skill and item difficulty. The learned factors can be used to predict user performance on different items. Adapting matrix factorization techniques from recommender systems is also popular. Thai-Nghe et al. predicted personalized student performance in algebra problems by modeling user-item interactions as inner products of user and item feature vectors [24]. This model can even incorporate temporal behavior by including time factors, both in the educational community [25] and in the game community [28]. Nevertheless, such approaches do not easily fit our goal, which requires predicting a fine granularity of action instead of predicting user's single valued performance.

Another line of research on learning a low-level user policy is optimizing a hidden reward or heuristic function from user trajectories in a state-action graph. Tastan and Sukthankar built a human-like opponent from experts' demonstrations in a first-person shooter game using inverse reinforcement learning in a Markov decision process context [23]. Jansen et al. learned a personalized chess strategy from individual play records by learning the weights of multiple heuristics on finite depth search models [12]. Similarly, Liu et al. learned player moves in an educational puzzle game by learning the weights of heuristics in a one-depth probabilistic search model [15]. However, such methods usually define a user reward as a combination of pre-defined heuristics. The quality of the learned policy is strongly dependent on these heuristics, which are often system-specific and need a lot of time to refine. Furthermore, these models assume that players do not change over time. Describing how people learn, which is frequently observed in interactive environments, would be another challenge to trying this approach in our domain.

Unlike the research listed above, our framework focuses on a data-driven approach with less system-specific authoring. Our work is a partial extension of that of Liu et al. [15], which is a mixture of a one-depth heuristic search model and a data-driven Markov model with no knowledge of individual history other than the current game state. We focus on the latter data-driven approach and build upon that model. Unlike their work, which makes the same prediction for all players in a state, we build a personalized policy that considers the full history of the user.

3. PROBLEM DEFINITION

Our framework works on a model that consists of a finite set of states S; a finite set of actions A, representing the different ways a user can interact with the game; and rules of transitioning between states based on an action. This paper considers domains with determinstic transitions ($f : S \times A \rightarrow S$), but this approach could also apply to domains with stochastic transitions ($f : S \times A \rightarrow \Pr(S)$). The demonstration of a user u, or a trajectory τ_u, is defined as a sequence of state-action pairs: $\tau_u = \{(s_u^0, a_u^0), \cdots, (s_u^{t_u}, a_u^{t_u})\}$. We will note $A_s \subseteq A$ as a set of valid actions on the given state, T as a set of trajectories, V and U as the set of users in the training set and test set, respectively.

Defining such a model is often intuitive in games, because actions and transition rules are already defined in game mechanics. For example, in blackjack, the configurations of the visible cards can be states, and available player decisions such as hit or stand will be actions. The transition function here will be stochastic, because we do not know which card will appear next. Defining a good state space remains an open problem. A good, compact state representation will capture the information most relevent to user behavior; this helps greatly reduce the size of the required training data. For the example above, using the sum of card scores for each side would be a better state representation than the individual cards on the table. Note that a state refers to the observable state of the game, not the state of the player. In many games, players base their decisions not just on what they currently see in front of them, but also on past experience. This non-Markovian property of human players motivates our framework, which leverages a user's past behavior to improve prediction quality.

Our objective is to learn a stochastic policy $\pi : S \times T \rightarrow \Pr(A_s)$ describing what action a user will take on a certain

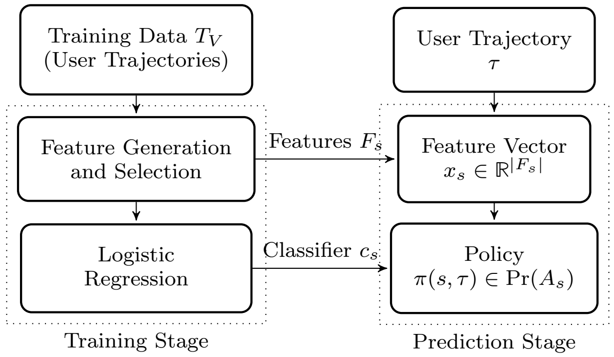

Training Data T_V (User Trajectories)
User Trajectory τ
Feature Generation and Selection
Features F_s
Feature Vector $x_s \in \mathbb{R}^{|F_s|}$
Logistic Regression
Classifier c_s
Policy $\pi(s,\tau) \in \Pr(A_s)$
Training Stage
Prediction Stage

Figure 1: An overview of our framework. For each state, we learn a set of features and a classifier in the training stage. With these, a trajectory is converted into a feature vector, and further into a stochastic policy in the prediction stage.

state based on his trajectory. We use a supervised learning approach, which attempts to predict the next action given a dataset of thousands of user trajectories in the training set $T_V = \{\tau_v | v \in V\}$.

4. EVALUATION

We evaluate the learned policy on every movement of each user trajectories in the test set $T_U = \{\tau_u | u \in U\}$. We use two evaluation metrics: log-likelihood and accuracy rate.

The log-likelihood is defined as

$$\sum_{u\in U} \sum_{t\in[0,t_u]} \log(\tilde{\pi}(a_u^t | s_u^t, \tau_u^t)),$$

while $\tilde{\pi}$ is the learned policy given a trajectory observed so far $\tau_u^t = \{(s_u^0, a_u^0), \cdots, (s_u^{t-1}, a_u^{t-1})\}$. Since the log function is undefined for the case $\pi(a|s,\tau) = 0$, we smooth the policy by ϵ: $\pi(a|s,\tau) = (1-\epsilon)\pi(a|s,\tau) + \epsilon/|A_s|$, while ϵ is set to 0.001 in our experiments unless otherwise specified. The log-likelihood is always smaller than or equal to zero, with 0 indicating perfect prediction.

With our stochastic policy, the accuracy rate is defined as

$$\frac{\sum_{u\in U} \sum_{t\in[0,t_u]} \tilde{\pi}(a_u^t | s_u^t, \tau_u^t)}{\sum_{u\in U} \sum_{t\in[0,t_u]} 1}.$$

We want to note that even though the accuracy rate is intuitive and widely used for measuring the performance of a classifier, using it as a single evaluation metric can be misleading, especially when outputs are highly skewed [13]. For example, a degenerate constant classifier that predicts only the dominant class may produce a high accuracy rate. Nevertheless, in all of our experiments, the ordering of performances in log-likelihoods is preserved with that in accuracy rates.

5. ALGORITHM

Figure 1 provides an overview of our data-driven framework. In the training stage, our framework learns a set of features and a classifier from training data. A simple way to do this might be to train a global classifier that takes a player trajectory and predicts an action. However, we suspect that the current state is the most important feature; in fact, the set of available actions A_s differs per state. Thus we train a separate classifier for each state, reducing the amount of data in the training set but increasing the relevancy. In the prediction stage, we take a trajectory and convert it into a feature vector using the learned features. Then we convert the feature vector into a policy from that state, using the learned classifier. Here, a feature is a function that takes a trajectory and returns a value $f : T \to \mathbb{R}$; a set of features converts a trajectory into a multi-dimensional feature vector $F : T \to \mathbb{R}^{|F|}$.

Algorithm 1 Training Stage

```
Require: a state $s$, training data $T_V$
$F_s \leftarrow$ a default set of features defined on $s$
$T_s, y_s \leftarrow$ a list of $\tau_v^{t-1}$ and $a_v^t$ such that $s = s_v^t\ \forall v, t$
$X_s \leftarrow F_s(T_s)$
$F_s \leftarrow$ SelectFeatures($F_s, X_s, y_s$)
if $F_s = \emptyset$ then
    $c_s \leftarrow$ LearnMarkovClassifier($y_s$)
else
    $X_s \leftarrow F_s(T_s)$
    $c_s \leftarrow$ LearnClassifier($X_s$, $y_s$)
end if
return $F_s, c_s$
```

Algorithm 1 describes a detailed process in the training stage. The SelectFeatures function takes a set of features, a set of feature vectors, and a set of performed actions as inputs and filters irrelevant features out. The LearnMarkovClassifier function takes a set of performed actions and returns a static classifier. The LearnClassifier function takes a set of feature vectors and performed actions as inputs and returns a classifier. We will explain each function in detail.

The training stage starts from a default set of features defined on a state-action graph. We use three kinds of binary features:

- whether the user has visited a certain state s: $\mathbb{1}[s \in \tau]$,
- whether the trajectory contains a certain state-action pair (s,a): $\mathbb{1}[(s,a) \in \tau]$, and
- whether the dth recent move is a certain state-action pair (s,a): $\mathbb{1}[(s,a) = (s^{|\tau|-d}, a^{|\tau|-d}) \in \tau]$.

The maximum number of d is set to 10 in our experiments. The features with sparsely-visited states and transitions are not counted. These features summarize which states and actions have been visited by the user, both in the full and recent history. We built a feature package defined on an abstract state-action graph so that it can be used generally in multiple systems without extra authoring.

The default set of features contains a huge amount of irrelevant features for the task, making our learning suffer from overfitting as well as prolonged training time. To remedy this, we apply a feature selection method using training data. For the SelectFeatures function, we use a chi-squared statistic for each feature on T_v and select features

State s	Action a	Next State $s' = T(s, a)$	$\pi(a\|s, \tau)$
`lv.3.4.1.x+a=b`	Subtract `a` on both sides with merging	`lv3.4.1.x+0=b-a`	0.4
	Subtract `a` on both sides	`lv3.4.1.x+a-a=b-a`	0.4
	Add `a` on both sides	`lv3.4.1.x+a+a=b+a`	0.2

Table 1: Examples of states, actions, transitions, and policies in DragonBox Adaptive

with p-value lower than 0.001. We used a univariate feature filtering approach because it is relatively fast compared to other feature selection techniques, such as L1-regularization. Also, we used a chi-squared test because it is one of the most effective methods of feature selection for classification [26].

Finally, we learn a classifier with the selected features. If any features are selected, we learn a supervised learning classifier using the LearnClassifier. We used a multi-class logistic regressor as a classifier, because it gives a natural probabilistic interpretation unlike decision trees or support vector machines. If no features are selected, we use a predictor built on a Markov model from Liu et al. [15], which learns the observed frequency of state-action pairs in training data:

$$\tilde{\pi}(a|s) = \frac{\sum_{v \in V} \sum_{t \in [0, t_v]} \mathbb{1}[(s, a) = (s_v^t, a_v^t)]}{\sum_{v \in V} \sum_{t \in [0, t_v]} \mathbb{1}[s = s_v^t]}.$$

When there are zero samples, i.e., when the denominator is zero, this equation is undefined and it returns a uniform distribution instead. For convenience, we will call this predictor the Markov predictor.

We can also interpret the Markov predictor as another logistic regressor with no features but only with an intercept for each action. With no regularization or features, logistic regression optimizes the log-likelihood on training data with a policy only dependant on the current state, whose optimal solution is the observed frequency for each class. This is exactly what the Markov predictor does.

In the prediction stage, we take a trajectory and a state as an input, and first check the type of the learned classifier on the given state. If the classifier is the Markov predictor, it does not need a feature vector and returns a policy only dependent on the current state. Otherwise, we use the learned features F_s and classifier c_s to convert a user's trajectory τ into a feature vector x, and then into a state-wise stochastic policy $\pi(s, \tau)$.

6. EXPERIMENT AND RESULT

6.1 DragonBox Adaptive

6.1.1 Game Description

DragonBox Adaptive is an educational math puzzle game designed to teach how to solve algebra equations to children ranging from kindergarteners to K-12 students [2], which is evolved from the original game DragonBox [1]. Figure 2 shows the screenshots of the game. Each game level represents an algebra equation to solve. The panel is divided into two sides filled with cards representing numbers and variables. A player can perform algebraic operations by merging cards in the equation (e.g., ‘`-a+a`’ → ‘`0`’), eliminating identities (e.g. ‘`0`’ → ‘ ’), or using a card in the deck to perform addition, multiplication, or division on the both sides of the equation. To clear a level, one should eliminate all the cards on one side of the panel except the DragonBox card, which stands for the unknown variable.

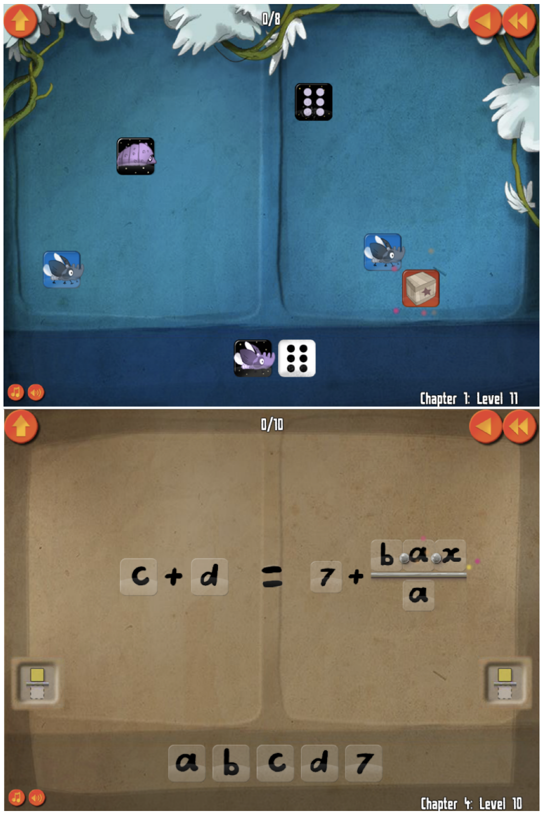


Figure 2: Screenshots of DragonBox Adaptive. (Top) The early stage of the game. It is equivalent to an equation `a-b=-6+a+x`. The card with a starred box on the bottom right is the DragonBox. (Bottom) The game teaches more and more concepts, and eventually kids learn to solve complex equations.

Table 1 shows the examples of states, actions, transitions, and policies in the game. Since DragonBox Adaptive is a card puzzle game, the game state and available actions are well discretized. A state is a pair of level ID and the current equation (e.g. `lv3.x-1=0`). Available actions for a specific state are the available movements, or algebraic operations, that the game mechanics allow the players to perform (e.g. add `a` on both sides or subtract `a` on both sides). Performing an action moves a user from one state to another state (e.g. an action adding `a` on both sides moves a user from `lv3.x-a=0` to `lv3.x-a+a=0+a`), which naturally defines a transition function. Since the transition is deterministic and injective, we use a notation $s \rightarrow s'$ for a transition from s to $s' =$

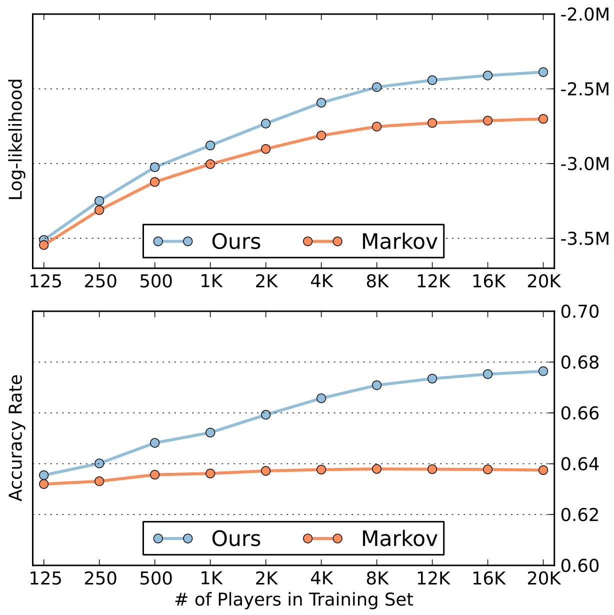


Figure 3: Performances of our predictor and that of the Markov predictor with different sizes of training data.

$f(s, a)$ (e.g. lv3.x-a=0 → lv3.x-a+a=0+a).

We collected the game logs from the Norway Algebra Challenge [3], a competition solving as many algebra equations as possible in DragonBox Adaptive. About 36,000 K-12 students across the country participated in the competition, which was held on January 2014 for a week. One characteristic of this dataset is a low quit rate, which is achieved because students intensively played the game with classmates to rank up their class. About 65% of the participants played the game more than an hour, and more than 200 equations were solved per student on average. This is important in our task because we can collect a lot of training data even in higher levels. After cleaning, we collected 24,000 students' logs with about 280,000 states, 540,000 transitions, and 21 million moves. 4,000 student's logs were assigned to test data and the rest as training data. As the parameters of our framework were determined with another Algebra Challenge dataset separate from Norway Challenge, there was no learning from the test data.

6.1.2 *Overall Performance*

Figure 3 shows the performances of our framework and those of the Markov predictor with different sizes of training data. We use the Markov predictor as a baseline, because it is another data-driven policy predictor working on a state-action graph structure, and because our model is using the Markov predictor when it could not find relevant features to the given state.

In both metrics, the performance of our predictor increases with the size of training data. As it has not converged yet, we can even expect better performance with additional training data. For the Markov predictor, its performance notably improves with the size of training data only in log-likelihood metric. We believe this is because the accuracy rate mostly comes from frequently visited states, which the predictor already reached to the point with no improvement, while a significant portion of the log-likelihood value comes from sparsely visited states, which improves as it gathers more data.

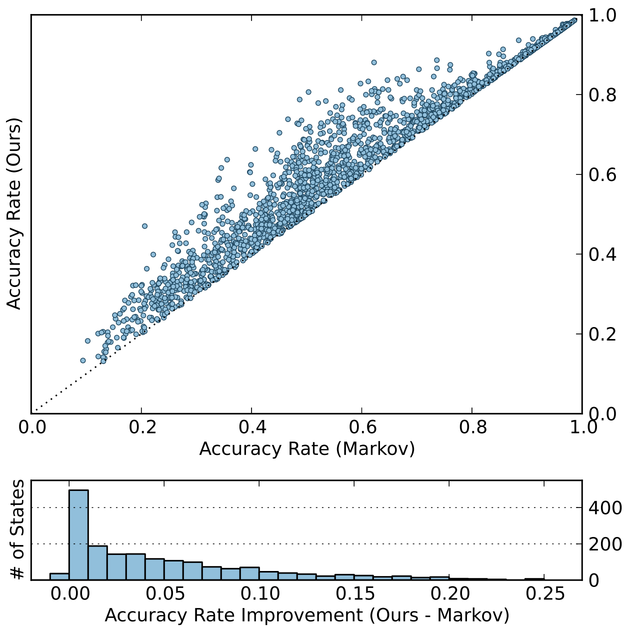


Figure 4: (Top) Scatter plot of accuracy rates of our predictor versus that of the Markov predictor for each state with learned features. We see the performance of our predictor is strictly better than that of the Markov predictor in most cases. (Bottom) Histogram of accuracy rate improvement, while one count is a state with learned features.

The difference between the two methods is increasing as the size of training data increases. With 20,000 user trajectories as training data, the log-likelihood of our predictor is almost 12% lower than that of the Markov predictor and the accuracy rate improves from 64% from 68%. Since the Markov predictor gives a static policy, we can say this gain comes from considering individual behavior differences. Also, even with a relatively small size of training data, the performance of our method is still higher than that of the Markov predictor. We believe our predictor performs strictly better than the Markov predictor even with insufficient data, because our method switches to the latter when it does not have enough confidence on selecting features.

6.1.3 *Statewise Performance*

In this subsection, we further analyze the performances of two methods in the smaller scope. Our predictor learned 1,838 logistic regression classifiers with 20,000 player trajectories as training data. Considering there are about 280,000 game states in total, our method selected no features for more than 99% of the states and decided to use the Markov predictor instead. However, more than 60% of the move-

Next State	Feature	Weight
lv.3.4.1.x+0=b-a	$\mathbb{1}[(\texttt{lv.3.5.1.x+a=b} \rightarrow \texttt{lv.3.5.1.x+0=b-a}) \in \tau_u]$	0.583
	$\mathbb{1}[(\texttt{lv.3.4.1.x+a=b} \rightarrow \texttt{lv.3.4.1.x+0=b-a}) \in \tau_u]$	0.568
	$\mathbb{1}[(\texttt{lv.3.4.1.x+a=b} \rightarrow \texttt{lv.3.4.1.x+0=b-a}) = (s_{t-3}, a_{t-3})]$	0.387
	$\mathbb{1}[(\texttt{lv.2.19.x*b+a=c} \rightarrow \texttt{lv.2.19.x*b+0=c-a}) \in \tau_u]$	0.181
lv.3.4.1.x+a-a=b-a	$\mathbb{1}[(\texttt{lv.3.4.1.x+a+a=b-a} \rightarrow \texttt{lv.3.4.1.x+0=b-a}) = (s_{t-3}, a_{t-3})]$	0.466
	$\mathbb{1}[(\texttt{lv.3.4.1.x+a=b} \rightarrow \texttt{lv.3.4.1.x+a-a=b-a}) = (s_{t-4}, a_{t-4})]$	0.457
	$\mathbb{1}[(\texttt{lv.3.3.2.x+a+b=c} \rightarrow \texttt{lv.3.3.2.x+a+b-b=c-b}) = (s_{t-7}, a_{t-7})]$	0.389
	$\mathbb{1}[(\texttt{lv.2.3.x+1=a} \rightarrow \texttt{lv.2.3.x+0=a-1}) \in \tau_u]$	0.119
lv.3.4.1.x+a+a=b+a	$\mathbb{1}[(\texttt{lv.3.4.1.x+a+a+a=b+a+a} \rightarrow \texttt{lv.3.4.1.x+a+a=b+a}) = (s_{t-2}, a_{t-2})]$	0.247
	$\mathbb{1}[(\texttt{lv.3.4.1.x+a=b} \rightarrow \texttt{lv.3.4.1.x+a+a=b+a}) = (s_{t-6}, a_{t-6})]$	0.192
	$\mathbb{1}[(\texttt{lv.1.18.x+a+a=a+b} \rightarrow \texttt{lv.1.18.x+a+0=b+0}) \in \tau]$	0.128
	$\mathbb{1}[(\texttt{lv.2.3.x+(-x)/(-x)+x/x=1+a} \rightarrow \texttt{lv.2.3.x+(-x)/(-x)+x/x-1=a+0}) \in \tau]$	0.115

Table 2: Selected features with high weights for predicting a transition from state 'lv.3.4.1.x+a=b' to each available action. The selected features closely related to the task it is going to predict. Interesting features mentioned in the text are highlighted.

Transition	Accuracy		Recall	
from x+a=b	LogReg	Markov	LogReg	Markov
x+0=-a+b	88.0	76.7	88.1	76.7
x+a-a=-a+b	60.6	19.2	60.3	19.2
x+a+a=a+b	11.5	3.7	11.6	3.8

Table 3: Accuracy and recall rate for each action on state lv.3.4.1.x+a=b. The performance of highlighted transitions are almost tripled.

ments in training data starts from the states with corresponding logistic regression classifiers. It means our framework invested its resources on a small set of states, which are so influential that they govern the majority of the prediction tasks.

Figure 4 shows the accuracy rates in both predictors for each state that learned a logistic regression classifier. We can see the performance of our predictor is better than that of the Markov predictor in the most cases. The other case is ignorable: the Markov performs better than ours in less than 2% of the states with logistic regressors, while the average performance drop for those states is 0.03%. This observation further supports our argument that our model performs strictly better than the Markov model because of our robust feature selection process.

Table 3 gives the evaluations on a state that showed an accuracy rate improvement from 63% to 80%. Since we are evaluating stochastic policies, we use the following definition for accuracy and recall for a pair (s, a):

$$\frac{\sum_{u \in U} \sum_{t \in [0,t_u]} \mathbb{1}[(s,a) = (s_v^t, a_v^t)] \cdot \tilde{\pi}(a|s, \tau_u^t)}{\sum_{u \in U} \sum_{t \in [0,t_u]} \mathbb{1}[(s,a) = (s_v^t, a_v^t)]},$$

$$\frac{\sum_{u \in U} \sum_{t \in [0,t_u]} \mathbb{1}[(s,a) = (s_v^t, a_v^t)] \cdot \tilde{\pi}(a|s, \tau_u^t)}{\sum_{u \in U} \sum_{t \in [0,t_u]} \mathbb{1}[s = s_v^t] \cdot \tilde{\pi}_u^t(a|s, \tau_u^t)}.$$

In the table, the starting equation is x+a=b, and a player can perform addition or subtraction with a card 'a' on the deck. There are three possible transitions because the game mechanics allow two ways to subtract: one putting '-a' card next to 'a' card, and another one putting '-a' card over 'a' to merge them to '0'. From now on, we will call them 'normal subtraction' and 'subtraction with merging'. Which move to use among them does not affect to clear the game, but the game can be cleared more efficiently with the latter move. The third transition is addition making the current equation to the equation x+a+a=b+a. This is not the right way to solve the level, because the DragonBox is not going to be isolated. We will call it 'unnecessary addition'.

For the normal subtraction, the predictive power is more than tripled. We believe our predictor successfully captured this habitual movement, which we also believe is not likely to change once fixed. Indeed, DragonBox Adaptive does not provide a strong reward on decreasing the number of movements, nor suggest a guide to promote the students using the subtraction with merging. For the unnecessary addition, the predictive power is also more than tripled. Because one must have seen similar problems several times, students rarely makes this mistake (3.8% in training data). In spite of such sparsity, our predictor improved its predictive power over that of the Markov predictor.

6.1.4 Learned Features and Feature Weights

In this subsection, we take a look at the learned features and classifiers. Most of the learned features with high weights are closely related to the task we are trying to predict when inspected by experts. Table 2 shows some of the learned features with high weights in the classifier for each transition in the previous subsection: subtraction with merging, normal subtraction, and unnecessary addition. Most of the selected features with high weights are actually related to the action that it is going to predict. For example, the movement in the first feature in the table x+a=b → x+0=b-a is exactly same as the movement we are trying to predict. The only difference is the level IDs. Note that having a feature set of the future level (lv3.5.1) is possible because our game progression forces students to visit previous levels when a student fails to clear a level.

For another example, the movement in the eleventh feature in the table x+a+a=a+b → x+a+0=b+0 implies an unnecessary

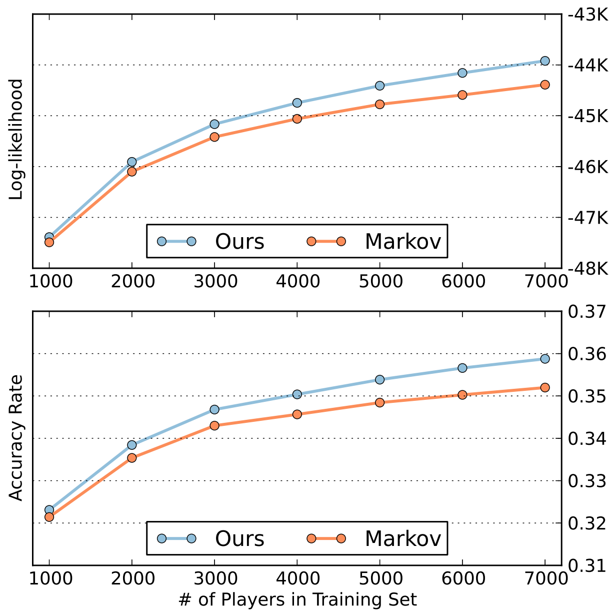


Figure 5: Performances of two methods on Refraction.

addition was performed, because the game level `1.18` starts from `x+a=b`: to reach the equation `x+a+a=a+b`, one has to perform the unnecessary action ahead. It is also interesting that the game level `1.18` is almost 25 levels away from the current level `3.4.1`. It would be a natural assumption that only recent playdata affect a user's behavior, but this provides an evidence that this is not the case. Considering that a player can only proceed a level after correcting previously made mistakes, it also implies the learning curve of this concept is relatively shallow.

One more thing to mention is that when a feature is decribing more recent part of the playdata, it tends to have a higher weight. In the table, features from level 3 usually receives higher weights compared to the features from level 1 or 2. This is intuitive because a user is more likely to change his or her behavior as time passes. This implies that our model is also capturing the temporal behavior of individuals.

6.2 Refraction

To demonstrate that our framework can be used on other systems without additional authoring, we also run an experiment with another educational game, Refraction [4]. As we use the same experimental setting and game data as in Liu et al. [15], we omit the details of the game and the state-action model. The dataset contains 8,000 users' gameplay data, with about 70,000 states, 460,000 transitions, and 360,000 moves. 1,000 users' gameplay data is assigned to a test set, and the rest as a training set. We use the gameplay data from level 1 to level 8 to predict the movements in level 8. There are no changes in our framework, except that the smoothing parameter ϵ in the log-likelihood metric is set to 0.3 to match the performance of the Markov predictor used in previous work [15].

Figure 5 shows the performance of our predictor and the Markov predictor. We see our predictor performs better than the Markov predictor, although the improvement is much smaller compared to DragonBox Adaptive. We believe this is because level 8 is an early level, and we do not have enough data. Level 8 is the first level without the tutorial, it would be difficult to detect a confident signal describing individual behavior. In other words, students did not have enough opportunity to show their personality. We could not run another experiment on a later level due to lack of players from the high drop-out rate. Moreover, the Refraction dataset (360,000 moves) is much smaller than the DragonBox Adaptive dataset (21 million moves), while the total number of transitions is similar in both.

Nevertheless, we successfully showed that our framework can be used in a different system with no additional expert authoring, and showed our predictor still performs better than the Markov predictor.

7. CONCLUSION AND FUTURE WORK

Modeling user behavior in open-ended environments has the potential to greatly increase undestanding of human learning processes, as well as helping us better adapt to students. In this paper, we present a data-driven individual policy-learning framework that reduces the burden of hand-designing a cognitive model and system-specific features. Our framework automatically selects relevant features from a default feature set defined on a general state-action graph structure, and learns an individual policy from the trajectory of a player. We apply our method to predict player movements in two educational puzzle games and showed our predictor outperforms a baseline predictor. We also show that the performance improvement comes not only from frequently observed movements, but also from sparsely observed erroneous movements. Finally, we see our robust feature selection makes the predictor more efficient, powerful, and interpretable by investing its resources on a small set of influential states and relevant features.

We see numerous opportunities for further improvement of our framework. First, we can experiment with different classifiers instead of a logistic regressor. Since a logistic regression model is a single layer artificial neural network (ANN), we believe using a multi-layered ANN is a natural extension to improve its predictive power. Using an ensemble of classifiers would be another way to boost the performance. Second, we can add more graph navigation features into the default feature set. A feature specifying whether a transition is not visited only when it was available to the user is the first thing to try, because it specifies whether a certain behavior has been avoided intentionally or if the user simply did not have an opportunity to make such a choice. A visit indicator of a specific chain of transitions or the time spent on a certain state can also be possible features. Finally, we can try other feature selection techniques. Recursive feature elimination or L1-based feature selection might produce a better result because univariate approaches, such as our chi-squared test, do not consider the effect of multiple features working together [10].

Overall, we are also very interested in building applications based on our framework. Integrating the individual behavior predictor into user-specific content generation such as a personalized hinting system or an adaptive level progression would be the first step. Moreover, we believe our framework will be also useful in other fields for learning individual behavior, such as spoken dialogue systems [14], robotic learning from demonstration [7], and recommender systems [18].

8. ACKNOWLEDGMENTS

This work was supported by the University of Washington Center for Game Science, the National Science Foundation Graduate Research Fellowship grant No. DGE-0718124, the Bill and Melinda Gates Foundation, and Samsung Scholarship Foundation. The authors would also like to thank Travis Mandel for helpful discussions.

9. REFERENCES

[1] Dragonbox. `http://www.dragonboxapp.com/`. *WeWantToKnow*, Retrieved 2014-02-27.

[2] Dragonbox adaptive. `http://centerforgamescience.org/portfolio/dragonbox/`. *Center for Game Science*, Retrieved 2014-02-27.

[3] Norway algebra challenge. `http://no.algebrachallenge.org/`. *Algebra Challenge*, Retrieved 2014-02-27.

[4] Refraction. `http://centerforgamescience.org/portfolio/refraction/`. *Center for Game Science*, Retrieved 2014-02-27.

[5] D. W. Albrecht, I. Zukerman, and A. E. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User modeling and user-adapted interaction*, 8(1-2):5–47, 1998.

[6] V. Aleven, B. M. McLaren, J. Sewall, and K. R. Koedinger. The cognitive tutor authoring tools (ctat): Preliminary evaluation of efficiency gains. In *Intelligent Tutoring Systems*, pages 61–70. Springer, 2006.

[7] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[8] Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. T. Seaton, and D. E. Pritchard. Model-Based Collaborative Filtering Analysis of Student Response Data: Machine-Learning Item Response Theory. In *EDM*, pages 95–102, 2012.

[9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[10] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[11] R. K. Hambleton, H. Swaminathan, and H. J. Rogers. *Fundamentals of item response theory*. Sage, 1991.

[12] A. R. Jansen, D. L. Dowe, and G. E. Farr. Inductive inference of chess player strategy. In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, PRICAI'00, pages 61–71, Berlin, Heidelberg, 2000. Springer-Verlag.

[13] L. A. Jeni, J. F. Cohn, and F. De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 245–251. IEEE, 2013.

[14] D. J. Litman and S. Pan. Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, 12(2-3):111–137, 2002.

[15] Y.-E. Liu, T. Mandel, E. Butler, E. Andersen, E. O'Rourke, E. Brunskill, and Z. Popović. Predicting player moves in an educational game: A hybrid approach. In *EDM*, pages 106–113, 2013.

[16] M. Mavrikis. Data-driven modelling of students' interactions in an ILE. In *EDM*, pages 87–96, 2008.

[17] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[18] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[19] M. Prensky. Computer games and learning: Digital game-based learning. *Handbook of computer game studies*, 18:97–122, 2005.

[20] S. Ritter, T. K. Harris, T. Nixon, D. Dickison, R. C. Murray, and B. Towle. Reducing the Knowledge Tracing Space. In *EDM*, pages 151–160, 2009.

[21] D. W. Shaffer. *How computer games help children learn*. Macmillan, 2006.

[22] G. Synnaeve and P. Bessière. A bayesian model for plan recognition in rts games applied to starcraft. In *Proceedings of the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2011)*, pages 79–84, 2011.

[23] B. Tastan and G. R. Sukthankar. Learning policies for first person shooter games using inverse reinforcement learning. 2011.

[24] N. Thai-Nghe, L. Drumond, A. Krohn-Grimberghe, and L. Schmidt-Thieme. Recommender system for predicting student performance. *Procedia Computer Science*, 1(2):2811–2819, 2010.

[25] N. Thai-Nghe, T. Horváth, and L. Schmidt-Thieme. Factorization Models for Forecasting Student Performance. In *EDM*, pages 11–20, 2011.

[26] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[27] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *Artificial Intelligence in Education*, pages 171–180. Springer, 2013.

[28] A. Zook and M. O. Riedl. A temporal data-driven player model for dynamic difficulty adjustment. In *AIIDE*, 2012.

Predicting Learning and Affect from Multimodal Data Streams in Task-Oriented Tutorial Dialogue

Joseph F. Grafsgaard[1], Joseph B. Wiggins[1], Kristy Elizabeth Boyer[1],
Eric N. Wiebe[2], James C. Lester[1]

[1]Department of Computer Science [2]Department of STEM Education
North Carolina State University, Raleigh, NC, USA

{jfgrafsg, jbwiggi3, keboyer, wiebe, lester}@ncsu.edu

ABSTRACT

Learners experience a wide array of cognitive and affective states during tutoring. Detecting and responding to these states is a core problem of adaptive learning environments that aim to foster motivation and increase learning. Recognizing learner affect through nonverbal behavior is particularly challenging, as students display affect across numerous modalities. This study utilizes an automatically extracted set of multimodal nonverbal behaviors and task actions to predict learning and affect in a data set of sixty-three computer-mediated human tutoring sessions. Predictive models of post-session self-reported *engagement*, *frustration*, and *learning* were evaluated with leave-one-out cross-validation. Nonverbal behaviors conditioned on task events and typing were found to be more predictive than incoming student self-efficacy and pretest score. Face and gesture were predictive of *engagement* and *frustration*, while face and posture was predictive of *learning*. The nonverbal model features captured moments when students were most active on the task, such as writing and testing the Java program. These results provide initial evidence linking affect, moment-by-moment multimodal nonverbal behavior, and task performance during tutoring. They improve understanding of learner affect and enable automated tutorial interventions that adapt to student states as a highly effective human tutor would.

Keywords

Affect, engagement, frustration, facial expression recognition, gesture, posture, multimodal nonverbal behavior, computer-mediated tutoring

1. INTRODUCTION

Mastery-oriented one-to-one human tutoring may provide two sigma learning gains [3]. In order to match this high bar of expert human tutoring effectiveness, automated tutorial interventions may need to be designed with both learner knowledge and motivation in mind [5, 9, 12, 31]. Highly effective human tutors simultaneously address cognitive and affective states of learners, adapting to the appropriate level of content difficulty and improving learner motivation through personalized instruction [25]. Just as human tutors consider more than task performance of the student, it may be necessary to bolster automated tutorial interventions with additional information regarding learner affect from nonverbal behavior [32].

Early studies of nonverbal behavior in tutoring relied on manual observations of affect and nonverbal behavior [2, 8, 14, 38]. More recently, automated techniques have been leveraged to track nonverbal behaviors [1, 11, 15, 17, 22]. Most studies have examined individual modalities in detail, such as facial expression [17, 35], posture [10, 18], or gesture [18, 28]. However, a much smaller set of studies has examined multiple modalities of nonverbal behavior [1, 11, 22]. It is likely that a multimodal combination of automatically tracked affective data streams would need to be considered to best adapt to learner affect during tutoring [9].

Facial expression is a particularly informative modality for analysis of affect, as indicated by decades of prior research. Many studies have utilized the Facial Action Coding System (FACS), a coding manual that describes the fine-grained movements of the human face as facial action units (AUs) [13]. Recent automated techniques have enabled FACS-based facial expression recognition. In particular, the Computer Expression Recognition Toolbox (CERT), used in this study, provides a state-of-the-art facial expression recognition tool that identifies facial action units [27]. CERT was trained using databases of spontaneous and posed expressions and has been validated on naturalistic video datasets [16, 26, 27]. Thus, frame-by-frame facial action unit tracking provides detailed affective information that is readily synchronized with additional modalities.

Gestures have been tangentially reported on in the intelligent tutoring systems community, but other phenomena were the primary focus of those studies [14, 38]. Recent work has begun to describe and track cognitive-affective gestures [15, 18, 21, 28]. This study uses an algorithm that processes three-dimensional Kinect™ depth images to identify when one or two hands contact the lower face [15].

Posture has been used as an affective feature in multiple systems, but interpretation of postural movements is very complex [10, 22, 23]. One result replicated across multiple studies is that increases in postural movement are linked with negative affect or disengagement [10, 15, 33, 38]. Early work used expensive pressure-sensitive chairs [22, 38]. Newer techniques rely on computer vision to interpret posture from video [10, 15, 33]. This study uses an algorithm that processes Kinect depth images to identify how far away the student is seated [15].

The analysis reported in this paper combines an automatically extracted set of multimodal nonverbal behaviors and task actions to predict learning and affect in a data set of sixty-three computer-mediated human tutoring sessions. Relative frequencies of nonverbal behaviors contingent on task events and typing statuses were used as predictive features. Model averaging identified the top twenty predictive features per model. Three models were built using stepwise forward linear regression with the Bayesian Information Criterion (BIC) to predict retrospective self-reports of *engagement* and *frustration*, as well as normalized learning gains. The models were evaluated with leave-one-out cross-validation. Nonverbal features were found to be more predictive than incoming student self-efficacy and pretest scores. Face and gesture were predictive of *engagement* and *frustration*, while face and posture were predictive of *learning*. Additionally, the majority of nonverbal predictive features occurred when the student was writing and testing the Java program, which shows that these moments may be most salient to affect. Further studies in this vein can inform the design of automated tutorial interventions in order to adapt to student affect as a highly effective human tutor would.

2. RELATED WORK

Few studies have examined multimodal nonverbal behavior features in a tutoring context. An initial study by Kapoor and Picard considered prediction of experienced teacher judgments of affect in young student (8-11 years of age) interactions with a game, Fripple Place [22]. Face, posture, and task features were used in a mixture of Gaussian processes. These models performed well at predicting teacher judgments of affect, which was an important initial step toward detecting cognitive-affective states involved in cognitively demanding tasks.

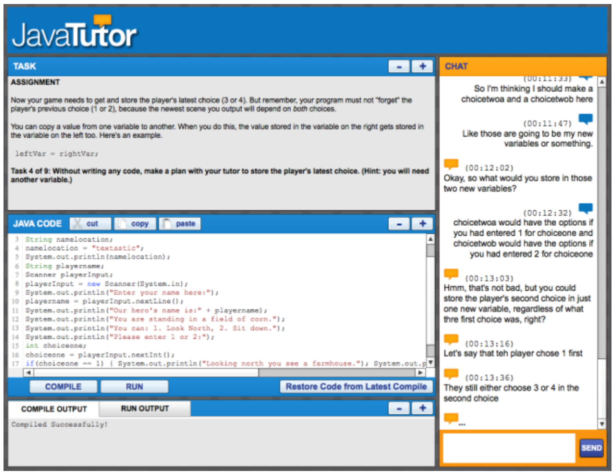
In research on the AutoTutor intelligent tutoring system, multimodal features were used to predict affect labels by expert judges [11]. Emotion labels were manually selected using six affective states (*boredom*, *confusion*, *engagement/flow*, *frustration*, *delight*, *surprise*) and a non-emotional/neutral choice at fixed time intervals and spontaneously across thirty-eight approximately half-hour tutoring sessions. These labels were then predicted using a multimodal feature set including manually annotated Facial Action Coding System facial movements, automatically extracted dialogue features from fifteen seconds prior to an emotion label, and automatically extracted posture features using a pressure-sensitive chair. The fully-featured models of face, dialogue, and posture produced the best levels of agreement, with Cohen's K of 0.33 for fixed emotion judgments and 0.39 for spontaneous ones.

Another line of research has investigated the use of multiple sensor technologies with the Wayang Outpost intelligent tutoring system [1]. A real-time facial expression analysis tool trained on posed cognitive-affective displays, MindReader, was used to estimate levels of *agreeing*, *concentration*, *interest*, *thinking*, and *unsureness*. Additionally, a pressure-sensitive mouse, skin conductance bracelet, and pressure-sensitive chair were also used. Student cognitive-affective self-reports were given during the tutoring session for states of *confidence*, *excitement*, *frustration*, and *interest*. Stepwise regression models were constructed across combinations of modalities (including tutorial context). The results found that best fit models were achieved through combinations of facial expression and tutoring context (for *confidence*, *excitement*, and *interest*) and posture and tutoring context (for *frustration*). The corresponding model effect sizes for the best fit models ranged from r = 0.54 to 0.83. A follow-up validation study was also conducted with a new set of students from a different school and a lower age group [7]. The results found that the previously used features were only partially generalizable to the validation population, with reduced accuracies for most features. This underscores the necessity of identifying generalizable affective features.

In contrast with prior studies, this paper presents models predicting affective and learning outcomes from moment-to-moment nonverbal behavior and task performance. This line of investigation seeks to identify nonverbal behavioral correlates of both affect and learning. The present results indicate that facial expression, gesture, and posture may have differing affective interpretations based on the tutoring context in which they occur. The nonverbal features were found to be more predictive than incoming student self-efficacy and pretest score. Additionally, the nonverbal features were largely contingent upon student work on the programming task, illustrating that these moments of student task activity may be most salient to affect. Further studies in this vein may produce affect recognition that enables detecting and responding to learner affect as a highly effective human tutor would.

3. TUTORING STUDY

The corpus consists of computer-mediated tutorial dialogue for introductory computer science collected during the 2011-2012 academic year. Students (N=67) and tutors interacted through a web-based interface that provided learning tasks, an interface for computer programming, and textual dialogue. The participants were university students in the United States, with average age of 18.5 years (*stdev*=1.5). The students voluntarily participated for course credit in an introductory engineering course, but no prior computer science knowledge was assumed or required. Each student was paired with a tutor for a total of six sessions on different days, limited to forty minutes each session. Recordings of the sessions included database logs, webcam video, skin conductance, and Kinect depth video. This study analyzes the database logs, webcam video, and Kinect depth video from the first lesson as a multimodal tutoring corpus, described further in Section 4. The JAVATUTOR interface is shown in Figure 1.



ASSIGNMENT

Now your game needs to get and store the player's latest choice (3 or 4). But remember, your program must not "forget" the player's previous choice (1 or 2), because the newest scene you output will depend on *both* choices.

You can copy a value from one variable to another. When you do this, the value stored in the variable on the right gets stored in the variable on the left too. Here's an example.

```
 leftVar = rightVar;
```

Task 4 of 9: Without writing any code, make a plan with your tutor to store the player's latest choice. (Hint: you will need another variable.)

```
String namelocation;
namelocation = "textastic";
System.out.println(namelocation);
String playername;
Scanner playerInput;
playerInput = new Scanner(System.in);
System.out.println("Enter your name here:");
playername = playerInput.nextLine();
System.out.println("Our hero's name is:" + playername);
System.out.println("You are standing in a field of corn.");
System.out.println("You can: 1. Look North, 2. Sit down.");
System.out.println("Please enter 1 or 2:");
int choiceone;
choiceone = playerInput.nextInt();
if(choiceone == 1) { System.out.println("Looking north you see a farmhouse."); System.out.p
```

```
Compiled Successfully!
```

(00:11:33)
So I'm thinking I should make a choicetwoa and a choicetwob here

(00:11:47)
Like those are going to be my new variables or something.

(00:12:02)
Okay, so what would you store in those two new variables?

(00:12:32)
choicetwoa would have the options if you had entered 1 for choiceone and choicetwob would have the options if you had entered 2 for choiceone

(00:13:03)
Hmm, that's not bad, but you could store the player's second choice in just one new variable, regardless of what thre first choice was, right?

(00:13:16)
Let's say that teh player chose 1 first

(00:13:36)
They still either choose 3 or 4 in the second choice

...

Figure 1. The JAVATUTOR interface

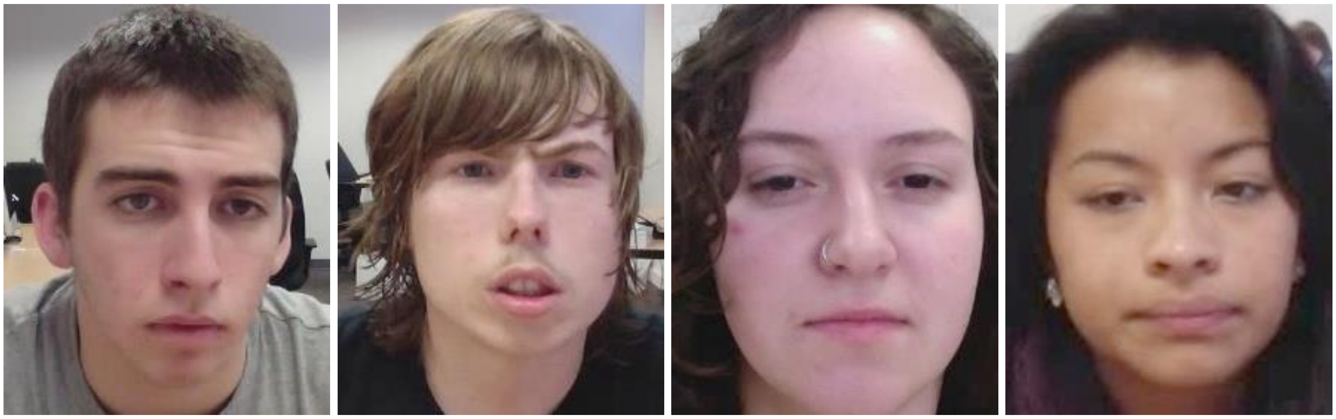


Figure 2. Facial action units recognized by CERT (left to right): AU1 (Inner Brow Raiser) & AU2 (Outer Brow Raiser), AU4 (Brow Lowerer), AU7 (Lid Tightener), AU14 (Mouth Dimpler)

On a day prior to the first tutoring session, students completed a set of surveys to measure incoming student characteristics. Two of these pre-session survey instruments are analyzed in this paper: computer science domain-specific self-efficacy and general self-efficacy. The computer science self-efficacy measure is comprised of the confidence items from the Computer Science Attitude Survey [37]. General self-efficacy was measured using the New General Self-Efficacy instrument [6]. Before each session, students completed a content-based pretest. After each session, students answered a post-session survey and posttest (identical to the pretest). The post-session survey items included the User Engagement Survey (UES) [30] and the NASA-TLX workload survey [20], which included an item for Frustration Level. There is a recent validation of the UES measure with further information [36].

4. MULTIMODAL TUTORING CORPUS

The tutoring session database logs were combined with automated facial action unit tracking on webcam videos and gesture and posture tracking across Kinect depth image frames. The automated tracking techniques are described in the following subsections. The resulting multimodal features are described in Section 4.3.

4.1 Facial Expression Recognition

A state-of-the-art facial expression recognition tool, the Computer Expression Recognition Toolbox (CERT) [19], was used for frame-by-frame tracking of a wide variety of facial action units. CERT finds faces in a video frame, locates facial features for the nearest face, and outputs weights for each tracked facial action unit using support vector machines. For a detailed description of the technology used in CERT, see [25]. The tutoring video corpus is comprised of approximately four million video frames totaling thirty-seven hours across the first tutoring session. Two session recordings were missing due to human error (*N*=65).

We previously validated an adjustment to CERT output that produced excellent aggregate agreement with manual FACS annotations across a subset of five action units [16]. The adjustment involves subtraction of the average value for each facial action unit as a baseline in order to reduce systematic tracking error. While any positive output value indicates that CERT recognizes an action unit, we empirically found that a higher threshold may reduce false positives. Thus, we consider an action unit to be present when the baseline-adjusted CERT output is at least 0.25. Examples of the five selected facial action units and their FACS codes (e.g., AU1) are shown in Figure 2.

4.2 Gesture and Posture Detection

Previously developed posture and gesture tracking techniques were applied to the recorded Kinect depth images. The posture tracking algorithm was previously evaluated to be 92.4% accurate, while gesture tracking was found to be 92.6% accurate [15]. The tracking algorithms were run on all sessions, but four sessions had no Kinect recordings due to human error (*N*=63). Examples of one-hand-to-face and two-hands-to-face gestures are shown in Figure 3.

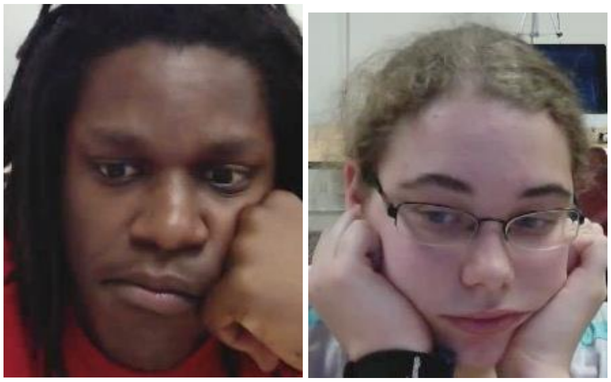


Figure 3. Examples of hand-to-face gestures

The median head distance of students at each workstation was selected as the “mid” postural position. Distances at one standard deviation (or more) closer or farther than “center” were labeled as “near” or “far,” respectively. Additionally, postural movements were identified based on acceleration of the head tracking point. The absolute sum of frame-to-frame acceleration was accumulated in a rolling one-second window at each frame. The average amount of acceleration in a one-second interval was computed across all students. If acceleration in the present interval was above average, it was marked as a postural movement (PosMove). Average frequencies of gesture and posture features are shown in Table 1. Students tended to spend more time in a Mid postural position and most frequently did not display a hand-to-face gesture. Additionally, students moved less than average during each interval, indicating that there were short moments of high movement that raised the average.

Table 1. Average frequency of gesture and posture features

Feature	Avg. Freq.	Feature	Avg. Freq.
Near	15%	OneHand	16%
Mid	68%	TwoHands	5%
Far	17%	NoGesture	79%
PosMove	29%		
NoMove	71%		

4.3 Multimodal Features

The automatically recognized nonverbal behaviors were combined with task-related features in order to form the multimodal tutoring corpus. As students worked on programming tasks, the database logged dialogue messages, typing, and task progress. Tutorial dialogue occurred at any time during the sessions, with student and tutor messages sent asynchronously (StudentMsg and TutorMsg, respectively). As a student completed the programming task, he or she would also press a compile button to convert the Java program code into a format that is ready to run. These compile attempts may be successful (CompileSuccess) or fail due to an error in the program code (CompileError). The student would also run his or her program (RunProgram) in order to test the output and interact with it. In parallel with the task events described above, the database logged whether the student was typing at any given moment. The student may not be typing anything (NotTyping), working on the program code (Coding), or typing a message to the tutor (TypingMsg) at each moment. Additionally, the student was considered to have paused on the task if he or she had made changes to the program and then stopped. This sort of break may be due to the student having resolved the current task, taking a moment to think, or going off-task; therefore, it was introduced as a task event (TaskPause). The average frequency of each task event and typing status is shown in Table 2. The majority of time intervals occurred after tutor messages and when students were not typing. These majority events represent moments when the student may have been reading the task description or reflecting on tutor messages. Tutors were also more active in the dialogue than students, resulting in more time following tutor messages.

Table 2. Average frequency of task events and typing status

Task Event	Avg. Freq.	Typing Status	Avg. Freq.
CompileError	1.7%	Coding	15%
CompileSuccess	2.1%	TypingMsg	12%
RunProgram	7.9%	NotTyping	73%
StudentMsg	26.4%		
TutorMsg	53.1%		
TaskPause	8.8%		

Task events and typing statuses were combined with nonverbal behaviors at one-second intervals across each tutoring session. The most recent event of a given type (nonverbal, task, typing) was counted as the current value at each interval. For instance, if a student had been typing but stopped after half a second into the current interval, the typing status would be assigned to NotTyping.

A tutoring session excerpt is shown in Figure 4. The excerpt shows a rich set of nonverbal behaviors occurring around student work on the programming task. This student produced a variety of facial expressions, particularly when examining and testing the Java program. Additionally, the student performed a one-hand-to-face gesture prior to compiling the program. The corresponding multimodal features for a segment of the excerpt are shown in Figure 5 (top of next page). The multimodal feature vectors cover a twelve-second segment from the excerpt.

26:54	Tutor:	ready?
26:59	Student:	yes! [*Student starts coding*]
28:02	Student:	TaskPause [*Student stops coding*]
28:03	Student:	Gesture: OneHandToFace; Face: AU2 & AU14
28:12	Student:	Task: CompileSuccess; Face: AU2
28:14	Student:	Face: AU14
28:17	Student:	Task: RunProgram; Face: AU1
28:19	Student:	Face: AU7
28:21	Tutor:	excellent

Figure 4. Tutoring session excerpt

Relative frequencies of nonverbal behavior were calculated separately for task events and typing status. For instance, at each one-second time interval, AU1 was marked as present or absent. Each interval was associated with a task event, with frequency counts tabulated across all task events. The relative frequency of AU1 presence and absence was computed across these task-contingent counts. Thus, the percentages of time intervals occurring with specific task events and particular values of AU1 presence or absence sum to one hundred percent. For instance, one student may have AU1 after RunProgram 2.12% of the time and NoAU1 after RunProgram 3.24% of the time. These relative frequencies sum to one hundred percent when combined with the remainder of task-contingent relative frequencies of AU1. Relative frequencies were similarly computed across typing statuses for each nonverbal behavior. Thus, the relative frequencies account for the percent of time in which a student displayed a nonverbal behavior after a specific task event or during a particular typing status (i.e., a student with a 5% relative frequency of OneHand after TutorMsg in a thirty minute session would have displayed a one-hand-to-face gesture for a total of ninety seconds after tutor messages). This resulted in a set of one hundred and sixty-two nonverbal features contingent upon task events and typing statuses. The distribution of these multimodal features across nonverbal behaviors, task events, and typing statuses is shown in Table 3.

Table 3. Counts of multimodal features across nonverbal behaviors, task events, and typing statuses

	Task Event	Typing Status
AU1	12	6
AU2	12	6
AU4	12	6
AU7	12	6
AU14	12	6
Gesture	18	9
Posture	18	9
PosMove	12	6

<table>
<tr><th></th><th>28:08</th><th>28:09</th><th>28:10</th><th>28:11</th><th>28:12</th><th>28:13</th><th>28:14</th><th>28:15</th><th>28:16</th><th>28:17</th><th>28:18</th><th>28:19</th></tr>
<tr><td>AU1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">AU1</td></tr>
<tr><td>AU2</td><td colspan="2">AU2</td><td></td><td></td><td colspan="8">AU2</td></tr>
<tr><td>AU4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>AU7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>AU7</td></tr>
<tr><td>AU14</td><td colspan="5">AU14</td><td></td><td colspan="4">AU14</td><td></td><td></td></tr>
<tr><td>OneHand</td><td colspan="11">OneHand</td><td></td></tr>
<tr><td>TwoHand</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Posture</td><td colspan="12">FarPosture</td></tr>
<tr><td>PosMove</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Task</td><td colspan="4">TaskPause</td><td colspan="5">CompileSuccess</td><td colspan="3">RunProgram</td></tr>
<tr><td>Typing</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Figure 5. Multimodal feature vectors for a twelve-second segment of tutoring: gray shading indicates presence of a nonverbal behavior, task event, or typing. Time is shown in minutes and seconds from the beginning of the tutoring session.

5. PREDICTIVE MODELS

Fine-grained analyses of multimodal affective expressions are enabled by automated tracking of nonverbal behavior. Such analyses have the potential to reveal previously undiscovered ways in which affective displays relate to task performance, learning, and affective outcomes within a tutoring context. For instance, the same affective expression may have different causes depending on the tutoring context. As a first step toward examining the fine-grained tutoring context of learner affective displays, predictive models of affective and learning outcomes were constructed using the multimodal tutoring corpus, in which facial expression, gesture, and posture are combined with task actions.

Initial feature selection was performed using model averaging in JMP statistical software, which created regression models for all possible combinations of predictive variables [34]. Model averaging is used to identify and remove weakly predictive variables across all models. Specifically, the twenty most predictive variables were selected using the average coefficient estimate from models with one, two, or three predictive variables. The predictive models were then constructed using minimum Bayesian Information Criterion (BIC) in forward stepwise linear regression. These models are conservative in how they select predictive features because the explanatory value of added parameters must offset the BIC penalty for model complexity. Thus, model averaging was used to identify the most generally predictive variables, while minimum BIC was used to constrain model complexity. Tutoring outcomes (*engagement*, *frustration* and *learning*) were the dependent variables. All variables were standardized (i.e., centered on the mean and scaled to unit standard deviation) to enable comparison. The predictive models shown in the following sub-sections have been constructed using the entire corpus, with associated regression coefficients and R^2 values. Additionally, leave-one-out cross-validated R^2 values were computed using the same predictive variables (but different coefficients in each fold) to examine generalizability of the predictive models.

5.1 Predicting Engagement

Each student's Engagement score was the sum of the Focused Attention, Felt Involvement, and Endurability sub-scales in the User Engagement Survey [30] administered following the tutoring session. This model only uses self-reports of engagement from students who fully completed the User Engagement Survey (N=61). The predictive model of *engagement* was composed of three features, including students' incoming computer science self-efficacy, one-hand-to-face gestures after successful compile, and brow lowering (AU4) after sending a student dialogue message. Each of the nonverbal features explains more variance than the trait-based feature of computer science self-efficacy. This seems to indicate that state-based nonverbal features are more indicative of *engagement*. The cross-validated model effect size was $r = 0.39$. The model is shown in Table 4.

Table 4. Stepwise linear regression model for Engagement

Engagement =	**Partial R^2**	**Model R^2**	***p***
0.31 * OneHand after CompileSuccess	0.10	0.10	0.009
-0.31 * AU4 after StudentMsg	0.09	0.19	0.008
0.27 * Computer Science Self-Efficacy	0.07	0.26	0.020
~0 (intercept)			0.959
RMSE = 0.88 standard deviations in Engagement			
Leave-One-Out Cross-Validated R^2 = 0.15			

5.2 Predicting Frustration

The Frustration Level scale from NASA-TLX [20] was the student's retrospective self-report of how insecure, agitated or upset he or she was during the tutoring session. The predictive model of *frustration* included students' incoming general self-efficacy and two features that accounted for the absence of nonverbal behavior. The sole feature predictive of higher *frustration* corresponded with compile errors, which intuitively may be frustrating. The absence of brow lowering (AU4) after running the Java program reinforces a prior result that indicated AU4 as a marker of *frustration* [17]. Also, students with higher general self-efficacy tended to have less *frustration*, as represented in the model. The cross-validated model effect size was $r = 0.41$. The model is shown in Table 5.

Table 5. Stepwise linear regression model for Frustration

Frustration =	Partial R^2	Model R^2	*p*
-0.42 * General Self-Efficacy	0.14	0.14	0.004
-0.56 * NoAU4 after RunProgram	0.08	0.22	0.004
0.42 * NoGesture after CompileError	0.08	0.30	0.011
~0 (intercept)			1.000
RMSE = 0.85 standard deviation in Frustration Level			
Leave-One-Out Cross-Validated R^2 = 0.17			

5.3 Predicting Learning Gain

Normalized learning gain measures how much a student learned relative to what he or she could have learned [29]. This accounts for relative differences in learning between students who scored high or low on the pretest. Normalized learning gain was computed using the following formula if posttest score was greater than pretest score:

$$NLG = \frac{\underline{Posttest - Pretest}}{1 - Pretest}$$

Otherwise, normalized learning gain was computed as follows:

$$NLG = \frac{\underline{Posttest - Pretest}}{Pretest}$$

The predictive model of normalized learning gain is the only one of the three to include postural features. These features indicate that MID and FAR postural positions are predictive of learning, though whether they are positive or negative predictors is dependent upon the tutoring context. Mouth dimpling (AU14) after running the Java program was predictive of learning. This supports a prior result that AU14 is positively associated with learning [17]. Finally, general self-efficacy predicted higher learning gains. The cross-validated model effect size was r = 0.62. The model is shown in Table 6.

Table 6. Stepwise linear regression model for Normalized Learning Gain

Norm. Learning Gain =	Partial R^2	Model R^2	*p*
0.10 * AU14 after RunProgram	0.11	0.11	0.004
0.10 * General Self-Efficacy	0.08	0.19	0.002
-0.12 * MidPosture after CompileError	0.08	0.27	<0.001
-0.21 * FarPosture during Coding	0.04	0.31	<0.001
0.20 * FarPosture after CompileSuccess	0.18	0.49	<0.001
0.43 (intercept)			<0.001
RMSE = 0.24 std. dev. in Normalized Learning Gain			
Leave-One-Out Cross-Validated R^2 = 0.38			

6. DISCUSSION

The results demonstrate that nonverbal behaviors at specific moments in the tutoring session are predictive of *engagement*, *frustration*, and *learning*. The combination of task events, typing, and nonverbal behaviors in multimodal features is predictive beyond incoming student characteristics, such as pretest score and self-efficacy. Additionally, the affective valence (positive or negative) of the nonverbal behaviors depended upon the tutoring context in which they occurred.

The predictive model of *engagement* was composed of three features, including students' incoming computer science self-efficacy, one-hand-to-face gestures after successful compile, and brow lowering (AU4) after sending a student dialogue message. One-hand-to-face gestures may have different affective valence depending on the physical position of the hand. A student may rest his/her head on the hand as a sign of boredom [2], or touch his/her chin in a moment of contemplation [28]. Here, one-hand-to-face gestures after compile success were predictive of higher post-session self-report of *engagement*. This may coincide with student focus on the programming task. In the moments after updating the program code and compiling it, the student is no longer typing and may then reflect on current progress. Brow lowering (AU4) after the student sends a dialogue message, on the other hand, was a predictor of lower *engagement*. This may indicate that a student is having difficulty with the subject matter, most likely responding to a tutor message (in this corpus, tutor messages were predominant and students rarely took initiative in the dialogue). Both of the nonverbal features were more predictive than computer science domain-specific self-efficacy, which was associated with greater *engagement*.

Frustration was significantly predicted by general self-efficacy. Higher levels of general self-efficacy coincided with lower post-session reports of *frustration*. Students with higher general self-efficacy are more confident in their ability to complete difficult tasks and therefore may be less intimidated by a novel learning task. However, inclusion of two nonverbal features doubled the explanatory power of the model. Each of the nonverbal features captured absence of nonverbal behaviors after specific task events. Absence of brow lowering (AU4) after running the Java program was predictive of lower *frustration*. At this point, the student is testing the program to see whether it matches his/her expectation. A prior result on this tutoring corpus found that AU4 was an indicator of *frustration*. Therefore, the present result supports that finding, but also provides a specific tutoring context (running the program) that is particularly meaningful for *frustration*. The sole feature predictive of higher *frustration* corresponded with compile errors (which occur when the program is incorrect). This correspondence between compiling the program and frustration is similar to results of prior analyses of student emotions during computer programming [4, 24]. Not all students had compile errors, so this feature represents those students who may have found the task to be more difficult. The absence of gestures after compile errors may be due to swift tutor interventions to remediate problems with the program. In this case, a student may feel frustrated due to overly active tutoring strategies.

Normalized learning gain was predicted by a combination of students' incoming general self-efficacy, mouth dimpling (AU14) after running the program, and three posture-related features. The model shows that students with more general confidence in their ability to complete novel and difficult tasks tended to learn more than their peers. Displays of AU14 after running the program also were predictive of higher learning gain. Two aspects of AU14 discovered in prior results may shed light on this. First, occurrence of AU14 in general was associated with greater learning gain [16]. Second, AU14 in the first five minutes of tutoring was correlated with higher *frustration*, while AU14 in the last five minutes of tutoring was correlated with greater learning gain [17]. Running the program occurs most frequently during the later portion of the session. So, AU14 displays after running the program may also occur toward the end. With this timing-related interpretation, it may be that continued mental effort throughout the tutoring session is reflected in displays of AU14. Further study of AU14 may confirm whether it is generally an indicator of mental effort.

The posture-related features included both MID and FAR distances. These postural positions may encode information beyond whether a student is sitting at a certain distance from the computer. For instance, when a student is sitting at MID distance, the shoulders may be hunched or the student may have a straight back. FAR postural position was both predictive of higher learning gain (when occurring after compile success) and lower learning gain (when present during coding). It may be that bored students slouched in a FAR position during coding, while relaxed (but active) students were similarly farther back. New tracking methods may be developed to disambiguate these subtleties of posture. Interestingly, postural position was predictive of learning, but moment-to-moment postural movement was not. Discretization across one-second intervals may not have adequately captured brief postural movements.

The predictive models largely include nonverbal features that occur around moments of student work on the programming task. These may be pivotal moments on a student's path to learning, as students are actively working on the task and confirming whether the program works as intended. Prior results in analysis of skin conductance on this tutoring corpus showed that students' physiological responses to compile attempts and failures were associated with *learning* and *frustration* [19]. The predictive models presented in this paper further underscore the importance of tutoring context in interpretation of nonverbal behavior.

7. CONCLUSION

This paper presented a multimodal analysis of automatically recognized nonverbal behaviors and task events. State-of-the-art facial expression recognition was leveraged, along with depth video-based gesture detection and posture tracking algorithms, in order to automatically annotate nonverbal behaviors across a corpus of sixty-three tutoring sessions. Multimodal feature vectors were constructed at one-second intervals, including facial expression, gesture, posture, the most recent task event, and whether the student was typing. These features were then used to predict post-session *engagement*, *frustration*, and *learning* outcomes. The results show that multimodal nonverbal behavior features are predictive of affect and learning beyond student incoming characteristics, such as self-efficacy and pretest scores.

These results are a first step toward understanding the relationship between affect, moment-by-moment nonverbal behavior, and task performance during tutoring. The multimodal data streams included nonverbal behavior (facial expression, gesture, posture) and task logs (discrete task events, typing status) across time intervals. This approach provides a basis for triangulating learner affect from multimodal time sequence data. The fine-grained data collected on task performance and nonverbal behavior provides an estimation of learners' underlying real-time cognitive and affective processes.

Further research may identify how facial expressions co-occur and provide further validation of fine-grained tracking of facial movements. Additionally, spatiotemporal features of gesture and posture have only just begun to be explored. Future work may disambiguate between different types of one-hand-to-face and two-hands-to-face gesture, as well as tracking more detailed postural information, such as slouching and leaning. Human tutors innately employ knowledge of nonverbal behavior, thus research in this vein brings the capabilities of automated tutorial intervention closer to those of human tutors. This line of investigation informs our understanding of learner affect and enables affective interventions that intelligently model nonverbal behavior and task actions, as a highly effective human tutor would.

ACKNOWLEDGMENTS

This work is supported in part by the North Carolina State University Department of Computer Science and the National Science Foundation through Grant DRL-1007962 and Grant CNS-1042468 (STARS Alliance). Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

REFERENCES

[1] Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K. and Christopherson, R.M. 2009. Emotion Sensors Go To School. *14th International Conference on Artificial Intelligence in Education* (2009), 17–24.

[2] Baker, R.S.J. d., D'Mello, S.K., Rodrigo, M.M.T. and Graesser, A.C. 2010. Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-Affective States during Interactions with Three Different Computer-Based Learning Environments. *International Journal of Human-Computer Studies*. 68, 4 (Apr. 2010), 223–241.

[3] Bloom, B.S. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher*. 13, 6 (1984), pp. 4–16.

[4] Bosch, N., D'Mello, S.K. and Mills, C. 2013. What Emotions Do Novices Experience during Their First Computer Programming Learning Session? *Proceedings of the 16th International Conference on Artificial Intelligence in Education* (2013), 11–20.

[5] du Boulay, B., Avramides, K., Luckin, R., Martinez-Miron, E., Mendez, G.R. and Carr, A. 2010. Towards Systems That Care: A Conceptual Framework based on Motivation, Metacognition and Affect. *International Journal of Artificial Intelligence in Education*. 20, 3 (2010).

[6] Chen, G., Gully, S.M. and Eden, D. 2001. Validation of a New General Self-Efficacy Scale. *Organizational Research Methods*. 4, 1 (2001), 62–83.

[7] Cooper, D.G., Muldner, K., Arroyo, I., Woolf, B.P. and Burleson, W. 2010. Ranking Feature Sets for Emotion Models used in Classroom Based Intelligent Tutoring Systems. *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization* (2010), 135–146.

[8] Craig, S.D., D'Mello, S.K., Witherspoon, A. and Graesser, A.C. 2008. Emote Aloud during Learning with AutoTutor: Applying

the Facial Action Coding System to Cognitive-Affective States during Learning. *Cognition & Emotion*. 22, 5 (2008), 777–788.

[9] D'Mello, S.K. and Calvo, R.A. 2011. Significant Accomplishments, New Challenges, and New Perspectives. *New Perspectives on Affect and Learning Technologies*. R.A. Calvo and S.K. D'Mello, eds. Springer. 255–271.

[10] D'Mello, S.K., Dale, R. and Graesser, A.C. 2012. Disequilibrium in the Mind, Disharmony in the Body. *Cognition & Emotion*. 26, 2 (2012), 362–374.

[11] D'Mello, S.K. and Graesser, A.C. 2010. Multimodal Semi-automated Affect Detection From Conversational Cues, Gross Body Language, and Facial Features. *User Modeling and User-Adapted Interaction*. 20, 2 (May 2010), 147–187.

[12] D'Mello, S.K., Lehman, B., Pekrun, R. and Graesser, A.C. 2012. Confusion Can Be Beneficial for Learning. *Learning & Instruction*. (2012).

[13] Ekman, P., Friesen, W. V. and Hager, J.C. 2002. *Facial Action Coding System*. A Human Face.

[14] Grafsgaard, J.F., Boyer, K.E., Phillips, R. and Lester, J.C. 2011. Modeling Confusion: Facial Expression, Task, and Discourse in Task-Oriented Tutorial Dialogue. *Proceedings of the 15th International Conference on Artificial Intelligence in Education* (2011), 98–105.

[15] Grafsgaard, J.F., Fulton, R.M., Boyer, K.E., Wiebe, E.N. and Lester, J.C. 2012. Multimodal Analysis of the Implicit Affective Channel in Computer-Mediated Textual Communication. *Proceedings of the 14th ACM International Conference on Multimodal Interaction* (2012), 145–152.

[16] Grafsgaard, J.F., Wiggins, J.B., Boyer, K.E., Wiebe, E.N. and Lester, J.C. 2013. Automatically Recognizing Facial Expression: Predicting Engagement and Frustration. *Proceedings of the 6th International Conference on Educational Data Mining* (Memphis, Tennessee, 2013).

[17] Grafsgaard, J.F., Wiggins, J.B., Boyer, K.E., Wiebe, E.N. and Lester, J.C. 2013. Automatically Recognizing Facial Indicators of Frustration: A Learning-Centric Analysis. *Proceedings of the 5th International Conference on Affective Computing and Intelligent Interaction* (2013), 159–165.

[18] Grafsgaard, J.F., Wiggins, J.B., Boyer, K.E., Wiebe, E.N. and Lester, J.C. 2013. Embodied Affect in Tutorial Dialogue: Student Gesture and Posture. *Proceedings of the 16th International Conference on Artificial Intelligence in Education* (Memphis, Tennessee, 2013).

[19] Hardy, M., Wiebe, E.N., Grafsgaard, J.F., Boyer, K.E. and Lester, J.C. 2013. Physiological Responses to Events During Training: Use of Skin Conductance to Inform Future Adaptive Learning Systems. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (2013), 2101–2105.

[20] Hart, S.G. and Staveland, L.E. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Human Mental Workload*. P.A. Hancock and N. Meshkati, eds. Elsevier Science. 139–183.

[21] El Kaliouby, R. and Robinson, P. 2005. The Emotional Hearing Aid: an Assistive Tool for Children with Asperger Syndrome. *Universal Access in the Information Society*. 4, 2 (Aug. 2005), 121–134.

[22] Kapoor, A. and Picard, R.W. 2005. Multimodal Affect Recognition in Learning Environments. *Proceedings of the 13th Annual ACM International Conference on Multimedia* (2005), 677–682.

[23] Kleinsmith, A. and Bianchi-Berthouze, N. 2012. Affective Body Expression Perception and Recognition: A Survey. *IEEE Transactions on Affective Computing*. (2012).

[24] Lee, D.M., Rodrigo, M.M.T., Baker, R.S.J. d., Sugay, J. and Coronel, A. 2011. Exploring the Relationship Between Novice Programmer Confusion and Achievement. *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction* (2011), 175–184.

[25] Lepper, M.R. and Woolverton, M. 2002. The Wisdom of Practice: Lessons Learned from the Study of Highly Effective Tutors. *Improving Academic Achievement*. J. Aronson, ed. Elsevier. 135–158.

[26] Littlewort, G., Bartlett, M.S., Salamanca, L.P. and Reilly, J. 2011. Automated Measurement of Children's Facial Expressions during Problem Solving Tasks. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition* (2011), 30–35.

[27] Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J.R. and Bartlett, M.S. 2011. The Computer Expression Recognition Toolbox (CERT). *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition* (2011), 298–305.

[28] Mahmoud, M. and Robinson, P. 2011. Interpreting Hand-Over-Face Gestures. *Proceedings of the International Conference on Affective Computing and Intelligent Interaction* (2011), 248–255.

[29] Marx, J.D. and Cummings, K. 2007. Normalized Change. *American Journal of Physics*. 75, 1 (2007), 87–91.

[30] O'Brien, H.L. and Toms, E.G. 2010. The Development and Evaluation of a Survey to Measure User Engagement. *Journal of the American Society for Information Science and Technology*. 61, 1 (2010), 50–69.

[31] Pekrun, R. 2006. The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice. *Educational Psychology Review*. 18, 4 (2006), 315–341.

[32] Picard, R.W., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., Machover, T., Resnick, M., Roy, D. and Strohecker, C. 2004. Affective Learning — A Manifesto. *BT Technology Journal*. 22, 4 (Oct. 2004), 253–269.

[33] Sanghvi, J., Castellano, G., Leite, I., Pereira, A., McOwan, P.W. and Paiva, A. 2011. Automatic Analysis of Affective Postures and Body Motion to Detect Engagement with a Game Companion. *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction* (2011), 305–311.

[34] Symonds, M.R.E. and Moussalli, A. 2010. A Brief Guide to Model Selection, Multimodel Inference and Model Averaging in Behavioural Ecology using Akaike's Information Criterion. *Behavioral Ecology and Sociobiology*. 65, 1 (Aug. 2010), 13–21.

[35] Whitehill, J., Serpell, Z., Foster, A., Lin, Y.-C., Pearson, B., Bartlett, M.S. and Movellan, J.R. 2011. Towards an Optimal Affect-Sensitive Instructional System of Cognitive Skills. *Proceedings of the Computer Vision and Pattern Recognition Workshop on Human Communicative Behavior* (Jun. 2011), 20–25.

[36] Wiebe, E.N., Lamb, A., Hardy, M. and Sharek, D. 2014. Measuring Engagement in Video Game-based Environments: Investigation of the User Engagement Scale. *Computers in Human Behavior*. 32, (Mar. 2014), 123–132.

[37] Wiebe, E.N., Williams, L., Yang, K. and Miller, C. 2003. Computer Science Attitude Survey. *North Carolina State University Technical Report TR-2003-1*. (2003).

[38] Woolf, B.P., Burleson, W., Arroyo, I., Dragon, T., Cooper, D.G. and Picard, R.W. 2009. Affect-Aware Tutors: Recognising and Responding to Student Affect. *International Journal of Learning Technology*. 4, 3-4 (2009), 129–164.

Sentiment Analysis in MOOC Discussion Forums: What does it tell us?

Miaomiao Wen
School of Computer Science
Carnegie Mellon University
5000 Forbes, Pittsburgh, PA
mwen@cs.cmu.edu

Diyi Yang
School of Computer Science
Carnegie Mellon University
5000 Forbes, Pittsburgh, PA
diyiy@cs.cmu.edu

Carolyn Penstein Rosé
School of Computer Science
Carnegie Mellon University
5000 Forbes, Pittsburgh, PA
cprose@cs.cmu.edu

ABSTRACT

Sentiment analysis is one of the great accomplishments of the last decade in the field of Language Technologies. In this paper, we explore mining collective sentiment from forum posts in a Massive Open Online Course (MOOC) in order to monitor students' trending opinions towards the course and major course tools, such as lecture and peer-assessment. We observe a correlation between sentiment ratio measured based on daily forum posts and number of students who drop out each day. On a user-level, we evaluate the impact of sentiment on attrition over time. A qualitative analysis clarifies the subtle differences in how these language behaviors are used in practice across three MOOCs. Implications for research and practice are discussed.

Keywords

Sentiment analysis, Opinion mining, Massive Open Online Course, MOOC, Forum posts

1. INTRODUCTION

Working towards improving MOOCs, it is important to know students' opinions about the course and also the major course tools. Based on opinions extracted from students' reviews, previous work illustrates that the most important factor to students is who is teaching the course [1]. However, for a given MOOC that will be offered again by the same instructor team, it is more critical to know what can be improved in the course. Recent research on social media use has demonstrated that sentiment analysis can reveal a variety of behavioral and affective trends. For example, collective sentiment analysis has been adopted to find the relationship between Twitter mood and consumer confidence, political opinion [20], and stock market fluctuations [5]. Course forums provide students with the chance to engage in social learning in MOOCs [6]. Analyzing the data from this part of the course, we can infer important information about attitudes prior to and even in the absence of post-course surveys [31]. The contribution of this paper is an investigation into what sentiment analysis can tell us about the students' opinions towards the course. We also analyze the impact of sentiment on attrition over time in MOOCs using a survival modeling technique.

Despite the great potential, the current generation of MOOCs has so far failed to produce evidence that the potential is being realized. Of particular concern is the extremely high rate of attrition that has been reported. Much of this research focuses specifically on summative measures of attrition. They seek to identify factors that predict completion of the course, for example, by conducting correlational analysis between course completion and click stream evidence of engagement with course activities [12]. However, what we see is that attrition happens over time. While a large proportion of students who drop out either fail to engage meaningfully in the course materials at all or drop out after the first week of participation, a significant proportion of students remain in the course longer than that but then drop out along the way. This suggests that there are students who are struggling to stay involved. Supporting the participation of these struggling students may be the first low hanging fruit for increasing the success rate of these courses. Before we can do so, we need to understand better their experience of participation along the way as they struggle and then ultimately drop out. Thus, in this paper we employ a survival modeling technique to study various factors' impact on attrition over course weeks.

As a reflection of student experience communicated through their posts, we investigate sentiment expressed in course forum posts. While the association between sentiment with summative course completion has been evaluated in prior work [24], and while the impact of other linguistic measures and social factors on attrition over time has been published as well[31, 26], this is the first work we know of that has brought this lens to explore what sentiment can tell us about drop out along the way in this type of environment. In particular, we explore this connection across three MOOCs in order to obtain a nuanced view into the ways in which sentiment is functioning similarly and differently or signaling similar and different things across these three courses. Our goal is for this analysis to reflect some of the flexibility in how these linguistic constructs are used in practice in order to inform application of such techniques in future analysis in this community.

In the remainder of the paper, we begin by describing our dataset and discussing related work. Next, we explain how a collective sentiment analysis can reflect students' attitudes towards the course and course tools. In light of the collective sentiment analysis, we continue with a survival analysis that shows what sentiment can tell us about drop out along the way in MOOC environments. Finally, we conclude with a summary and possible future work.

2. COURSERA DATASET

The data used for the analysis presented here was extracted from three courses by permission from Coursera.org using a screen scraping protocol. The three courses cover a wide range of topics. Our dataset consists of three courses: one social science course, *Accountable Talk: Conversation that works*[1], offered in October 2013. We refer to this course as the *Teaching* course; one literature course, *Fantasy and Science Fiction: the human mind, our modern world*[2], offered in June 2013. We refer to this course as the *Fantasy* course; one programming course, *Learn to Program: The Fundamentals*[3], offered in August 2013. We refer to this course as the *Python* course. Statistics about the three courses are listed in Table 1.

3. RELATED WORK

3.1 Sentiment Analysis for Social Media

Affect mined from Facebook and Twitter posts is known to be reflective of public behavior and opinion trends [20, 5]. The results generated via the analysis of collective mood aggregators are compelling and indicate that accurate public mood indicators can be extracted from online materials. Sentiment analysis has been used as an invaluable tool for identification of markers of affective responses to crisis [10], as well as depression [9], anxiety, and other psychological disorders [8] from social media sites. Using publicly available online data to perform sentiment analyses requires far less cost in terms of effort and time than would be needed to administer large-scale public surveys and questionnaires. Most MOOCs offer course forums as a communication and learning tool. While only a small percentage of students actively participate in the threaded discussions, if course instructors can use automated analysis of those posts as a probe that indicates whether things are going well in the course, and the analysis reveals something about what the issues are, they will be better prepared to intervene as necessary.

3.2 Sentiment Analysis for Educational Data Mining

Mackness et al. [15] posed the question of how to design a MOOC that can provide participants with positive experiences. Most of the prior work that addresses this question involved conducting surveys and interviews [25, 2]. In contrast, in some prior E-learning research, automatic text analysis, content analysis and text mining techniques have been used to mine opinions from user-generated content, such as reviews, forums or blogs [27, 4, 11]. Attitude is important to monitor since learners with a positive attitude have been demonstrated to be more motivated in E-learning settings [18]. Correspondingly, previous work reveals that boredom was associated with poorer learning and problematic behavior. In contrast, frustration was less associated with poorer learning [3]. Based on user-generated online textual reviews submitted after taking the courses, Adamopoulos [1] has applied sentiment analysis to collect students' opinions towards MOOC features such as the course characteristics and university characteristics. In that work, the goal was to determine which factors affect course completion, it is also important to address the related but different question of what can be improved when the course is offered again.

MOOC	Active Users	Total Days	Total Posts	Avg. Posts Per Day
Teaching	1,146	53	5,107	96
Fantasy	771	43	6520	152
Python	3,590	49	24963	510

Table 1: Statistics of the three Coursera MOOCs. Active users refer to those who post at least one post in a course forum.

Given the recent work on MOOC user dropout analysis, very little has attempted finer-grained content analysis of the course discussion forums. Brinton et al. [6] identified high decline rate and high-volume, noisy discussions as the two most salient features of MOOC forum activities. Ramesh et al. [24] use sentiment and subjectivity of user posts to predict engagement/disengagement. However, neither sentiment nor subjectivity was strongly predictive of engagement in that work. One explanation is that engaged learners also post content with negative sentiment on the course, such as complaints about peer-grading. Thus, the problem is more complex than the operationalization used in that work. Taking the analysis a step further to explore such nuances is the goal of this paper.

4. METHOD

This work reflects on how sentiment analysis can be useful in a MOOC context. On the course-level, we use collective sentiment analysis, which has been successfully applied in many social media investigations, to explore the relation between opinions expressed by students and the students' dropout rate. To help MOOC instructors collect students' opinions towards various course tool designs, we extract the positive and negative sentiment words that are associated most with the course tool topic keywords. In order to understand the impact of sentiment on the user-level, we adopt survival analysis to to examine how sentiment that members have expressed and are exposed to in a particular week predicted their continued participation in the forum discussion.

4.1 Course-level Sentiment Analysis: Collective Sentiment Analysis

In this section we first describe how we use collective sentiment analysis to study students' attitudes towards the course and course tools based on forum posts. To improve MOOC design, it is important to obtain feedback from students. Most MOOCs conduct post-course surveys where students' opinions towards the course are elicited. However, only a very limited portion of students who registered for the course will actually fill out the survey. A discussion forum is

[1] https://www.coursera.org/course/accountabletalk
[2] https://www.coursera.org/course/fantasysf
[3] https://www.coursera.org/course/programming1

MOOC	Course	Lecture	Assignment	Peer-assessment
Teaching	820	725	904	97
Fantasy	731	327	2515	375
Python	1430	2492	3700	-

Table 2: Number of course tool-related posts in the three courses' forums. The Python course did not implement peer-assessment.

a natural place where students convey their satisfaction or dissatisfaction with the course. Can we analyze forum posts to infer students' opinions towards the course in the same manner that post-course surveys elicit such feedback from the students? If so, then tracking students' opinion based on daily forum post content could be a far more timely alternative to post-course surveys, and may provide a less biased view of the course because it would have the opportunity to capture the attitude of students who drop out before the post-course survey is administered.

Sentiment polarity analysis techniques applied to individual messages may make many errors, partly due to the extent to which the text is taken out of context. However, with a large number of such measurements aggregated together, the errors might cancel, and the resulting composite indicator may be a more faithful indicator of public opinion. In previous work on text-based social media sites, summary statistics derived from similarly simple sentiment analysis are demonstrated to correlate with many objective measures of population level behaviors and opinions [20, 21]. In this section, we use sentiment analysis to understand students' opinion towards the course and course tools.

4.1.1 Setting

To extract students' aggregate opinions on a topic, we need to first identify posts relating to a topic (*post retrieval step*). Then we need to estimate the opinion of these posts (*opinion estimation step*). Following the same methodology used in previous work [20], in the *post retrieval step*, we only use messages containing a topic keyword. To decide which topic keywords to use for each course tool of interest, we run a distributional similarity technique called Brown clustering [7, 13] on all three courses' posts in order to identify clusters of words that occur in similar contexts. It is conceptually similar to Latent Semantic Analysis, but is capable of identifying finer grained clusters of words. From the results, we construct keyword lists for topics by starting with hand-selected keywords, and then finding the clusters that contain those words and manually choosing the words that are human-identified as being related to the same topic. The numbers of posts retrieved for each topic are shown in Table 2.

- For *Course* topic, we use "the course", "this course", "our course" and the name of each MOOC.
- For *Lecture* topic, we use "lecture" and "video".
- For *Assignment* topic, we use "assignment", "essay", "reading" and "task".
- For *Peer-assessment* topic, we use "peer assessment", "peer grading", "assess your peer", "peer score", "peer feedback", "peer review" and "peer evaluation".

In the *opinion estimation step*, for each set of posts that are related to a topic, we define the **topic sentiment ratio** x_t on day t as the ratio of positive versus negative words used in that day's post set. Positive and negative terms used in this paper are defined by the sentiment lexicon from [14], a word list containing about 2,000 and 4,800 words marked as positive and negative, respectively.

$$x_t = \frac{Total\, poistive\, terms}{Total\, negative\, terms}$$

Day-to-day, the topic sentiment ratio rapidly rises and falls each day. In order to derive a more consistent signal, and following the same methodology used in previous work [20], we smooth the sentiment ratio with one of the simplest possible temporal smoothing techniques, a moving average over a window of the past k days:

$$MA_t = \frac{1}{k}(x_{t-k+1} + x_{t-k+2} + ... + x_t)$$

The moving average of sentiment ratio MA_t is our estimation of collective opinion expressed by the students in the course forum during day t.

4.1.2 Results

Part 1. Opinion towards the course

In this section, we explore the correlation between collective opinions mined from the forum posts and objective measures related to students' actual opinions.

To objectively measure students' opinions towards the course, we count how many students drop out of the course each day based on the students' login information. Here we consider that a student drops the course on day t if the student's last login date of the course is on day t. There are many students who just register for a course to see what it is about, without serious intention of actually taking the course. The number of students who drop out in the first week is much larger than the other weeks. We calculate the correlation between the number of users who drop the course and the Course sentiment ratio (MA_t) starting from course week 2 until the last course day in order to avoid the analysis being muddied by the very different factors that affect dropout in the first week.

In the Teaching course we can operationalize drop out in two different ways because we have both forum data and login data. In the other two courses, we have only forum data. Thus, we are able to measure the correlation between sentiment and dropout two different ways in the Teaching course, which enables us to understand how these two operationalizations may reveal different patterns, and then we can use that understanding to interpret what we find in the other two courses.

First we explore how sentiment shifts over the seven weeks of each course. In Figure 1, we show how the number of students who drop out and the Course topic sentiment ratio vary from day-to-day. In all three courses, Course sentiment ratio is much higher during the last course week.

		Course	Lecture	Assignment	Peer-assessment
Teaching	**Pos**	incredibly,benefits enjoyment,richer,greatest	incredibly,benefits,richer guarantee,gaining	substantive,benefits rich,soft,gaining	smart,kudos,praise prominent,mastery
	Neg	missed,negative,low-rated taxing,superficial	breaking,worry,missed challenging,thoughtless	struggles,taxing,poor struggled,unacceptable	riled,worry,missed challenging,conflict
Fantasy	**Pos**	avidly,incredibly,substantive benefits,guarantee	incredibly,kudos,smart beauteous,consistent	masterfully,avidly,substantive guarantee,admiration	consistent,benefits,richer competitive,richer,balanced
	Neg	damnation,lie,missed belated,negative	lie,breaking,wrong anxious,worry	shortcomings,confuse creeping,menace,flaky	negative,frustrating,wrong invasive,hate
Python	**Pos**	self-satisfaction,impresses providence,kudos,smart	remedy,convenient,merit gaining,smartest	incredibly,guarantee,richer benefits,proud	- -
	Neg	forbidden,unforeseen,worry breaking,challenging	embarrassing,worry, challenging,missed	unforeseen,confuse,bug swamped,shock	- -

Table 3: Sentiment words associated most with each course tool.

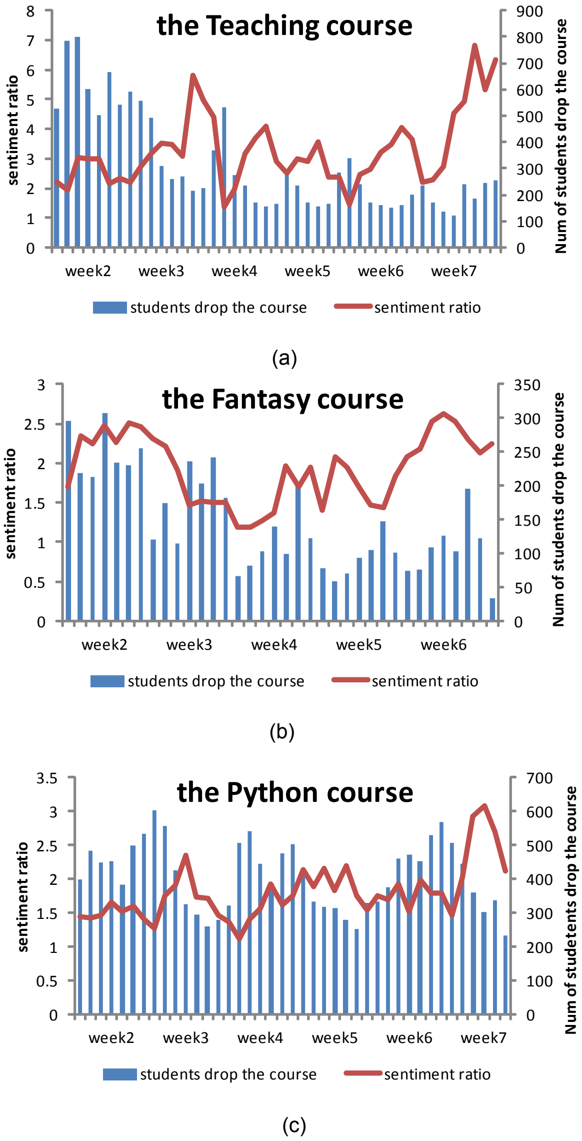


Figure 1: Moving average MA_t of Course topic sentiment ratio and number of students who drop out over course weeks. Window size k equals 3.

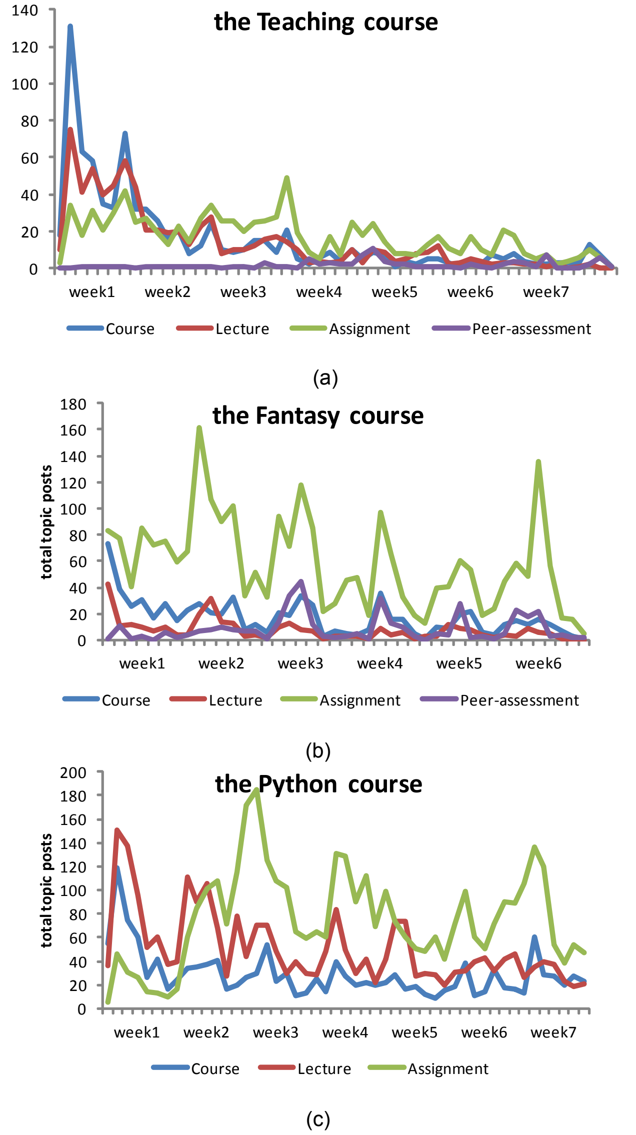


Figure 2: Trends of total number of course tool related posts.

Because many students post "thank you" and positive feedback towards the course during the last course week. First we consider a student to drop out from the course forum if the student posts his/her last post on day t. Across all three courses, we observe a trend in the expected direction, namely that higher sentiment ratios are associated with fewer dropouts, which is significant in two out of the three courses(r = -0.25, $p < 0.05$ for the Teaching course; r = -0.12 for the Fantasy course(Figure 1(b)); r = -0.43, $p < 0.01$ for the Python course(Figure 1(c))). In the Teaching course, where we can determine dropout more precisely from login information, we see a stronger correlation(r = -0.39, $p < 0.01$). This analysis serves as a validation that the collective sentiment extracted from these posts can partly reflect the opinion of the entire student population towards the course.

Part 2. Opinion towards the course tools

As important components of the course, the course tools have a big impact on a student's experience in a MOOC. For example, peer-assessment serves as a critical tool for scaling the grading of complex, open-ended assignments to MOOCs with thousands of students. But it does not always deliver accurate results compared to human experts [22]. In the Fantasy course, we see heated discussions about peer-assessment during course week 3 and week 4 when peer-assessment was conducted. One discussion thread with the title "Why I'm dropping out (the fundamental issue is the peer grading)" got more than 50 comments and many students expressed their opinions after receiving their peer grades in that thread.

Though one could be generally happy about the course, he/she might not be satisfied with a certain course tool. In many MOOCs' post-course surveys, students are required to separately rate the course tools such as lecture, assignment and peer-assessment. Then the instructors are able to obtain summative feedbacks on various course components. In the course discussion forums, students naturally express their opinions towards these course tools. We show the total number of related posts for each course in Table 2. It is impossible for course instructors to read hundreds or even thousands of potentially related-posts. In this session, we try to extract the most prominent opinions associated with each course tool from these posts.

In Figure 2, we show the number of topic-related posts on each course day. Across the three courses, all topics have a weekly cyclical structure, occurring more frequently on weekdays, especially in the middle of the week, compared to weekends. Talk about assignments is the most frequent topic since TAs post weekly discussion threads for students to discuss assignments.

For each course tool, we extract the positive and negative sentiment words that associate most frequently with the course tool topic keywords. We rank the sentiment words by the Pointwise Mutual Information (PMI) [16] between the word and the topic keyword:

$$PMI_{w,TopicKeyword} = \frac{P(w, TopicKeyword)}{P(w)P(TopicKeyword)}$$

Where $P(w, keyword)$ is the probability of the sentiment word w and a topic keyword appears in the same post; $P(w)$ is the probability of a sentiment word w appears in a post; $P(TopicKeyword)$ is the probability that at least one topic keyword appears in a post. We show the top five sentiment words that are most frequently associated with the course tool topic keywords in Table 3. We can see that some of the words are wrongly identified as sentiment words, such as "lie", "missed" and "soft". From the table we can identify some of the merits and problems of a course tool. These representative sentiment words can complement the rating obtained from the post-course survey.

4.2 User-level Sentiment Analysis: Survival Analysis

In the previous section, we measured sentiment and dropout on a course-level. Our goal in this section is to understand how expression of sentiment relates to attrition over time in the MOOCs on a user-level. We apply survival analysis to test if students' attitudes as expressed in their posts or the ones they are exposed to correlate with dropout from the forum discussion. Recent work has questioned the impact of course forum flaming on students in MOOC courses [29]. We explore sentiment as it relates to an individual's own posts during a week as well as to the other posts that appear on the same thread as that individual's posts during the same period of time. While we cannot be sure that the student read all and only the other posts appearing on the same threads where that student posted during a week, this provides a reasonable proxy for what conversational behavior a student was exposed to within a period of time in the absence of data about what students have viewed. A similar approach was used in a previous analysis of social support in an online medical support community [30].

4.2.1 Survival Analysis Setting

In our survival model, the dependent measure is **Dropout**, which is 1 on a student's last week of active participation unless it is the last course week (i.e. the seventh course week), and 0 on other weeks. In our sentiment analysis, we separate the measure of positivity and negativity rather than operationalizing them together as a single scale. For each week, we measure a student's expressed sentiment to see if the sentiment a student expressed in his/her posts is correlated with drop out. To study if a student would be influenced by the sentiment expressed in their peers' posts, we measure the amount of sentiment a student is exposed to during that week.

In our data, we find across the three courses correlations with R value less than .13 between a measure of positivity and of negativity. Thus, we separate these measures and evaluate them separately in our survival models.

Individual Positivity (Indiv. Positivity): average positivity in the user's posts that week

$$Indiv.\ Positivity = \frac{Total\ positive\ terms}{Total\ number\ of\ words}$$

Individual Negativity (Indiv. Negativity): average negativity in the user's posts that week

$$Indiv.\ Negativity = \frac{Total\ negative\ terms}{Total\ number\ of\ words}$$

Thread Positivity: this variable measures the average positivity a user was exposed to in a week. It was calculated by dividing the total number of positive words in the threads in a week where the user had posted by the total number of words in those threads.
Thread Negativity: this variable measures the average negativity a user was exposed to in a week. It was calculated by dividing the total number of negative words in the threads in a week where the user had posted by the total number of words in those threads.

4.2.2 Modeling

Survival analysis is a statistical modeling technique used to model the effect of one or more indicator variables at a time point on the probability of an event occurring on the next time point. In our case, we are modeling the effect of certain language behaviors (i.e., expression or exposure to expression of sentiment) on probability that a student drops out of the forum participation on the next time point. Survival models are a form of proportional odds logistic regression, and they are known to provide less biased estimates than simpler techniques (e.g., standard least squares linear regression) that do not take into account the potentially truncated nature of time-to-event data (e.g., users who had not yet ceased their participation at the time of the analysis but might at some point subsequently). In a survival model, a prediction about the probability of an event occurring is made at each time point based on the presence of some set of predictors. The estimated weights on the predictors are referred to as hazard ratios. The hazard ratio of a predictor indicates how the relative likelihood of the failure (in our case, student dropout) occurring increases or decreases with an increase or decrease in the associated predictor. A hazard ratio of 1 means the factor has no effect. If the hazard ratio is a fraction, then the factor decreases the probability of the event. For example, if the hazard ratio was a number n of value .4, it would mean that for every standard deviation greater than average the predictor variable is, the event is 60% less likely to occur (i.e., 1 - n). If the hazard ratio is instead greater than 1, that would mean that the factor has a positive effect on the probability of the event. In particular, if the hazard ratio is 1.25, then for every standard deviation greater than average the predictor variable is, the event is 25% more likely to occur (i.e., n - 1).

4.2.3 Quantitative Analysis

Intuitively, we might expect that positive sentiment indicates that students are enjoying or benefitting from a course whereas negative sentiment might indicate that a student is frustrated with a course. The results from our quantitative analysis are not consistent with our intuition. A qualitative analysis of how these features play out across the three courses, which is provided in Section 5, will offer a more nuanced view.

A summary of the results of the survival analysis are presented in Table 4. Typically, we observe lexical accommodation in discussions, including threaded discussion forums [19]. Consistent with this, we find low but significant correlations between individual level sentiment scores and thread level sentiment scores. The correlations are low enough that they are not problematic with respect to including these variables together within the survival models. Including them together allows us to compare the effect of a student's behavior with the effect of exposure to other students' behavior. As we see in Table 4, not only do we see differential effects across courses, we also see differential effects between behavior and exposure.

Indep. Variable	**Teaching**	**Fantasy**	**Python**
Indiv. Positivity	1.03	0.97	1.04*
Indiv. Negativity	0.99	0.84**	1.05**
Thread Positivity	0.95	0.99	1.02
Thread Negativity	1.06*	0.82**	0.98

Table 4: Hazard ratios of sentiment variables in the survival analysis(*: p<0.05, **: p<0.01).

Specifically, in the Python course, we see a significant association between both positive and negative expression and student dropout. In particular, students who express a standard deviation more positive emotion than average are 4% more likely to drop out of the course by the next time point than students who express an average level of positive emotion. Similarly, students who express a standard deviation more negativity than average are 5% more likely to drop out by the next time point than students who express an average amount of negative emotion. Exposure to emotion makes no significant prediction about dropout in this course.

In the Fantasy course, the pattern is different. Negative emotion, whether expressed by an individual or present on the threads that student participated in, is associated with less attrition. In particular, students who either express a standard deviation more negativity or are exposed to a standard deviation more negativity on a time point are nearly 20% less likely to drop out on the next time point than students who express or are exposed to an average amount of negativity. However, positivity has no significant effect.

In the Teaching course, again the pattern is different. There is no effect of expressing negativity or positivity. But students who are exposed to a standard deviation more negativity are 6% more likely to drop out on the next time point than students who are exposed to an average amount of positivity.

We might expect that differences in effect might be related to differences in norms of behavior between courses. For example, positivity and negativity might have more of an effect where they are unusual. However, while one important difference across courses is the average level of positivity and negativity that is present in the discussions, this pattern is not consistent with what we would expect if differences in behavioral norms was the explanation for the differences in effect. The qualitative analysis in the discussion section will again elucidate some potential explanations for differences in behavioral norms.

We also tried to measure individual and thread sentiment based on topic post set retrieved in Section 4.1.1. However, as users typically have too few posts that contain a topic keyword in each course week, the positivity/negativity scores are not available for most of the users. So the survival analysis results on each topic post set might not be meaningful.

5. DISCUSSION: QUALITATIVE ANALYSIS

The results of the survival analysis were not completely consistent either across courses or with an initial naive expectation. In this section, we elucidate those quantitative results with qualitative analysis.

In the Fantasy course, negative comments were ones where people are describing some characters in the fiction. They use some strong negative words which should be very rare in usual conversation, such as "destroy, "devil", "evil", "wicked", "death", "zombie", "horror", etc. One example post is shown below, the negative words are underlined. These messages got high negativity scores because of words taken out of context, which seemed to happen more for negative words than positive ones. The negative word use in this course is actually a sign of engagement because messages with negative words are more likely to be describing science fantasy related literature or even posting their own essay for suggestions.

- Indiv. Negativity = 0.23, the Fantasy course
  "The Death Gate Cycle was such a haunting story!"

In the Python course, forum posts are mostly problem-solving. Both very positive and very negative messages are predictive of more dropout. The most positive messages were thanking for a response. E.g. "Cool!" or "Thank you!". Messages rated as very negative were mainly reporting problems in order to get help or commiserate on an issue already posted. E.g. "It's my error." or "Same problem here." Users who post messages like this are more in the role of forum content consumer. They may only be browsing the forum to look for answers for their particular problems without the intention of contributing content, such as solving the other's problems.

The Teaching course was a social science course about good communication skills. In that course, most forum posts are discussing course-related concepts and techniques. Nevertheless, negativity was low on average, perhaps because the community had a higher politeness norm. A lot of messages contain negative words because of discussion about problem solving. One example post is shown below. It is important to note that in this course, discussion of problems takes on a different significance than in the Python course because changing your interpersonal practices takes time. Whereas in Python you can get your question answered and move on, when it comes to behavior change, discussion of such personal questions signals more intimacy.

- Indiv. Negativity = 0.22, the Teaching course
  A lot of people got crushed by their overloaded work pressure, so why bother yourself talking so complex, complicated, irrelevant and non-rewarding topics while you can spare yourself in those funny little talks and relax a little.

The important take home message here is that the explanation for the pattern goes beyond simple ideas about sentiment and what it represents. We see that expressions of sentiment are being used in different kinds of contexts to serve different functions, and thus this operationalization of attitude is not picking up on the same things across the three courses. With respect to sentiment, we cannot afford to make intuitive assumptions about what it means when variables related to sentiment achieve high predictive value in models that predict choices that people make.

6. LIMITATIONS

We use a relatively simple sentiment detector to explore the uses of sentiment analysis in MOOC context. The sentiment lexicon we utilize is designed for predicting sentiment polarity of product reviews. Creating a more comprehensive lexicon specifically for a MOOC context could improve the system [23]. We associate the opinion to a topic term coexisting in the same context. If we have enough posts with annotated sentiment and topic, many machine learning approaches could capture the mixture of document topics and sentiments simultaneously and substantially improve the accuracy of opinion tracking [17, 28].

7. CONCLUSIONS AND IMPLICATIONS FOR PRACTICE

In this paper, we utilize sentiment analysis to study drop out behavior in three MOOCs. Using a simple collective sentiment analysis, we observe a significant correlation between sentiment expressed in the course forum posts and the number of students who drop the course. Through a more detailed survival analysis, we did not observe consistent influence of expressed sentiment or sentiment a student is exposed to on user dropout. This analysis suggests that sentiment analysis should be used with caution in practice, especially when the texts are very noisy and limited in quantity. However, we see that within a specific course, the relationship between sentiment and dropout makes sense once one examines practices for expressing sentiment within that specific course context. Thus, reports of sentiment could be valuable if they also provide users with examples of how the sentiment words are typically used in that course.

8. ACKNOWLEDGMENTS

We want to thank Ryan Carlson, David Adamson and Tanmay Sinha, who helped provide the data for this project. The research reported here was supported by National Science Foundation grant 0835426.

9. REFERENCES

[1] P. Adamopoulos. What makes a great mooc? an interdisciplinary analysis of student retention in online courses. In *Proceedings of the 34th International Conference on Information Systems, ICIS*, volume 2013, 2013.

[2] C. e. a. Alario-Hoyos. Analysing the impact of built-in and external social tools in a mooc on educational technologies. In *Scaling up learning for sustained impact*, pages 5–18. Springer, 2013.

[3] R. S. Baker, S. K. D'Mello, M. M. T. Rodrigo, and A. C. Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learnersŠ cognitive–affective states during interactions with three different computer-based learning environments.

International Journal of Human-Computer Studies, 68(4):223–241, 2010.

[4] H. H. Binali, C. Wu, and V. Potdar. A new significant area: Emotion detection in e-learning using opinion mining techniques. In *Digital Ecosystems and Technologies, 2009. DEST'09. 3rd IEEE International Conference on*, pages 259–264. IEEE, 2009.

[5] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[6] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, and F. M. F. Wong. Learning about social learning in moocs: From statistical analysis to generative model. *arXiv preprint arXiv:1312.2159*, 2013.

[7] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[8] M. De Choudhury, S. Counts, and E. Horvitz. Major life changes and behavioral markers in social media: case of childbirth. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1431–1442. ACM, 2013.

[9] M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz. Predicting depression via social media. In *AAAI Conference on Weblogs and Social Media*, 2013.

[10] M. De Choudhury, A. Monroy-Hernandez, and G. Mark. "narco" emotions: Affect and desensitization in social media during the mexican drug war. In *CHI*. ACM, 2014.

[11] A. El-Halees. Mining opinions in user-generated contents to improve course evaluation. In *Software Engineering and Computer Systems*, pages 107–115. Springer, 2011.

[12] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 170–179. ACM, 2013.

[13] P. Liang. *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology, 2005.

[14] B. Liu. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:627–666, 2010.

[15] J. Mackness, S. Mak, and R. Williams. The ideals and reality of participating in a mooc. In *Networked Learing Conference*, pages 266–275. University of Lancaster, 2010.

[16] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[17] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM, 2007.

[18] J. Moshinskie. How to keep e-learners from e-scaping. *Performance Improvement*, 40(6):30–37, 2001.

[19] A. Nenkova, A. Gravano, and J. Hirschberg. High frequency word entrainment in spoken dialogue. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 169–172. Association for Computational Linguistics, 2008.

[20] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11:122–129, 2010.

[21] M. J. Paul and M. Dredze. You are what you tweet: Analyzing twitter for public health. In *ICWSM*, 2011.

[22] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. Tuned models of peer assessment in moocs. *EDM*, 2013.

[23] G. Qiu, B. Liu, J. Bu, and C. Chen. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27, 2011.

[24] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor. Modeling learner engagement in moocs using probabilistic soft logic. In *Workshop on Data Driven Education, Advances in Neural Information Processing Systems 2013*, 2013.

[25] C. O. Rodriguez. Moocs and the ai-stanford like courses: Two successful and distinct course formats for massive open online courses. *European Journal of Open, Distance and E-Learning*, 2012.

[26] C. Rosé, R. Carlson, D. Yang, M. Wen, L. Resnick, P. Goldman, and J. Sheerer. Social factors that contribute to attrition in moocs. In *ACM Learning at Scale*, 2014.

[27] D. Song, H. Lin, and Z. Yang. Opinion mining in e-learning system. In *Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on*, pages 788–792. IEEE, 2007.

[28] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120. ACM, 2008.

[29] Y. Wang and R. Baker. Mooc learner motivation and course completion rate. *MOOC Research Initiative Conference*, 2013.

[30] Y.-C. Wang, R. Kraut, and J. M. Levine. To stay or leave?: the relationship of emotional and informational support to commitment in online health support groups. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 833–842. ACM, 2012.

[31] M. Wen, D. Yang, and C. Rosé. Linguistic reflections of student engagement in massive open online courses. *In International AAAI Conference on Weblogs and Social Media*, 2014.

The Effect of Mutual Gaze Perception on Students' Verbal Coordination

Bertrand Schneider
Stanford University
schneibe@stanford.edu

Roy Pea
Stanford University
roypea@stanford.edu

ABSTRACT

In a previous study, we found that *real-time mutual gaze perception* (i.e., being able to see the gaze of your partner in real time on a computer screen while solving a learning task) had a positive effect on students' collaboration and learning [8]. The goals of this paper are to: 1) explore a variety of computational techniques for analyzing the transcripts of students' discussions; 2) examine whether any of those measures sheds new light on our previous results; and 3) test whether those metrics have any predictive power regarding learning outcomes. Using various natural language processing algorithms, we found that linguistic coordination (i.e., the extent to which students mimic each other in terms of their grammatical structure) did not predict the quality of student collaboration or learning gains. However, we found that the *coherence* of students' discourse was significantly different across our experimental conditions; this measure was positively correlated with their learning gains. Finally, using various language metrics, we were able to roughly (i.e., using a median-split) predict learning gains with a 94.4% accuracy using Support Vector Machine. The accuracy dropped to 75% when we used our model on a validation set. We conclude by discussing the benefits of using computational techniques on educational datasets.

Keywords

Natural Language Processing; Eye-tracking; Learning Analytics; Computer-Supported Collaborative Learning.

1. INTRODUCTION

Despite recent efforts in developing automated ways to analyze students' discourse, most educational researchers still rely on traditional tools to analyze transcripts from students. Traditional methods include time-consuming qualitative analyses and the development of manual coding schemes. The field of Natural Language Processing (NLP) has significantly grown and gained in maturity over the past decades, and computational techniques can now be advantageously applied to educational datasets. Recent efforts in topic modeling, for instance, seem to be especially promising in terms of gaining insights into students' discourse and cognitive processes [9]. Unfortunately, social scientists willing to learn those tools are a rare breed, and multi-disciplinary work is slow to appear between educational researchers and computer scientists. In this paper, we describe our attempt at applying NLP techniques to educational transcripts.

2. THE CURRENT DATASET

In a previous work [8], we conducted a study on the effect of *mutual visual gaze perception* on students' collaborative problem-solving processes. In this experiment, student dyads were asked to remotely collaborate on a set of diagrams to discover how the human brain processes visual information. Each student was located in a different room, and could communicate with his/her partner via an audio channel. The information on the screen was similar for both participants (i.e., the brain diagrams shown in Fig. 1). The structure of the activity was as follows: in the first step, students analyzed brain diagrams (12 minutes); in a second step, they were asked to read a textbook chapter about human vision and discuss their understanding of this topic (12 minutes). Finally, before the analysis activity and after the reading task, students were asked to complete a learning test (pre/post-questionnaires).

Half of our participants were assigned to an experimental group ("visible-gaze") where they could see the gaze of their partner displayed in real time on a screen. To achieve this, we used two Tobii X1 eye-trackers running at 30Hz which recorded students' gaze. In a control group ("no-gaze"), the other half of our participants did not have access to this visualization. This intervention helped students in the first group achieve higher learning gains (Fig. 2) and a higher quality of collaboration (as measured by [4]).

We also recorded students' gaze movements and their collaborative discourse. Interestingly, by analyzing the eye-tracking data we found that participants in the experimental condition had more moments of joint attention (i.e., they were more likely to be looking at the same diagram at the same time on the screen), and this measure was significantly correlated with positive learning gains. This result reinforced the assumption that joint visual attention is a crucial mechanism for coordinating social interactions [10].

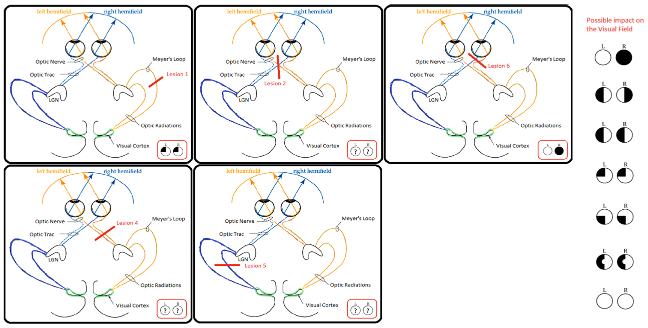


Figure 1: Diagrams students had to analyze. Five contrasting cases show the visual pathways of the human brain; students had to identify the effect of each lesion on the visual field.

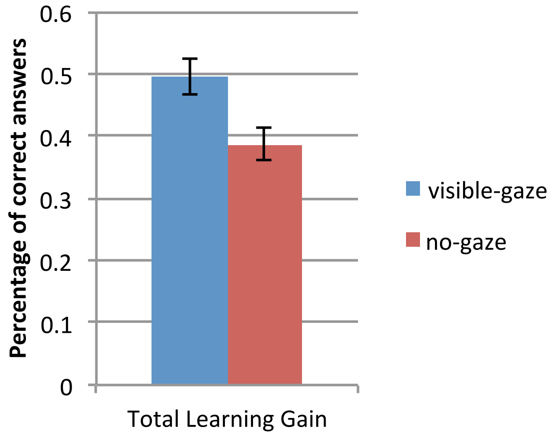


Figure 2: Learning gains for the two experimental groups of the study (p < .01).

In a subsequent analysis, we also suggested that our intervention helped students because: 1) they were able to anticipate what their partner was about to say, because *they could already see the location of their partner's gaze on the screen*; 2) *they could use gaze as a pointer to complement their discourse*, and thus remove the need to explicitly mention locations on the diagrams; and finally, 3) they could monitor the visual activity of their partner at all times, providing an aid to establishing a common ground.

We propose to use computational techniques to further illuminate this dataset. More specifically, we are interested in exploring three aspects of students' dialogues:

1. Are there ways to characterize the effect of our intervention on students' discourse?
2. Is it possible to find markers of productive learning trajectories?
3. Is it possible to find markers of constructive collaborations?

Technically, we can answer the first question by designing linguistic metrics and running statistical tests (i.e., ANOVA) between our two experimental conditions. The second and third questions can be answered by running correlations between our measures of interest, learning gains and collaboration scores.

3. NATURAL LANGUAGE PROCESSING AND MUTUAL GAZE PERCEPTION

In the next sections, we describe the measures used to provide a preliminary answer to those questions. First, we looked at unigrams, bigrams and trigrams counts to build categories of interest using a bag of words model. Next, we looked at the coordination of linguistic styles among students: are students more likely to mimic the grammatical structure of their peers in a good collaboration (as suggested by [2])? We then assessed the *coherence* of students' discourse, by comparing the similarity of consecutive sub-sections of the transcripts; our goal was to evaluate the extent to which students were building on each other's ideas during the task. Finally, we gathered all the previous measures and ran a machine-learning algorithm (Support Vector Machine) to roughly predict students' learning gains.

3.1 N-GRAMS

To get a sense of our dataset, we first computed unigram, bigram and trigram probabilities. This helped us understand which words were frequently used in our two experimental groups, and allowed us to build relevant categories for grouping our n-grams. For instance, we observed that the word "look" was positively correlated with learning gains (r(37) = 0.42, p = 0.008), which can be associated with either the content to be learned (i.e., the brain diagrams showed how visual information is processed by the human brain) or a verbal indication to share visual information (e.g., "look at my gaze!"). However, we did not conduct in-depth analyses of the unigrams alone, because they were difficult to interpret: unigrams are often ambiguous (see the example above), and bigrams or trigrams are usually so rare that they don't provide strong evidence for any type of hypothesis. This is why we decided to group them by categories instead of analyzing them in isolation. As a first pass, we decided to create those categories based on common sense: a researcher looked at the 200 most common words and manually created groups of words that seemed to relate to a common topic.

For instance, the category '*anaphora'* contained the words "it", "some", "that", "which", "each", "few" and so on; the category '*conceptual discussion*' contained "think", "cause", "because", "suppose", "impact", and so on. Table 1 shows the final 8 categories constructed from our dataset. We agree that those groups were built in an arbitrary manner, and that some words could belong to several categories. Nonetheless, our approach was data-driven—in the sense that we used the most common words from our dataset—and theory-driven, in that we designed potential indicators for collaborative learning. For instance, the category '*conceptual discussion'* is likely to be associated with higher learning gains, and the category '*anaphoras'* is likely to be associated with a higher quality of collaboration. Why? Because this measure can serve as a proxy for measuring the quality of a common ground between two participants: since anaphoras are ambiguous by nature, they have to be correctly interpreted by the interlocutor and thus indicate a stronger coordination between students. Herbert Clark has developed a considerable body of work investigating this topic [1].

Table 1: Categories built on common unigrams.

Category	**Unigrams**
Jargon	hemi, field, hemifield, brain, eye, lesion, optic, vision, meyers, track, gaze, nerve, hemisphere, loop, information, blind, radiation, meyer, LGN
Diagram	blue, orange, case, circle, box, yellow, line, arrow, white, black, circle, number, half
Location	right, middle, left, top, bottom, diagram, opposite, corner, side, down, underneath, back, inner, outer, between, toward, lower, here, there, first, second, third, fourth, fifth, one, two, three, four, five
Conceptual discussion	think, cause, because, since, change, figure, would, wouldn't, impact, affect, explain, suppose, interpret
Uncertainty	maybe, possible, though, but, know, could, guess
Anaphora (person)	anybody, anyone, both, each, each, other, everybody, everyone, he, her, hers, herself, him, himself, his, I, it, its, itself, me, mine, myself, neither, nobody, others, ours, ourselves, several, she, somebody, someone, their, theirs, them, themselves, they, us, we, who, whoever, whom, whomever, whose, you, your, yours, yourself, yourselves

Anaphora (thing)	all, another, anything, both, each, each, other, everything, few, it, its, itself, most, much, neither, one, none, nothing, one, one, another, other, others, several, some, something, that, these, this, those, what, which

Participants in the experimental group used more anaphoras compared to participants in the control group: F(1,41) = 4.88, p = 0.03. Our results suggest that *real-time mutual gaze perception* may be a way to support dyads in establishing common ground. The findings indicate that participants in the *real-time mutual gaze perception condition* were able to exploit this information to the extent that they could employ ambiguous anaphora, realizing that the pointing manifested by their partner's gaze would disambiguate the referent of their speech act. Additionally, there appears to be a trend showing that more conceptual discussion occurred in the "visible-gaze" group (Fig. 3, right side): F(1,41) = 5.52, p = 0.02. One limitation of this measure is that the number of words representing this construct is relatively small (between 0 and three words used every minute). The other categories did not yield any significant effect.

Even with these limitations, it is interesting to see that categories built on n-grams frequencies can offer a new window into students' collaborative learning processes. In the next section, we employ algorithms from the field of information retrieval to further explore the differences between our experimental groups.

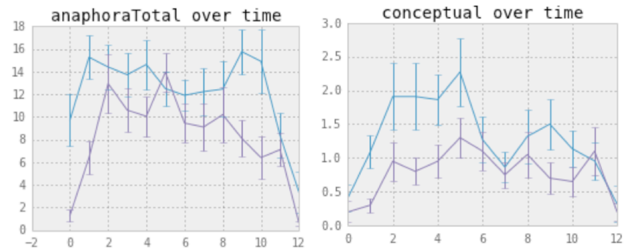


Figure 3: Evolution of words related to conceptual discussion and anaphoras over time. Blue line corresponds to the "visible-gaze" group; purple line to the "no-gaze" group.

3.2 COORDINATION OF LINGUISTIC STYLES (CONVERGENCE)

Computing n-grams counts and probabilities is an interesting way to look at students' discussions. However it doesn't contribute to our understanding of the linguistic patterns used in collaborative learning discussions. To address this issue, we propose studying the ways in which students build a discourse around the instructional material. More specifically, we looked at a specific phenomenon in social interactions called the *chameleon effect*. In a previous study, Danescu [2] showed how in a social setting people tend to mimic their interlocutor's grammatical structure. Here is an example:

Doc: At least you were outside.

Carol: It doesn't make much difference where you are [...]

From Danescu: "Note that "Carol" used a quantifier, one that is different than the one "Doc" employed. Also, notice that "Carol" could just as well have replied in a way that doesn't include a quantifier, for example, "It doesn't really matter where you are...".

In two large datasets (movie dialogues and twitter), Danescu importantly shows that this effect (called *convergence*) is relatively robust and pervasive. That is, people tend to consistently mimic the grammatical structure used by their interlocutor. Previous research suggests that this convergence is associated with enhanced communication in organizational contexts and in psychotherapy (cited in [2]). Our goals are to 1) replicate Danescu's results on our dataset, and 2) test whether *mutual visual gaze perception* supports convergence.

Concretely, Danescu used 9 categories from the LIWC corpus (Linguistic Inquiry and Word Counts [7]) to compute converge measures. Those categories are: articles, auxiliary verbs, conjunctions, high-frequency adverbs, impersonal pronouns, negations, personal pronouns, prepositions, and quantifiers. The way convergence is computed is relatively trivial:

$$P(b^t_{\hookrightarrow a} = 1 | a^t = 1) - P(b^t_{\hookrightarrow a} = 1).$$

The first expression is the conditional probability of seeing word type *t* expressed by person *b* in answer to person *a*, given that *a* used this word type in the previous utterance. The second expression is just the probability of seeing a particular word type in the entire corpus. Subtracting the second expression from the first one gives us a measure of *convergence*.

Figure 4 shows Danescu's results for his dataset. Error bars are flat and barely visible (shown in red) because his dataset is relatively large; dark blue bars show the probability of using a particular word type (e.g., articles, pronouns) and light blue bars show the conditional probability of using a particular word type, given that an interlocutor used the same word type in the previous utterance. Figure 5 shows our replication of Danescu's results. We can see the same pattern emerging: light blue bars (conditional probability that a certain category of words is mirrored by the same word type in the interlocutor's response) are always higher than the probabilities of this type of word in the corpus. Due to our smaller corpus, not all differences are statistically significant, but most of them are (i.e., where the standard errors do not overlap).

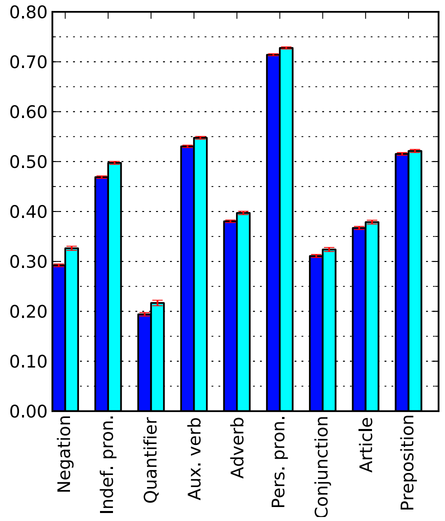


Figure 4: From Danescu [2], this graph shows how people tend to mimic the grammatical structure of their interlocutor. Light blue bars show the conditional probability of using a particular word type, given that an interlocutor used it in the previous utterance. Dark blue bars show the probability of using a particular word type in the entire corpus.

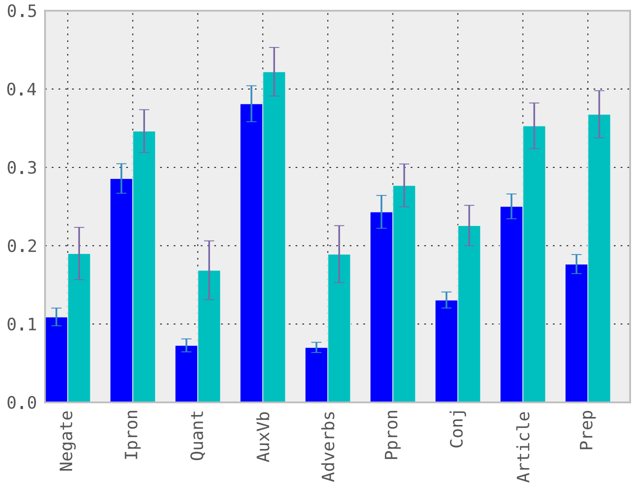


Figure 5: A replication of Danescu's results on the current dataset. Errors bars show standard errors. Non-overlapping error bars show statistically significant differences.

Most importantly, there was special potential in using this measure to discriminate between the two experimental groups (e.g. "visible-gaze" vs "no-gaze"; productive vs poor collaborators; good vs poor learners). Unfortunately, there wasn't any significant difference between those groups on our convergence measure (F < 1). This means that, at least in our corpus, coordination of linguistic styles is not predictive of positive learning gains. It also shows that *mutual gaze perception* doesn't influence this effect: students are not more likely to imitate each others' grammatical patterns if they can see the gaze of their partner in real time.

This convergence measure, however, only looks at superficial features of collaborative dialogues (i.e., word types). It would be much more interesting to look at the words themselves. If one could show that productive students are more likely to mimic the *content* mentioned by their partner, this would be a more interesting result.

3.3 BUILDING ON YOUR PARTNER'S IDEAS (COHERENCE)

In this section, we describe how we summarized our data in a very high dimensional space, separated the transcripts in several consecutive segments, and applied cosine similarity metrics to measure students' coherence. A cosine similarity score indicates how similar two text documents (or subsections of a transcript) are. Our approach was to segment students' transcripts into smaller texts and compute similarity measures between those segments. By iteratively repeating this procedure, we can evaluate the *coherence* of a discussion [6]. The idea behind coherence is that interlocutors tend to adapt to the patterns in each other's utterances; this alignment, in turn, is believed to be indicative of shared understanding (or common ground). Ward and Litman, for instance, showed that coherence was predictive of learning in tutoring dialogues [11]. There has been a significant amount of additional work on this topic, in various domains. We won't summarize the literature on coherence, but the interested reader can look at the work done around Coh-Metrix [3] for more information.

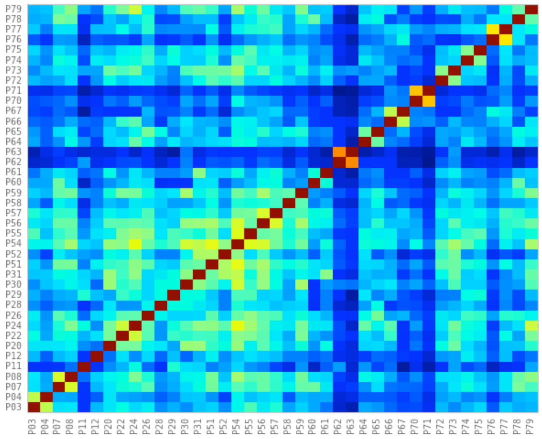


Figure 6: cosine similarity between each participant of the experiment. The diagonal is red because it represents each students' perfect similarity with herself / himself.

The first step of the process was to apply tf-idf transformations (term frequency–inverse document frequency) to our dataset. Tf-idf is commonly used to summarize a text corpus. The value of highly frequent words is decreased, and is offset by their frequency in the corpus; this way, rare words gain a bigger weight and common words (e.g., "the", "it") gain a smaller weight. This technique is used in information retrieval to score documents' relevance to a query. We then compared each student's discourse similarity with other participants by using a cosine similarity measure over the entire transcripts. A cosine similarity measure takes two vectors and computes the magnitude of the angle between them to represent their similarity. We show every pairwise comparison in Figure 6: dark blue lines show students who are very dissimilar to everyone else; hot colors represent similarity. As a sanity check, we can observe that students are identical to themselves (red diagonal); Students in the same group are next to each other on each axis, and we can see that students belonging to the same group tend to resemble each other (2x2 squares along the diagonal). Finally, we can isolate students who are very different from everyone else (e.g. P62 and P63) and try to explain why they are very distinct from other participants: in our case, P63 achieved the lowest learning gain after the activity. P62 was within one standard deviation of the mean.

Additionally, we tried to reorganize students on each axis based on their learning scores (Fig.7, left side) and their quality of collaboration (Fig.7, right side). The first approach did not cluster students in any meaningful way; however, the second one showed that students with a poor quality of collaboration (left and bottom rows) tend to look very dissimilar to everyone else (shown in dark blue). This result suggests that poor collaborative groups can potentially be detected using cosine similarity measures.

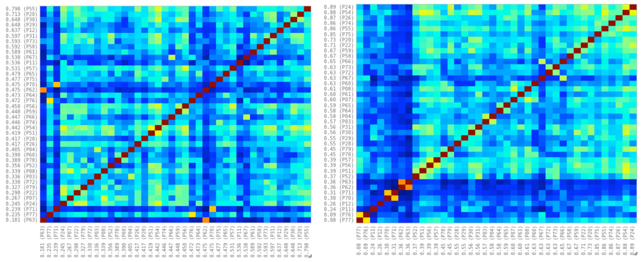


Figure 7: cosine similarity matrix, reorganized with students' learning scores (left) and quality of collaboration (right).

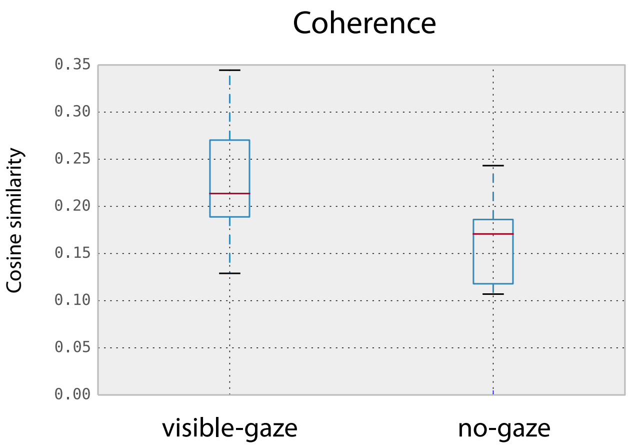


Figure 8: Students' coherence when discussing the task. Students in the "visible-gaze" group were significantly more coherent (p < .05); higher coherence was also significantly correlated with higher learning gains (p < .05).

We then computed a first measure of students' *coherence*: while our approach was simplistic (more complicated measures of coherence do exist [3]), it provided an approach relatively easy to understand and to apply. We built on our previous results using tf-idf and cosine similarity to assess whether students were re-using ideas mentioned earlier in their discussion. More specifically, we considered *n* exchanges and compared them to the *m* previous exchanges. For instance, where n=5 and m=5, we computed the similarity between utterances 15 to 20 (current discussion) with utterances 10 to 15 (ideas exchanged at the beginning of the experiment).

We then iteratively moved this 5-exchanges window through the transcript and averaged the similarity across all exchanges to compute our measure of coherence. Using this measure, we found that students in the "visible-gaze" condition were more coherent than students in the "no-gaze" condition (Fig. 8): F(1,20) = 7.45, p = 0.01, Cohen's d = 0.34 (for the visible-gaze group, mean=0.23, SD=0.07; for the no-gaze group, mean=0.15, SD=0.06). This measure was positively correlated with students' learning gain: r(19) = 0.540, p = 0.011 (Fig. 9). *Those results suggest that students who could see the gaze of their partner in real time on the screen were more likely to have a coherent discourse; additionally, a coherent discourse was more likely to lead to higher learning gains*.

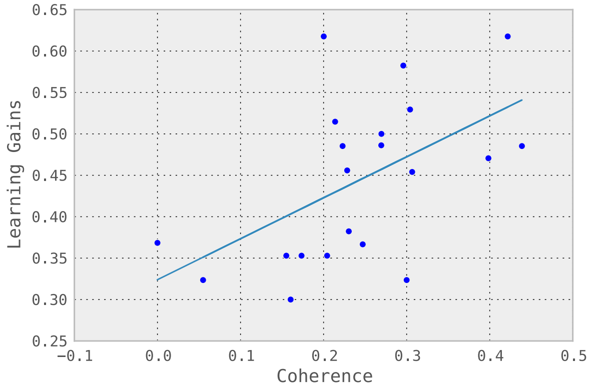


Figure 9: Correlation between dyads' dialogue coherence and learning gain: r(19) = 0.540, p = 0.011.

On a side note, we tried various values for *n* and *m*. Some of those results were not significant, but we always found that students in the "visible-gaze" group were more coherent than students in the "no-gaze" group. At the end, we observed that comparing 5 exchanges with the 5 previous utterances produced the results that were the clearest and easier to interpret.

Here we provide an example of a highly coherent exchange (cosine similarity of 0.5). We highlighted similar words between the two sets of utterances in bold:

--- *Exchange 1* ---

A: **I think** that we did say **the fifth one down**.

B: **OK**. So then it's **lesion five**. **OK**.

A: And you **said** for your answer, you said the third one down whereas I **said** the sixth one down. The rest are **kind of** similar besides for that **kind of** like semi-circle in the middle being **kind of** white.

B: **Right, right**. Hold on. **Number** six, < mumbling to self >, the **number** for that side is gonna be, um, this is tricky business.

A: Yeah it is. < Laughs >.

--- *Exchange 2 (same discussion, continued)* ---

B: **Kind of**? < Laughs >.

A: Yeah. So what do you want to do for **lesion five**?

B: **For lesion five**? Um, **number… the fifth one down**, is that what we **said** originally? **I think** that that's still the correct way to go

A: **OK**.

B: That's what we **said** initially, **right**?

--- *End of Exchange 2* ---

We can observe at least three common repetitions across those two segments. First, the reference to lesion 5 introduced by A in the first exchange and repeated by B in the second exchange. Secondly, both participants express uncertainty by saying "kind of" in the two segments. Finally, there is an abundance of acknowledgement in the form of keywords like "OK" and "right". All those elements point to a relatively solid common ground between the two participants, which is captured by our measure of coherence. Our results, illustrated by the exchange above, is in line with the results of [5], who showed that convergence is not only associated with conceptual understanding but also with affective components such as frustration, engagement and confusion.

3.4 ADDITIONAL RESULTS

In a subsequent step, we sought baselines to use for comparing students' utterance corpora. For instance, we can imagine that comparing the transcripts of students with a baseline of an expert discussion on this topic would be predictive of their learning gains. To this end, we used two corpora as references: first, we used the best student (in terms of her learning score) of our dataset (P55). She was in the visible-gaze condition and got an impressive 80% gain on the post-test, where the average was around 50%. Second, we inserted the text that students had to read in the 2nd step of the experiment into our dataset. This text is highly technical and is likely to pick up students' use of the particular terminology associated with this domain.

We found that students in the "visible-gaze" group looked more like P55: F(1,39), p = 0.04, Cohen's d = 0.35 (visible-gaze mean=0.97, SD=0.27; no-gaze mean=0.80, SD=0.20).

Interestingly, this measure was positively correlated with students' quality of collaboration: r(38) = 0.545, p < 0.001. There wasn't any difference between the two groups when looking at their similarity with the textbook chapter: F(1,39), p = 0.17, Cohen's d = 0.10 (visible-gaze mean=0.11, SD=0.04; no-gaze mean=0.09, SD=0.04). However, this measure was significantly correlated with students' conceptual understanding of the topic taught: r(38) = 0.335, p = 0.035.

In summary, it appears that taking different baselines is helpful for finding relevant predictors of good learning groups. Taking a student's cosine similarity with a standard reference of domain knowledge (i.e., a textbook chapter) seems to be associated with higher learning on a test. Taking a student's cosine similarity with the "best" student of the dataset seems to be associated with productive patterns of collaboration. This makes sense, since students' utterances reflect the way novices discuss and learn about a new topic; a scientific text, on the other hand, is produced by experts who have mastered the concepts and terminology of a domain. In sum, those two features could be advantageously used to further explore students' discussion, as well as to feed machine learning algorithms trying to predict students' learning.

3.5 PUTTING OUR MEASURES TOGETHER: PREDICTING STUDENTS' QUALITY OF COLLABORATION AND LEARNING GAINS USING LINGUISTIC FEATURES

Our final contribution is to test whether the measures described above have any predictive value. More specifically, can we roughly classify students in terms of their learning gains using machine learning algorithms? To answer this question, we separated our participants into two groups based on the median value of students' learning gains. We then tried to predict in which group each student belonged, i.e., below or above the median split.

We then used our hand-labeled categories from section one (n-grams), the cosine similarity scores, the convergence measures and the coherence metrics as features. The complete dataframe contained 60 features and 40 rows. We used the built-in version of Support Vector Machine (SVM) provided by Matlab with a forward search feature selection and tried various kernels (linear, quadratic, polynomial, Gaussian, multilayer perceptron). For the learning scores, we found that SVM with a multilayer perceptron kernel and 8 features could correctly classify 94.44% of our participants. We also used a *validation set* (4 participants, which constitutes 10% of our sample). Those 4 participants were randomly selected from our dataset and we predicted whether they were above or below the median split on the learning gains after we found our best model. On the validation set, our model correctly classified 75% of the participants (3/4).

Those results are impressive, but they need to be hedged with healthy skepticism. First, many features were used to make this prediction. It is probable that the algorithm is cherry-picking the relevant features to improve its accuracy (which is also over-fitting the data). Secondly, the training set is rather small. There are only ~40 students to classify, which is another serious limitation. Finally, even though we are using a validation set, it should be kept in mind that this set is small (only four datapoints). Finally, those results should be contrasted with other baselines, such as decision trees or naïve bayes.

Table 2: Rough classification of students (using a median-split) in terms of their learning gains.

	Accuracy on the test set	Accuracy on the validation set	Features
SVM	94.44% (34/36)	75% (3/4)	Uncertainty, Negations, Aux. Verbs, Length Sentence, Prepositions, Number of words used, Number of Anaphoras, Impersonal Pronouns

In sum, these analyses indicate noteworthy promise in using linguistic features to predict students' learning and ability to collaborate with their peers, but those results need to be replicated on larger datasets to be truly convincing.

Interestingly, SVM selected some of the correlations we found above between students' learning gains and particular features of our transcripts: number of anaphoras used and keyword showing students' uncertainty. However, other measures such as coherence, cosine similarity with a textbook chapter were not included in our final model. Instead, it favored low-level measures, such as the number of words used by students, the length of their sentence and particular grammatical forms (negations, auxiliary verbs, prepositions). This shows that some variables may be good predictors in isolation, but lose their predictive power when associated with other measures.

4. DISCUSSION

The goal of this project was to explore various NLP techniques to make sense of educational datasets; we favored a "breadth" approach where we tried promising techniques rather than exploring one specific measure in depth. In future work, we will go back to our most promising results (e.g., coherence and cosine similarity) and explore them in more detail, as well as to examine not only the cosine similarity to the best student of the other students' transcripts but to more aggregate exemplars of 'better or worse students', such as the upper and lower quartile of the students in terms of learning score.

To recap our results, we have found that: 1) n-grams probabilities can help characterize groups of students in terms of building a common ground with their partners (anaphoras); 2) cosine similarity measures are most useful when used with a "reference" corpus (e.g., textbook chapter; transcript of a very good student as measured by learning gains); 3) coordination of linguistic style has little predictive power in terms of explaining dyads' collaborative learning processes; 4) coherence measures, on the other hand, are positively associated with students' learning 5) using SVM and the features mentioned above, we can roughly predict students' learning outcomes with an accuracy higher than 90% (which dropped to 75% for our validation set).

We argue that our approach is especially useful when analyzing the results of a controlled experiment. We were able to characterize the effects of *mutual gaze perception* on students' discourse, and we found interesting predictors for learning gains and students' collaboration quality. However, we also argue that those techniques could be used in other domains. For instance, comparing the similarity between a reference text and students' utterances has already been used for assessing essays. Coherence can be used in similar contexts. More interestingly, those metrics could be advantageously used on multi-modal datasets. Eye-tracking data, for instance, could be converted in a series of word tokens representing the location of students' gaze over time. Similarity measures could then be used as described above to characterize visual exploration of a problem space. We believe that NLP measures have been too rarely used on non-linguistic datasets (e.g., gestures, as measured by a kinect sensor; gaze, as measured by eye-trackers; arousal, as measured by galvanic skin response devices) and could provide new insights into the ways that students construct their understanding of a particular concept, and to establish a productive collaboration with one another.

Limitations of this work have been mentioned in previous sections (e.g., small dataset, limited amount of error analysis). Replicating those results on larger datasets would make a more convincing argument for using NLP measures in education.

5. CONCLUSION

This paper showed NLP approaches offer substantial promise for understanding educational datasets and automating currently unwieldy and time-consuming hand analyses. The measures described above could easily be applied to other settings, such as forums or online discussions. Future work includes refining those measures and deepening our sense of their predictive value; replicating those results on other datasets; and exploring additional topics in NLP (e.g., topic modeling with Latent Semantic Analsyis or Latent Dirichlet Allocation).

6. ACKNOWLEDGMENTS

We gratefully acknowledge grant support from the National Science Foundation (NSF) for this work from the LIFE Center (NSF #0835854).

7. REFERENCES

1. Clark, H.H. and Brennan, S.E. “Grounding in communication.” In L.B. Resnick, J.M. Levine and S.D. Teasley, eds., *Perspectives on socially shared cognition*. American Psychological Association, Washington, DC. 1991, 127–149.
2. Danescu-Niculescu-Mizil, C. and Lee, L. Chameleons in Imagined Conversations: A New Approach to Understanding Coordination of Linguistic Style in Dialogs. *Proceedings of the 2Nd Workshop on Cognitive Modeling and Computational Linguistics*, Association for Computational Linguistics (2011), 76–87.
3. Graesser, A.C., McNamara, D.S., Louwerse, M.M., and Cai, Z. “Coh-Metrix: Analysis of text on cohesion and language.” *Behavior Research Methods, Instruments, & Computers 36*, 2 (2004), 193–202.
4. Meier, Anne, Hans Spada, and Nikol Rummel. "A rating scheme for assessing the quality of computer-supported collaboration processes." *International Journal of Computer-Supported Collaborative Learning, 2*, 1 (2007), 63-86.
5. Mitchell, C.M., Boyer, K.E., and Lester, J.C. From Strangers to Partners: Examining Convergence Within a Longitudinal Study of Task-oriented Dialogue. *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Association for Computational Linguistics (2012), 94–98.
6. Pickering, M.J. and Garrod, S. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences 27*, 02 (2004), 169–190.
7. Pennebaker, J W., M. E. Francis, and R. J. Booth. "Linguistic inquiry and word count: LIWC 2001." *Mahway, NJ: Lawrence Erlbaum Associates,* 2001, 71.
8. Schneider, B. and Pea, R. Real-time mutual gaze perception enhances collaborative learning and collaboration quality. *International Journal of Computer-Supported Collaborative Learning 8*, 4 (2013), 375–397.
9. Sherin, B. "Using computational methods to discover student science conceptions in interview data." *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. New York: ACM, (2012).
10. Tomasello, M. “Joint attention as social cognition.” In C. Moore and P.J. Dunham, eds., *Joint attention: Its origins and role in development*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1995, 103–130.
11. Ward, A. and D. Litman. Dialog convergence and learning. In *International Conference on Artificial Intelligence in Education* (AIED). 2007. Los Angeles, CA.

The Opportunities and Limitations of Scaling Up Sensor-Free Affect Detection

Michael Wixon
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
mwixon@wpi.edu

Ivon Arroyo
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
iarroyo@wpi.edu

Kasia Muldner
Arizona State University
699 Mill Street
Tempe, Arizona
Katarzyna.Muldner@asu.edu

Winslow Burleson
Arizona State University
699 Mill Street
Tempe, Arizona
winslow.burleson@asu.edu

Cecil Lozano
Arizona State University
699 Mill Street
Tempe, Arizona
calozano@asu.edu

Beverly Woolf
University of Massachusetts-Amherst
140 Governors' Drive
Amherst, Massachusetts
bev@cs.umass.edu

ABSTRACT

We develop and analyze affect detectors for four affective states: confidence, excitement, frustration and interest. We utilize easy to implement self-report based "ground truth" measurements of affect within a tutor, and model them as continuous variables that are later discretized into positive, neutral, and negative valence classifications; this distinguishes our work from detectors which model affective states as binary. We explore the opportunities and limitations of cross validation with regard to potentially distinct sample groups.

Keywords

Affective computing, human factors, intelligent tutors, prediction, models, feature engineering, sensor-free affect detection

1. INTRODUCTION

One key factor that influences students' academic success is their emotions and general affective experience while learning. For instance, positive affect has a facilitative effect on cognitive functioning in general [1], and improved performance on creative problem solving in particular [2, 3]. Moreover, students who are interested in an activity persevere in the face of failure, invest time when needed, and engage in mindful processing [4]. Even some emotions traditionally viewed as negative can be beneficial – for example, confusion is associated with learning under certain conditions [5]. In contrast, the affective state of boredom reduces task performance [6], increases ineffective behaviors such as gaming the system [7], and tends to be persistent once experienced [7].

Given the pivotal role that affect plays in education, both in short-term performance outcomes and in long-term career choices, there is growing interest in developing educational technologies that can recognize and respond to student affect. Here, we focus on the first thrust, namely affect recognition.

The process of modeling motivation and emotion is summarized in Figure 1, which shows how emotions are highly dependent on context, and are expressed in behaviors. Thus, when designing models to assess student emotion, it is essential to empirically understand which factors impact a student's emotional state, and how the affective state is revealed by the student in terms of subsequent actions and behaviors.

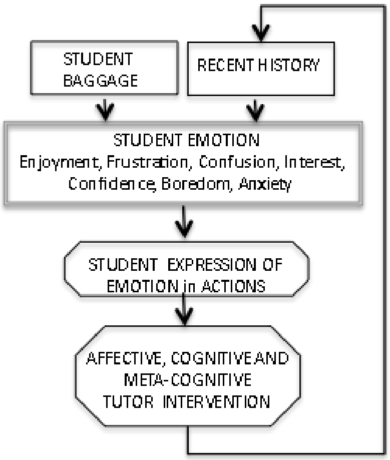

STUDENT BAGGAGE
RECENT HISTORY
STUDENT EMOTION
Enjoyment, Frustration, Confusion, Interest, Confidence, Boredom, Anxiety
STUDENT EXPRESSION OF EMOTION in ACTIONS
AFFECTIVE, COGNITIVE AND META-COGNITIVE TUTOR INTERVENTION

Figure 1: Model of Student Emotion while Learning (arrows indicate dependence, causality, precedence). A student's emotion while learning (grey frame) is originally unknown and hidden. It is influenced by the "student's baggage" (initial achievement, affective predisposition) and recent history of the student in the software (tutor moves or student actions). This article focuses on the top boxes: how student baggage and recent history help to predict current emotional states.

One approach to modeling affect, summarized in a recent review [8], pertains to using sensing devices. For instance, in our past work, we have created models of affect using data from a camera, pressure mouse, skin conductance bracelet, and pressure chair cushions, in conjunction with data coming from a student's interaction with an intelligent tutoring system [9-11]. The subsequent models achieved 85% accuracy when compared to the students' self-reported emotion. Muldner et al. [12] used data from a subset of these sensing devices plus an eye tracker to detect moments of delight during instructional activities. D'Mello et al. [13] used dialog and posture features to model affective states. In Conati's model [14], affect is modeled using one sensor modality, namely an EEG, in addition to interaction features [15].

While this research highlights the utility of sensors for affect recognition, they can not be widely disseminated in schools where the tutoring systems are used, though this may not be true in the future. Data collection is thus more challenging beyond lab studies. Thus, researchers have begun exploring sensor free affective detection. For instance, Baker et al. [16] used only data from students' interaction with a tutor to model affective states such as frustration.

The work reported in this paper adds to research on sensor-free affective models. Specifically, our goal is to better understand contextual predictors of student emotion, and to generate models that use the context in which student emotion occurs to predict this emotion, based on student behaviors within the software. To replace the rich physiological information that sensors provided, we focus on feature engineering, such as summaries of "recent history" of student actions. Additionally, our second goal was understand the utility of students' affective predispositions – attitudes, general values, preferences, and self-efficacy for the domain – for affect detection (see Figure 1). Last but not least, we analyze the generalizability of our affect detectors to different populations of students to other students in new schools.

2. METHODS

2.1 Participants

We used three data sets to train and test ten separate models.

2009 Data Set. An affect detector was built and tested using 295 students, 7th, 8th, 9th and 10th graders from two rural area schools in Massachusetts in the Spring of 2009, using six fold student level batch cross validation [17]. On average, 1138 instances (problem-student interactions) were split across six batches used to train and test each affect model.

2011 Data Set. An affect detector was built and tested using 123 students, 7th and 8th graders from a third rural area school in Massachusetts in 2011, using three fold student level batch cross validation [17]. On average, 120 instances (problem-student interactions) were split across three batches and used to train and test each affect model.

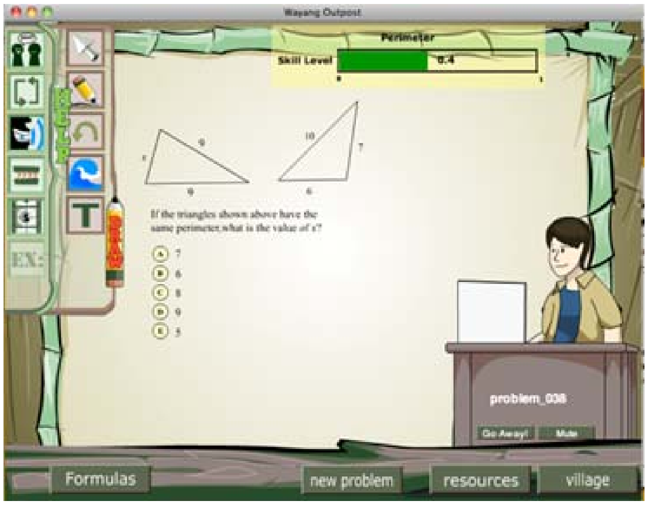
2013 Data Set. An affect detector was built and tested using 43 students, 7th, and 8th graders from two schools in California and Arizona in the Summer of 2013, using two fold student level batch cross validation [17]. On average, 76 instances (problem-student interactions) were split across two batches and used to train and test each affect model.

2.2 Wayang Outpost

The test-bed for this research was Wayang Outpost (see Figure 2). Developed at UMass-Amherst, this tutor shows evidence of promoting effective math learning, has been used by tens of thousands of students in the United States and has consistently shown significant learning gains, e.g., on mathematics tests (an increase of 12% from pre- to post-test after only 4 class periods), and on state standard exams (92%) as compared to students not using Wayang (76%) [11, 18, 19]. Students using Wayang have also improved more on MAP scores compared to control groups (MAP is a national test of Northwest Evaluation Association on specific topics).

2.2.1 *Pedagogical Approach*

The pedagogical approach of the Wayang Tutor is based on cognitive apprenticeship [20] and mastery learning. Cognitive apprenticeships are designed to bring tacit processes into the open, so that students can observe, enact, and practice them with help from the teacher. This process involves several phases: modeling (introduction to the topic via worked-out examples, making steps explicit, and working through a problem aloud); practice with coaching (offering feedback and hints to sculpt performance to that of an expert's); scaffolding (putting into place strategies and methods to support student learning, offering hints as well as worked-out examples and tutorial videos); and reflection (self-referenced progress charts that allow students to look back and analyze their performance).



If the triangles shown above have the same perimeter, what is the value of x?

Ⓐ 7
Ⓑ 6
Ⓒ 8
Ⓓ 9
Ⓔ 5

Figure 2: Learning companions use gestures to offer advice and encouragement. Students can ask for hints or click the "solve it" button. Animations, videos and worked-out examples add to the spoken hints about the steps in a problem.

An important part of cognitive apprenticeship is the provision of materials just beyond what the learners can accomplish by themselves. Vygotsky referred to this as the Zone of Proximal Development (ZPD) and believed that fostering development within this zone leads to the most rapid learning [21]. We have operationalized and parameterized ZPD within the context of intelligent tutoring systems [19] and formalized a mechanism for adaptive problem selection that tailors the difficulty of subsequent math problems to past student performance and effort [19]. Wayang also identifies the most critical cognitive skills and predicts the likelihood of success on future problems related to these skills [9]. Wayang supports students by offering hints, examples, short video tutorials, and animations [22-24]. Rich multimedia help is provided when students make mistakes or ask for help, following principles of multimedia learning theory [25].

Teachers can access real-time assessments about individual student progress via the "Teacher Tools", which allow them to spot and focus on students who need help, problems that are hard for everybody, and math skills with which the class as a whole is struggling.

2.2.2 *Affective Learning Companions.*

In our past work, we integrated into Wayang gendered and ethnic learning companions (male and female, White, Hispanic and African American), whom offered advice and encouragement by talking to students (see Figure 2 for a sample character). These companions can gesture and train attributions for "success/failure", e.g., that intelligence is malleable, perseverance and practice are needed to learn, making mistakes is an essential part of learning, and failure is not due to a lack of innate ability. In controlled randomized studies with hundreds of students, certain groups of students (females and students with disabilities) reported decreased frustration and increased confidence levels when working with learning companions and increased frustration when companions were not present [26]. In addition, student enjoyment and interest were higher compared to students not given learning companions, suggesting that such affective pedagogical agents can impact students' emotions [27, 28]. Moreover, students receiving companions described higher self-efficacy in mathematics, and exhibited more productive behaviors within the tutor.

3. PROCEDURE

In the present study, while working within Wayang Outpost, students were periodically prompted to report their current affective state, using a simple dialogue box. The design of these prompts was based on prior work used to gather information on "the range of various emotional states during learning" [29], where affective states are placed on spectra ranging in valence from negative to positive. The following affective states were measured with a Likert scale (1-5): confidence, excitement, frustration and interest. Each of these scales is bipolar (e.g. confidence/anxiety). For simplicity we will refer to each of these bipolar scales as confidence, excitement, frustration and interest. In this article, a higher Likert score indicates a positive level of the affect in question (i.e., for confidence, 5 is Highly Confident, while 1 is Anxious). In the 2009 and 2011 data sets all four affects were examined, however for the 2013 data set only excitement and interest were measured via self-report.

Recognizing emotion from log data involved a seven step process. First, mathematics problems that students were not expected to solve were removed (e.g., topic introductions and example problems). Second, the student data was batched to ensure each batch had a representative sample of all "ground truth" Likert scale self-reports for all four emotions. Third, missing values were imputed at the batch level using a multiple regression algorithm in SPSS [30], thus filling all cells of missing data with estimate values. Fourth, outliers were identified at the full data set level also using SPSS. Fifth, engineered features were computed from the initial raw log data; some rows of data (e.g. topic introduction problems & example problems where students were meant to observe rather than interact with the system) were removed at this level as well. Sixth, the data was split into ten data sets: one for each combination of year and the four affects to be detected (e.g. confidence 2009, excitement 2013, etc). Seventh, forward feature selection and a linear regression algorithm was run in Rapidminer [31] under batch cross validation [17] in order to build the ten regression models, one for detecting each of the four affects in each of the two sample groups, 2009 and 2011, and two for detecting excitement and interest in the 2013 data set (as only two emotions were self-reported in 2013). Step two (Data Cleaning & Batching), step five (Feature Engineering), and step seven (Model Creation--running the linear regression algorithm) will be addressed in greater detail.

3.1 Data Cleaning & Batching

Data was batched at the student level, meaning that the data from one student could span across more than one batch. The process of batching was not completely random as consideration was given to preserving roughly equal representations of the target self-reported affect in each batch. Thus, students were assigned to batches randomly several times, and each batch was examined to show how many times students had responded with each value of the Likert scale for a given affect. For example, if one batch included 80 instances of students responding with 1 (one) in terms of frustration (low frustration) and another batch included only 10 instances of students with responses of 1 for frustration then that set of batches was rejected and batching was performed again. In some cases it was necessary to manually swap individual students between batches in order to maintain a balanced ratio of responses. The size and quantity of the batches were also limited by concerns of over representation. For example, in the 2011 data set there were only 10 reported cases of interest > 3 out of a total of 105 cases. The fact that less than 10% of our data reported a positive valence in interest for this data set partially explains the relatively poor results of the detector trained on 2011 data, and attempts to "balance" batches by making proportions of each Likert response across batches as equal as possible. It also addresses why the large 2009 data set is split into six batches while the much smaller 2011 data set could only be split into three batches.

3.2 Feature Engineering

The majority of the features were derived from eight low level descriptions of students' behavior with each problem a student saw (Table 1). Each state first acts as an if-statement predicated that the statement preceding it is not true (e.g. if a student did not SKIP, then the problem is evaluated to see if they met the criteria of NOTR and so on).

These eight simple student states were mutually exclusive and assigned per problem, i.e., for a given problem a student's actions might be classified as ATT vs. SOF. From these seven features, 21 new features were generated by looking at the prior 3 actions (i.e., NOTRLast3), each of which weighs a more recent instance more heavily than the one that preceded it; for instance, in NOTRLast3 the immediate preceding action is worth 3, the action before that is worth 2, and so on. The remaining features were patters of behaviors, derived from transitions from one student-problem interaction state to another (e.g. NOTR→ATT means that the current student-problem interaction has a state of ATT, and the previous one had a state of NOTR). Due to the fact that several features were based on the prior three actions or prior three transitions between problems, the first four problems of any student's work within Wayang were excluded from our analyses. This also means that, going forward, these detectors will only be usable after the student has already completed four problems.

Features also included running tallies of incorrect attempts, hints seen, problems solved on first attempt, and other assorted student actions aggregated over the current problem and prior three problems. Several hundred features were generated and only a small number were selected for use in models in this work; we limit our discussion to the features that were selected.

Table 1. Eight Low Level Student States

Student State	Description of student Behavior
SKIP	The student did nothing and skipped the problem.
NOTR (Not Reading)	The student made a first attempt to solve a problem in a time under 4 seconds –not enough time to even read the problem.
GIVEUP	The student took some action, but then skipped the problem without solving it.
SOF (Solved on First Attempt)	The student solved the problem on their first attempt, without seeing any help.
BOTT (Bottom Out Hint)	The student saw all hints available, including the last available hint that gave the answer.
SHINT (Student Hint Request)	Student answered the math problem eventually right, with at least 1 hint.
ATT (Attempt)	The student didn't see any hints and solved it correctly after 1 wrong attempt.
GUESS	The student solved it correctly with no hints and more than 1 incorrect attempt.

For the features shown in Table 2, "Avg" denotes an average taken across the prior four problems, "Last4" denotes the sum of the prior four problems, "Max" denotes the maximum number of actions in a given problem over the prior four problems, "Min" denotes the minimum number of actions in a given problem over the prior four problems, and % denotes the ratio of a particular action in the past four problems over the total actions in the past four problems.

3.3 Model Creation

Once the batching of the data was finalized, each data set was split into the four subsets, each addressing the emotion in question: confidence, excitement, frustration, and interest. Initially, forward feature selection (with a limit of ten features) was carried out for each of the four types of affect for each data set, with student-level batch cross validation [17].

Linear regression was performed in Rapidminer [31] on each of these new subsets under batch cross validation [17]. The models were assessed by Pearson's R to determine their correlation with the target affect. Further, in order to create a discrete classification measure of affect, the Likert scale responses and linear regression model output were rounded to the nearest integer and then discretized as follows: All responses below 3 on the Likert scale were labeled as "negative", all responses equal to 3 were labelled "neutral", and all responses above 3 were labeled as positive. These classification results were assessed using weighted kappa [32], which is a measure of agreement for polynomial classified targets. Similarly to typical Cohen's kappa [33], a zero denotes agreement due to random chance, while a one denotes perfect agreement between the model and student self-reports of affect.

While detector results obtained under batch cross validation should guard against overfitting, there is still the potential risk that the results may be overfit to the sample group used in the study. In particular, even with batch cross validation, all the batches are drawn from the same sample group, who may share various specific traits. Therefore, the batch cross validated models trained using the 2009 data set was applied to the 2011 & 2013 data sets and vice-versa. This was done to provide a more conservative estimate of the models' generalizablity to new data sets, given that the samples were collected from distinct groups of students at distinct points in time.

Table 2. Features from Students' Interaction in Wayang

AvgTimeToSolve – The average of time to solve a problem.
LogTimePerAction – The logarithm $\log_{10}$ of the time per action
AvgTimePerAction – The average time per action
Hints – Total hints given on current problem
Wrong – Total wrong attempts on the prior problem
WrongLast4 – Total wrong attempts aggregated over the current and last 3 problems.
MaxWrong – The maximum number of incorrect attempts
MaxActions – The maximum number of actions
MinWrong – The minimum number of incorrect attempts
TimetoSolve – Time to solve a problem
LogTimeInTutor – Logarithm $\log_{10}$ of student's time in tutor.
TimeInTutor – Total student's time in tutor.
MinTimePerAction – The minimum time per action of the past 4 problems.
MinLogTimePerAction – The minimum of the logarithm $\log_{10}$ of seconds per action.
TotalActions – The total actions of the prior problem.
%Wrong – The percent of incorrect attempts.

4. RESULTS

4.1 Feature Selection

Forward feature selection yielded a total of forty eight features. These features were split across ten different detectors/models, four for the 2009 data set, four for the 2011 data set, and two for the 2013 data set where only self-reports on excitement and interest were collected. While there were ten models and ten features used per model, only 48 features were required rather than 100, because some features were used in more than one model. Twenty seven of these features were engineered from the states described in Table 1. Of the remaining twenty one features, sixteen were based on other student interactions within the system (see Table 2). Many of these features were based on student actions on an immediate given problem, but some denoted with "Avg", "Max", "Min" or "%" are based upon the current problem and three preceding problems: "Avg" denoting Average, "Max" denoting maximum, "Min" denoting minimum, and "%" denoting the percentage of a particular action out of the total actions taken over the current and prior three problems.

The remaining five features (see Table 3) were based on students' responses on the pretest, surveys, and the experimental conditions. These features remain constant from problem to problem.

Table 3. Pretest and Agent Based Features

Features Based on Survey Responses and Agent's Behavior
CON – Baseline measure of confidence when problem solving.
FRUS – Baseline measure of frustration when problem solving.
INT – Baseline measure of interest towards problem solving.
MathValuing – Baseline measure of the degree to which the student values mathematics.
pre_lor –Student’s mastery orientation (willingness to learn new and interesting things in spite of challenge) based on a survey.

4.2 Model Performance

The R values of the linear regression models derived from the selected features achieved a fit comparable with prior work detecting frustration [34, 35], as well as boredom, confusion, and flow [35]. Specifically, prior work has achieved detectors of frustration with kappa values ranging from 0.16 to 0.32 [16], and boredom at kappa = 0.28 [16]. While the detectors presented in this paper may achieve slightly lower kappas than detectors presented in the above cited work, it’s important to note that our kappas are weighted [32], which suffer a penalty as compared to the typical Cohen’s kappa [33] that is meant for bivariate classification. Consequently our model distinguishes between three possible classifications rather than two. This increased the likelihood of accidental misclassification, but with the benefit of more sensitive measurement. One cost of modeling affect as polynomial rather than binary is that binary classification has metrics for false and true positive and negative rates such as sensitivity and specificity [36] or A’ [37], which we cannot utilize in this work.

It is important to note the sample size when considering the relative strength of each model. As previously mentioned the largest sample was found in the 2009 data set, where each model was built on an average of 1138 instances split across six batches. The 2011 data set contains 120 instances split across three batches. However, for the 2011 data set there were only ten instances of positively valenced interest. The particularly low values of interest in 2011 may explain why the 2009 derived model better predicts interest in that sample than the 2011 derived model.

Tables 4 through 7 show performance indicators of each model, which consist of R values (indicating model fit) and weighted kappas [32] (denoted by “K”, indicating classification power into low/neutral/high levels). Each cell contains performance results for a model created from a dataset indicated by the column, and evaluated over a dataset indicated by the row. Note that values along the diagonal (in bold) correspond to testing and training over the same data set. In such cases, student level batch cross validation was used to prevent overfitting. The process of applying the model to the same data set (to generate estimates of the emotion) is thus slightly different than for other cells. Under batch cross validation, a separate model is generated (i.e. trained) for each batch, and estimations/classifications are made for the testing batch. The performance of six distinct models is thus aggregated in the end for the 2009 data set; the performance of three distinct models is aggregated in the 2011 data set); and the performance of two distinct models is aggregated in the case of the 2013 data set.

Table 4. Confidence Detector Performance (Pearson’s R & Cohen’s Kappa)

	2009 Model	2011 Model

2009 Data Set
N = 1102
	R = 0.404
	K = 0.200
		R = 0.306
		K = 0.163

2011 Data Set
N = 127
	R = 0.515
	K = 0.249
		R = 0.238
		K = 0.147

Table 5. Frustration Detector Performance (Pearson’s R & Cohen’s Kappa)

	2009 Model	2011 Model

2009 Data Set
N = 1159
	R = 0.372
	K = 0.173
		R = 0.307
		K = 0.146

2011 Data Set
N = 125
	R = 0.374
	K = 0.139
		R = 0.341
		K = 0.281

Table 6. Excitement Detector Performance (Pearson’s R & Weighted Kappa)

	2009 Model	2011 Model	2013 Model

2009 Data
N = 1145
	R = 0.224
	K = 0.151
		R = 0.211
		K = 0.083
			R = -0.089
			K = -0.022

2011 Data
N = 122
	R = 0.454
	K = 0.278
		R = 0.316
		K = 0.131
			R = -0.142
			K = -0.050

2013 Data
N = 66
	R = 0.004
	K = 0.102
		R = 0.201
		K = -0.024
			R = 0.137
			K = 0.192

Table 7. Interest Detector Performance (Pearson’s R & Weighted Kappa)

	2009 Model	2011 Model	2013 Model

2009 Data
N = 1145
	R = 0.240
	K = 0.090
		R = 0.058
		K = 0.026
			R = 0.071
			K = -0.024

2011 Data
N = 105
	R = 0.300
	K = 0.140
		R = 0.174
		K = 0.005
			R = -0.001
			K = -0.036

2013 Data
N = 86
	R = 0.006
	K = 0.055
		R = 0.153
		K = -0.023
			R = 0.020
			K = -0.144

In general, the results in Tables 4-7 show that: a) affect detectors for confidence/anxiety, excitement and frustration achieve reasonable levels of performance, while for interest/boredom, the R and Kappa values are much lower; b) models generated over larger datasets transfer better to smaller datasets, compared to the other way round; c) models perform similarly well across 2009 and 2011 but not as well over the 2013 dataset, which corresponded to a summer camp in a different part of the country; d) models created over the 2013 dataset don’t transfer well to the 2009-2011 datasets either. These points will be explored in the discussion section.

4.3 Linear Regression Models

The linear regression models for the four affect states are displayed in Tables 8 through 11.

Table 8. Models of Confidence

2009 Features	Weight	2011 Features	Weight
NOTR→BOTT	-53.00	GIVEUPLast3	75.77
BOTT→GUESS	-21.64	NOTR→BOTT	-40.42
GIVEUPLast3	-10.74	BOTT→BOTT	5.14
SOFLast3	0.34	SOF→BOTT	-4.96
Pre_LOR	0.34	SOFLast3	1.06
MinLogTimePerAction	0.31	Pre_LOR	0.87
Wrong	-0.20	MaxWrong	0.28
WrongLast4	-0.07	WrongLast4	-0.27
FRUS	-0.04	CON	0.10
CON	0.04	TimetoSolve	0.01

Table 9. Models of Frustration

2009 Features	Weight	2011 Features	Weight
NOTR→NOTR	-99.37	GUESS→NOTR	-79.74
GIVEUP	11.56	SHINT→NOTR	-36.07
GUESS→SOF	-2.47	GIVEUP	-22.85
SHINT→SOF	-1.58	SHINT	-3.32
%Wrong	0.66	SOF	-1.77
AvgTimePerAction	-0.24	%Wrong	0.98
WrongLast4	0.09	Pre_LOR	-0.53
TotalActions	0.05	INT	-0.12
FRUS	0.04	CON	-0.09
INT	-0.04	MaxActions	0.08

Table 10. Models of Excitement

2009 Features	Weight	2011 Features	Weight	2013 Features	Weight
BOTT→ NOTR	-74.00	GIVEUP	35.24	SHINT→ BOTT	66.89
SOF→ NOTR	-22.52	BOTT→ SHINT	25.32	SHINT→ SKIP	-4.31
Min Wrong	-2.57	Pre_LOR	-0.84	SKIP	2.80
SOF→ BOTT	2.55	Hints Seen	-0.49	SHINT→ SHINT	2.09
Incorrect Attempts	0.14	INT	-0.14	Pre_LOR	-0.76
INT	-0.14	Wrong Last4	0.05	Hints Seen	-0.36
Wrong Last4	0.12	CON	0.05	CON	0.08
Max Wrong	-0.07	LogTime InTutor	-0.04	AvgTime ToSolve	0.01
MinTime PerActio	-0.01	AvgTime PerAction	-0.01	TimeIn Tutor	< 0.01
TimeIn Tutor	< 0.01	AvgTime ToSolve	< 0.01		

Table 11. Models of Interest

2009 Features	Weight	2011 Features	Weight	2013 Features	Weight
SOF→ SHINT	1.16	GIVEUP	349.31	NOTR→ SOF	30.20
SHINT	1.06	GIVEUP →SOF	-180.20	BOTT	19.68
%Wrong	-0.56	SHINT→ NOTR	52.42	SOF→ GUESS	-8.15
SOF	0.41	SHINT→ SHINT	26.43	SKIP→ SOF	-7.33
Pre_LOR	0.37	NOTR→ SOF	-17.61	SHINT →GUES	6.43
INT	0.08	SOF→ NOTR	17.00	LogTime InTutor	-0.09
Total Actions	-0.05	BOTT→ BOTT	7.14	Max Wrong	-0.07
MinTime PerActio	-0.02	Math Valuing	0.09	INT	0.05
TimeIn Tutor	< 0.01	MinTime PerAction	0.03	TimeIn Tutor	< 0.01
		LogTime InTutor	0.02		

5. DISCUSSION

In this paper, we have proposed several models of affect based on students' interaction with a tutoring system. In so doing, we have independently replicated prior work on sensor-free affect detection and contributed to existing work on predictive features of student affect and methods for building models of affect. In the following section we address opportunities and challenges regarding generalizability of the models to new populations.

A major opportunity is to develop detectors which respond to differences between classrooms, schools, and even different regions of the country. We generated a rich set of features which combined student behaviors in the last problem seen, recent history, patterns of student behaviors, and even students' affective background before starting the tutoring session. A combination of features from all these categories were best predictors for each affective state, showing that a variety of student descriptors as well as their behaviors can help to predict emotional states while learning.

It is important to note that while some of the features we used bear a similarity to those in other research, the features are dependent on the environment from which they are inferred. Thus, validation is needed to ensure that these features transfer and apply to other tutoring systems, such as Wayang Outpost.

In designing the features used, consideration was given to other detectors of affect [16, 38]. There is a tension between trying to use similar features from other systems, and recognizing features as being contextually distinct; this makes detector construction a custom work on each system. In the future, it is our hope to design even more informative features. This could be done by examining the data to look for patterns of behavior that align to affective states, and to observe students using the software for behaviors that might have been overlooked and could be indicators of affect. While examining the data in such a way could "pollute" a researcher's perspective and result in features that may overfit to a particular data set, this may be a necessary build generalizable detectors.

Much of our feature selection work relied on the atheoretical approach of simple forward selection that yielded some features that may be only coincidentally correlated with our target affects. The best way to increase fidelity in identifying which features are true expressions of an affective state is to examine which coefficients remain similar in sign and magnitude across detectors built for different data sets. For example, in both confidence models generated, NOTR→BOTT enters into the regression model with a negative coefficient. This means that transitioning from responding to a problem in under four seconds to using a bottom out hint is negatively correlated with confidence, in both models generated over different data sets. Both of these behaviors seem expressions of disengagement, and other potentially disengaged student states like GIVEUP and GUESS also figure largely into both models. Unfortunately, the similarity in these states (as expressions of disengagement) may make the models more different than they need to be as in the case of NOTR→NOTR versus GUESS→NOTR in the case of frustration.

The statistical power of using a larger and therefore likely more diverse data set is evident from our findings. In all cases (with the exception of frustration), the 2009 model outperforms the 2011 when applied to the 2011 data set. The fact that the 2009 data set has about twice as many participants and roughly ten times as many affect reports may explain this trend. Thus, a larger and more diverse data set seems to generalize better to new samples and groups of students.

Finally, it's worth noting that the 2013 models transferred poorly to 2009 and 2011 datasets, and that the 2013 data set came from summer school students from the southwestern United States (Arizona & California). Models trained on the 2009 or 2011 data sets do not appear to generalize to the 2013 data set, or vice versa. We believe this is because the 2013 dataset was unique in several ways: it came from a different region of the country; it corresponded to students working in a summer program as opposed to during a typical school year; a slightly different version of Wayang Outpost was used. In addition, the 2013 students only self-reported on two affective states: excitement and interest, but not confidence or frustration. While batch cross validation may address within sample distinctness between participants, it does little to address how well the model will perform when applied to a distinct new sample group whose participants are distinct from the training group (e.g. summer school vs. not summer school, within a regular math class).

Limitations of generalizability across samples might be the largest challenge, also found in other work. In a recent study [39], detectors trained on student sample groups from urban, suburban, and rural areas were shown to have difficulty generalizing to a different sample group. For example, a detector of Confusion trained on suburban students under batch cross validation achieved a kappa of 0.38 when applied to suburban students, but performed at chance when applied to rural students with a kappa of 0, and only slightly better when applied to urban students with a kappa of 0.06 [39]. This shows that while cross validation may provide a conservative estimate on how well a model may generalize to new data, the accuracy of this estimate is conditioned upon the training data being representative of the population to which the model is to be applied to.

6. ACKNOWLEDGEMENTS

This research was funded by the National Science Foundation, #1324385, Cyberlearning DIP, Impact of Adaptive Interventions on Student Affect, Performance, and Learning; Burleson, Arroyo and Woolf (PIs). Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

7. REFERENCES

[1] Hidi, S. (1990) Interest and Its Contribution as a Mental Resource for Learning. *Review of Educational Research*. 60(4).

[2] Isen, A.M., K. Daubman, and G. Nowicki (1987) Positive affect facilitates creative problem solving. *Journal of Personality and Social Psychology*. 52, 1122–1131.

[3] Pekrun, R., A.J. Elliot, and M.A. Maier (2009) Achievement Goals and Achievement Emotions: Testing a Model of Their Joint Relations With Academic Performance. *Journal of Educational Psychology*. 101(1), 115–135.

[4] Lepper, M. (1988) Motivational Considerations in the study of Instruction. *Cognition and Instruction*. 5(4), 289-309.

[5] D'Mello, S.K., B. Lehman, R. Pekrun, and A.C. Graesser (in press) Confusion Can be Beneficial For Learning. *Learning & Instruction*.

[6] Pekrun, R., T. Goetz, L. Daniels, R. Stupinsky, and R. Perry (2010) Boredom in Achievement Settings: Exploring Control–Value Antecedents and Performance Outcomes of a Neglected Emotion. *Journal of Educational Psychology*. 102(3), 531-549.

[7] Baker, R.S.J.d., S.K. D'Mello, M.M.T. Rodrigo, and A.C. Graesser (2010) Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-Affective States during Interactions with Three Different Computer-Based Learning Environments. *International Journal of Human-Computer Studies*. 68(4), 223-241.

[8] Calvo, R.A. and S. D'Mello (2010) Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE transactions on affective computing*. 1(1), 18-37.

[9] Cooper, D.G., K. Muldner, I. Arroyo, B.P. Woolf, W. Burleson, and R. Dolan (2010) Ranking Feature Sets for Emotion Models used in Classroom Based Intelligent Tutoring Systems. *In Proceedings of International Conference on User Modeling and Adaptive Presentation (UMAP'10)*, 135-146.

[10] Cooper, D., I. Arroyo, B. Woolf, K. Muldner, W. Burleson, and R. Christopherson (2009) Sensors Model Student Self Concept in the Classroom. *In Proceedings of UMAP 2009, First and Seventeenth International Conference on User Modeling, Adaptation and Personalization*, 30-41.

[11] Arroyo, I., D.G. Cooper, W. Burleson, B.P. Woolf, K. Muldner, and R. Christopherson (2009) Emotion Sensors Go To School. *In Proceedings of 14th International Conference on Artificial Intelligence in Education (AIED'09)*, 17-24.

[12] Muldner, K., W. Burleson, and K. VanLehn (2010) "Yes!": Using Tutor and Sensor Data to Predict Moments of Delight

during Instructional Activities. *In Proceedings of User Modeling, Adaptation, and Personalization*, 159-170.

[13] D'Mello, S. and A. Graesser (2007) Mind and Body: Dialogue and Posture for Affect Detection in Learning Environments. *In Proceedings of International Conference on Artificial Intelligence in Education*, 161-168.

[14] Conati, C. and X. Zhou (2002) Modeling Students' Emotions from Cognitive Appraisal in Educational Games. *In Proceedings of ITS 2002, 6th International Conference on Intelligent Tutoring Systems.*,

[15] Conati, C. and H. Maclaren (2009) Modeling User Affect from Causes and Effects. *In Proceedings of UMAP 2009, First and Seventeenth International Conference on User Modeling, Adaptation and Personalization*, 10 pages.

[16] Baker, R.S.J.d., S.M. Gowda, M. Wixon, J. Kalka, A.Z. Wagner, A. Salvi, V. Aleven, G. Kusbit, J. Ocumpaugh, and L. Rossi (2012) Sensor-free automated detection of affect in a Cognitive Tutor for Algebra. *In Proceedings of 5th International Conference on Educational Data Mining*, 126-133.

[17] Efron, B. and G. Gong (1983) A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician*. 37, 36-48.

[18] Arroyo, I., H. Mehranian, and B. Woolf (2010) Effort-based tutoring: An empirical approach to intelligent tutoring. *In Proceedings of 3rd International Conference on Educational Data Mining*, 1-10.

[19] Beal, C.R., R. Walles, I. Arroyo, and B.P. Woolf (2007) On-line tutoring for math achievement testing: A controlled evaluation. *Journal of Interactive Online Learning*. 6(1), 43-55.

[20] Collins, A., J.S. Brown, and S.E. Newman (1989) *Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics*, in *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser* L.B. Resnick, Editor Lawrence Erlbaum Associates: Hillsdale, NJ. p. 453-494.

[21] Vygotsky, L. (1978) *Mind in society*: Harvard University Press.

[22] Arroyo, I. and B. Woolf (2005) Inferring learning and attitudes from a Bayesian Network of log file data. *In Proceedings of 12th International Conference on Artificial Intelligence in Education (AIED'05)*, 33-40.

[23] Arroyo, I., C. Beal, T. Murray, R. Walles, and B.P. Woolf (2004) Web-Based Intelligent Multimedia Tutoring for High Stakes Achievement Tests Intelligent Tutoring Systems. *In Proceedings of 7th Internatinal Conference on Intelligent Tutoring Systems (ITS'04)*, 142-169.

[24] Arroyo, I., K. Ferguson, J. Johns, T. Dragon, H. Meheranian, D. Fisher, A. Barto, S. Mahadevan, and W. B.P. (2007) Repairing disengagement with non-invasive intervention. *In Proceedings of 13th International Conference on Artificial Intelligence in Education*, 195-202.

[25] Mayer, R.E. (2001) *Multimedia Learning* New York: Cambridge University Press.

[26] Arroyo, I., W. Burleson, M. Tai, K. Muldner, and B. Woolf (in press) Gender Differences In the Use and Benefit of Advanced Learning Technologies for Mathematics. *Journal of Educational Psychology*.

[27] Woolf, B., I. Arroyo, K. Muldner, W. Burleson, D. Cooper, R. Dolan, and R. Christopherson (2010) The Effect of Motivational Learning Companions on Low Achieving Students and Students with Disabilities. *In Proceedings of The 10th International Conference on Intelligent Tutoring Systems (ITS'10)*, 327-337.

[28] Arroyo, I., B.P. Woolf, J.M. Royer, M. Tai, K. Muldner, W. Burleson, and D. Cooper, *Gender Matters: The Impact of Animated Agents on Students' Affect, Behavior and Learning*, in *Technical report UM-CS-2010-020*2010, UMASS Amherst.

[29] B. Kort, R.R. and R.W. Picard (2001) An Affective Model of Interplay Between Emotions and Learning: Reengineering Educational Pedagogy-Building a Learning Companion. *In Proceedings of International Conference on Advanced Learning Technologies*,

[30] IBM (2012) *IBM SPSS Statistics for Windows, Version 21.0* Armonk, NY: IBM Corp.

[31] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T. (2006) YALE: Rapid Prototyping for Complex Data Mining Tasks. *In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)*, 935-940.

[32] Cohen , J. (1968) Weighted kappas: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*. 70, 213-220.

[33] Cohen, J. (1960) A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*. 20(1), 37–46.

[34] Rodrigo, M. and R. Baker (2009) Coarse-Grained Detection of Student Frustration in an Introductory Programming Course. *In Proceedings of ICER 2009: the International Computing Education Workshop*,

[35] D'Mello, S., S. Craig, A. Witherspoon, B. McDaniel, and A. Graesser (2008) Automatic Detection of Learner's Affect from Conversational Cues. *User Modeling and User-Adapted Interaction*. 18(1-2), 45-80.

[36] Altman, D.G. and M. Bland (1994) Diagnostic tests 1: sensitivity and specificity. *BMJ*. 308, 1552.

[37] Hanley, J. and B. McNeil (1982) The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology*. 143, 29-36.

[38] Pardos, Z.A., R.S.J.d. Baker, M.O.C.Z. San Pedro, S.M. Gowda, and S.M. Gowda (2013) Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. *In Proceedings of 3rd International Conference on Learning Analytics and Knowledge*, 117-124.

[39] Ocumpaugh, J., R. Baker, S. Gowda, N. Heffernan, and C. Heffernan (in press) Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology*.

The *Problem Solving Genome*: Analyzing Sequential Patterns of Student Work with Parameterized Exercises

Julio Guerra[†], Shaghayegh Sahebi[⋆], Peter Brusilovsky[†], Yu-Ru Lin[†]

[†]School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260, USA
{jdg60, peterb, yurulin}@pitt.edu

[⋆]Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260, USA
shs106@pitt.edu

ABSTRACT

Parameterized exercises are an important tool for online assessment and learning. The ability to generate multiple versions of the same exercise with different parameters helps to support learning-by-doing and decreases cheating during assessment. At the same time, our experience using parameterized exercises for Java programming reveals suboptimal use of this technology as demonstrated by repeated successful and failed attempts to solve the same problem. In this paper we present the results of our work on modeling and examining patterns of student behavior with parameterized exercises using the Problem Solving Genome, a compact encapsulation of individual behavior patterns. We started with micro-patterns (genes) that describe small chunks of repetitive behavior and constructed individual genomes as frequency profiles that show the dominance of each gene in individual behavior. The exploration of student genomes revealed the individual genome is considerably stable, distinguishing students from their peers. Using the genome, we were able to analyze student behavior on the group level and identify genes associated with good and poor learning performance.

Categories and Subject Descriptors

Information systems [**Information Systems Applications**]: Data mining

Keywords

sequential pattern mining, parameterized exercises

1. INTRODUCTION

Parameterized exercises have recently emerged as an important tool for online assessment and learning. A parameterized exercise is essentially an exercise template that is instantiated at runtime with randomly generated parameters. As a result, a single template is able to produce a large number of similar, but distinct questions. While parameterized questions are considerably harder to implement than traditional "static" questions, the benefits offered by this technology make this additional investment worthwhile. During assessment, a reasonably small number of question templates can be used to produce online individualized assessments for large classes minimizing cheating problems [12]. In a self-assessment context, the same question can be used again and again with different parameters, allowing every student to achieve understanding and mastery. The aforementioned properties of parameterized exercises made them very attractive for the large-scale online learning context. At the same time, parameterized exercises as a learning technology have their own problems. Our experience with personalized exercises for SQL [17] and Java [7] in the self-assessment context demonstrated that the important ability to try the same question again and again is not always beneficial, especially for students who are not good at managing their learning. The analysis of a large number of student logs revealed some considerable number of unproductive repetitions. We observed many cases where students kept solving the same exercise correctly again and again with different parameters, well past the point when it could offer any educational benefit. While it might increase self-confidence, students' time and effort might be spent better by advancing to more challenging questions. We also observed cases where students persisted in failing to solve the same, too difficult exercise, instead of focusing on filling the apparent knowledge gap or switching to simpler exercises.

The work presented in this paper was motivated by our belief that the educational value of parameterized exercises could be increased by a personalized guidance mechanism that can predict non-productive behavior and intercept it by recommending a more efficient learning path. The main challenge with predicting unproductive behavior is to examine the stability of behavior patterns in the problem solving process. If the patterns, such as specific unproductive sequences, appear at random, there is a slim chance to predict and prevent them. If, on the contrary, specific patterns are associated with certain features of the student (such as knowledge and individual traits), exercise complexity, or the learning process stage, there is a good chance to learn the association rules and use it for prediction. In this paper we performed an extended study of problem solving patterns in the context of parameterized exercises. We explored the connection between these patterns and the components of the learning process mentioned above. Our study produced a rather unusual result. While it was more plausible to expect that the patterns are related to the current level of student knowledge, our analyses revealed that the patterns are related to student problem solving tendency. More exactly, we discovered that every student has a specific combination of micro-patterns, a kind of problem solving genome. We observed that this genome is relatively stable, distinguishing every student from his or her peers; it changes very little with the growth of the student knowledge over the course. We also discovered that genomes are not randomly distributed, and instead, students with similar genomes form cohorts that perform relatively similarly in the problem solving process. We believe that our discovery of the problem solving genome is a very important step toward our goal of predicting and preventing unproductive behavior. Indeed, the stability of patterns on the personal level makes the task of pattern prediction feasible while the presence of cohorts opens the way to detect the student problem-solving genome early in the learning process. In this paper we present our approach of detecting student problem-solving genome and report our exploration of the genome on the level of individual students and cohorts.

The rest of the paper is structured as follows. The next section briefly reviews several areas of related work. Section 3 describes the dataset used in the study. Section 4 presents the method for building the Problem Solving Genome. In Section 5 we explore the Genome's stability and its relation with performance groups and the complexity of the exercises. Section 6 summarizes the contribution and discusses future work.

2. RELATED WORK

2.1 Parameterized Questions and Exercises

Recent studies in educational technology have demonstrated promising results by leveraging computer and Web abilities to deliver parameterized exercises worldwide, which has become one of the focusing topics in Web-enhanced education. One of the most influential systems, CAPA [9], was evaluated in a number of careful studies, providing clear evidence that individualized exercises can significantly reduce cheating while improving student understanding and exam performance. The CAPA technology has been later integrated into the popular LON-CAPA platform [12] and its functionality defined the assessment architecture of the MOOC platform eDX [1]. Due to the complexity of parameterized assessment, the majority of work on parameterized questions and exercises was done in physics and other math-related domains where a correct answer to a parameterized question can be calculated by a formula. There are, however, examples of using this technology in other domains. In particular, our team focused on parameterized exercises for teaching programming. We developed and explored the QuizPACK platform for C-programming [3] and the similar QuizJET platform for Java programming [7]. Problem solving repetition behaviors have been studied by psychologists in different ways, providing evidence that repetition behaviors have roots in cognitive, metacognitive and motivational aspects and explaining why some students quit and others persist when facing challenging problems [14]. Schunk [16] shows a positive correlation between persistency in repeating and *self-efficacy* (believe on self-capabilities to solve a problem). The attribution theory [19] describes how students that attribute performance outcomes (successes, failures) to effort tend to work harder than students who attribute them to ability. Grounded in the literature in educational psychology, we conjecture that patterns on problem solving repetition may be explained by individual learners' motivational traits that are part of learners' personality [15]. These theories provide insights into analyzing to what extent these behaviors are stable in students.

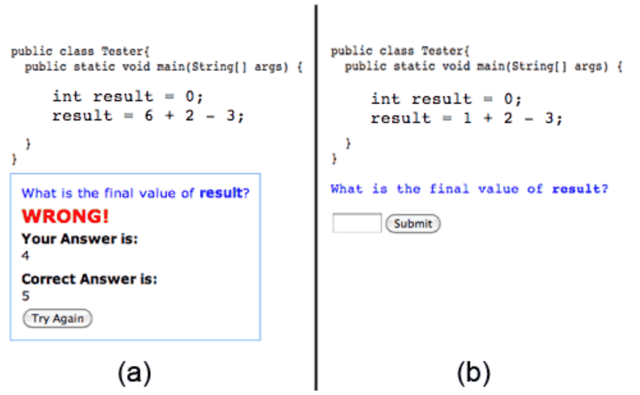
2.2 Sequential Pattern Mining in Education

Mining sequential patterns of students actions has recently gained attention in educational data mining field. Using activity data collected from groups of students working with interactive tabletops, Martinez et al [13], mined and clustered frequent patterns to compare distinct behaviors between low and high achievement groups. The differential sequence mining method, introduced by Kinnebrew and Biswas [11] has been successfully used to differentiate behavioral patterns among groups of students (such as low and high performance students). The method uses *SPAM* [1] to find common patterns in the sequences of the whole dataset, and then applies statistical tests to reveal differences in the frequencies of the discovered patterns among different groups. The same authors have applied this technique in data collected from the system Betty's Brain to discovered patterns that can distinguish self-regulated behaviors in successful and non-successful students [2] and to analyze the evolution of reading behaviors in high and low performance students during productive and non-productive phases of work [10]. Herold, Zundel and Stahovich [4] have used the differential sequence mining on sequences of actions on handwritten tasks and proposed a model to predict performance on the course based on pattern features. Our work extends this prior work by utilizing and aggregating the mined sequence patterns to construct student activity profiles. Such profiles enable us to evaluate the statistical differences at the student, exercise, and group levels.

3. SYSTEM AND DATASET

We collected answers of students who worked with QuizJET [7] parameterized Java exercises in the context of an introductory object-oriented programming class at the School of Information Sciences in the University of Pittsburgh. The students accessed the exercises through the Progressor+ interface [6]. The system was provided for self-study and its use was not mandatory. Each QuizJET exercise was generated from a template by substituting a parameter variable with a randomly generated value. Exercises generated using the same template were equal from a semantics point of view. To answer the exercise the student had to mentally execute a fragment of Java code to determine the value of a specific variable or the content printed on a console. When the user answers, the system evaluates the correctness, reports to the student whether the answer was correct or wrong, shows the correct response, and invites the student to "try again". Next time, the exercise is be generated with other values and the correct answer will be different. In this way, the student can try the same exercise many times, leaving a trace of successes and failures. Figure 1 shows a simple parameterized

[1] http://www.edx.org/



```
public class Tester{
  public static void main(String[] args) {

     int result = 0;
     result = 6 + 2 - 3;

  }
}
```

What is the final value of **result**?

WRONG!

Your Answer is:
4

Correct Answer is:
5

```
public class Tester{
  public static void main(String[] args) {

     int result = 0;
     result = 1 + 2 - 3;

  }
}
```

What is the final value of **result**?

Figure 1: A parameterized problem in QuizJET. In (a) the student answers wrongly and then hits "Try Again" button. In (b) the problem is reloaded with different numbers.

Java problem answered incorrectly by the student (a) and then repeated (b). Note the differences in the numbers in the second attempt (b) which correspond to the same problem. Progressor+ provided access to 103 different parameterized exercises organized in 19 topics (Variables, Objects, Arrays, etc.). Exercises are labeled in terms of complexity and there are 41 *easy* exercises, 41 *medium* exercises and 19 *hard* exercises.

The dataset includes three semesters of student data (Spring 2012, Fall 2012 and Spring 2013) in which the use of the system was optional. Overall, 101 students used the system making 6489 incorrect and 14726 correct attempts. Easy exercises were attempted 10620 times, medium complexity exercises were attempted 7876, and hard exercises were attempted 2719 times. Once a student started to work with an exercise she might attempt it just once or try it several times in a sequence. The dataset includes 4212 single attempts (no repetition) and 4758 sequences with more than one attempt. Among these there are 2717 with more than two attempts, 1583 with more than three attempts, and 1016 sequences with more than four attempts.

4. BUILDING THE PROBLEM-SOLVING GENOME

The key idea of our "genome" approach is to build a compact characteristics of student problem-solving behavior on the level of micro-patterns. To build a genome we started by finding proper micro-patterns (genes) and then built a genome of a student as a vector representing the frequencies of different micro-pattern occurrences in the student problem-solving logs. An overview of the genome-building process is shown in Figure 2. To build the genes, we started by labeling students' attempts using time and correctness (Figure 2(a), Section 4.1). We then apply sequential pattern mining to extract sequential micro-patterns Figure 2 (b), Section 4.2). The most frequent micro-patterns were selected as *genes* and used as a basis for the *Problem Solving Genome*, which is a vector of gene frequencies (Figure 2(c), Section 4.3). This section presents the genome-building process in detail while the next sections report our exploration of the Genome.

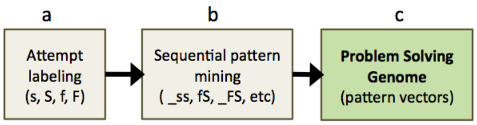

a
b
c
Attempt labeling (s, S, f, F)
Sequential pattern mining (_ss, fS, _FS, etc)
Problem Solving Genome (pattern vectors)

Figure 2: Steps for building the Problem Solving Genome.

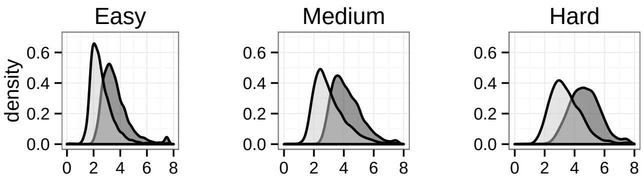


Figure 3: Time distributions (logarithmic) for easy, medium and hard exercises. The right curve is always the first attempt time distribution, showing that first attempts usually take longer times.

4.1 Attempts labeling

We use both time and correctness of each attempt to label it for further use in sequential pattern mining analysis. In this way, each action will convey more information than using correctness only. As shown in Figure 3, distribution of times for first attempts are different from other (non-first) attempts. This is reasonable if we consider that the user needs extra time the first time to read and understand the exercise. Additionally, time distribution is different for different exercises, as in general, complex exercises need longer times. Thus, for labeling the time factor, we used time information of historical records in our system to compute the median times for each exercise for both first and other attempts. Then, we labeled the attempt as *short* or *long* depending on the time being shorter or greater than the median of the distribution for the specific exercise. Combining correctness and time, we finally label the attempts using the letters 's' (lowercase s) for a short success, 'S' (uppercase S) for a long Success, 'f' for a short failure, 'F' for a long Failure.

The labeled attempts are organized in sequences by pairs student-question within a session in the system. Each sequence $s_{u,e}$ represent the sequential attempts of user u in the exercise e within a session. If the user attempted the same exercise in different sessions, there will be more than one sequence $s_{u,e}$. Additionally, we mark starting and ending points on sequences using '_' (underscore). For example, a sequence *_fSs_* means start with a short failure, make a long success and then finish with a short success.

4.2 Sequential pattern mining

To discover frequent patterns, we use the PexSPAM algorithm [5], which extends the fast SPAM algorithm [1] with gap and regular expression constraints. Given a sequence database $D = s_1, s_2, ..., s_n$, the support of a pattern α is the proportion of sequences of D which contains α as a subsequence at least once. If the support of α is bigger than a threshold, then α is considered a frequent pattern. Support measure does not inform for multiple occurrences of the pattern within a sequence. In this work, we set a small minimum support in 1% because even when a pattern oc-

Table 1: Top 20 patterns (genes) ordered by support (the percentage of sequences that contain the pattern). Observe the presence of many inefficient patterns like 'ss' or 'FF' among top 20.

	Pattern	Support		Pattern	Support
1	`ss_`	0.163	11	`_FS`	0.07
2	`ss`	0.107	12	`FS`	0.066
3	`Ss`	0.101	13	`FS_`	0.060
4	`SS_`	0.091	14	`FF`	0.059
5	`_FS_`	0.086	15	`SS`	0.058
6	`_FF`	0.083	16	`_SS`	0.054
7	`Ss_`	0.081	17	`_ss_`	0.053
8	`_fS_`	0.079	18	`_SS_`	0.052
9	`_fF`	0.077	19	`sss`	0.050
10	`sss_`	0.074	20	`_fS`	0.048

curs in overall few sequences, it can still make a difference when looking at the aggregation of pattern occurrences by student. Additionally, since we are interested in looking at patterns of 2 or more sequential attempts, we set the gap to 0 and considered only sequences with more than one attempt. After running the mining algorithm, we discover 102 common patterns occurring at least in 1% of the sequences. These common micro-patterns of student behavior play the role of genes in our approach. The top 20 genes and the corresponding support can be seen in Table 1.

4.3 The problem solving genome: characterizing students with pattern vectors

Using the 102 gene patterns discovered by the sequential pattern mining, we build individual frequency vectors that show how frequently each gene appears in student problem solving behavior. Since this vector captures in a compact form the specifics of student problem solving behavior, we call it student *Problem Solving Genome*. Note that the frequency-based approach allows building individual genome using any subset of gene sequences, for example, all sequences in the term, the first half of sequences of the student activity in the term, a random subset of sequences, etc. Since a pattern might occur more than once in a sequence, and more than one pattern may occur in a sequence, the frequency vectors are not summing to 1. Thus, we normalize the vectors for further analysis.

5. EXPLORING THE GENOME

In this section we analyze the pattern vectors within and between students, across problem complexity levels, and across different student performance groups. We use the same dataset for all further analysis: we select 68 students having pretest/posttest (see section 5.3.1) and a minimum amount of usage of the system of 20 sequences and two sessions. Addtionally, we exclude one outlier student with a very unusual number of repetitions in the first 6 sequences. At the end our dataset consists of 67 students.

5.1 Problem Solving Genome stability

The first step of problem-solving genome exploration is assessing its stability. To what extent the name "genome" that we assigned to the micro-pattern frequency vector is justified? Is it just a random mix of pattern which could be different for different time slots or, like a real genome, it is a stable characteristic of a user that distinguishes him or her from peers? A good approach to check genome stability is to randomly split sequences of user activity patterns into two equal sets and build the genome vector from each of two halves. If the genome is stable, then two random halves of the split genome should be significantly closer to each other than to half-genomes of other users. In contrast, if genome halves are no closer to each other than to half-genome vectors of other users, we can't consider genomes as stable user characteristics. To assess the stability hypothesis we built two half-genomes for each user by randomly splitting her observed sequences in half and compiling gene frequency vectors for each half. We then calculate pairwise distances between all half-genomes.

To compute distances, we use Jensen-Shannon (JS) divergence as it is a symmetric version of Kullback-Leibler divergence and has been widely used for computing distance between frequency distributions. We filter out all students with less than 60 sequences, limiting differences due to extreme difference on amount of activity. Among the 67 students in our dataset, there are 32 students with at least 60 sequences. In this analysis we use a paired samples *t-test* on the difference between the self and other distances. The normality assumption is met. Results are shown in Table 2 first row (a). Students self-distances are significantly smaller ($M = .2370$, $SE = .0169$) than distances to other students ($M = .4815$, $SE = .0141$), $t = -15.224$, $p < .001$, Cohen's $d = 2.693$.

While similarity of random half-genomes is a very strong argument in favor of genome stability, the random split has one weak aspect: since each of the random halves represents student micro patterns over the whole duration of the course, it is still possible that the student genome gradually changes over the course duration from one pattern frequency to another. To assess the temporal stability of the genome we need to use temporal split, i.e., to compare half-genomes built from the temporally first half (early) and second half (late) of student sequences. Results in Table 2 second row (b) confirm the temporal stability hypothesis: while the distance between temporary split half-genomes is larger than between randomly split halves ($M = .3211$, $SE = .0214$) it is still significantly smaller than between-student distances ($M = .4997$, $SE = .0164$), $t = -6815$, $p < .001$, Cohen's $d = 1.205$. This result confirms that frequencies of micro-pattern appearances act as a true problem solving genome "genome": it is considerably stable, characterizing each user as individual over the course progression, while sufficiently distinguishing this user from others.

5.2 Effect of complexity

While we discovered that the knowledge level and course stage doesn't affect the genome, it is still possible that behavior patterns are affected by exercise complexity. To understand how the complexity level of the exercises impacts on the pattern frequencies, we analyze distances between the genome of the exercises (i.e. pattern frequency vector for each exercise). Having the exercises' genome and the predefined classification in *easy*, *medium* and *hard*, we select pairs of exercises within and between complexity levels and compute distances using Jensen-Shannon divergence. We filter out all questions with less than 20 sequences and

Table 2: Statistical tests comparing students with themselves and others.

		self distances		dist. to others				
		M	SE	M	SE	t	*sig.*	Cohen's d
a)	randomly split genome	.2370	.0169	.4815	.0141	-15.224	$<$ .001	2.693
b)	early/late genome	.3211	.0214	.4997	.0164	-6.815	$<$ .001	1.205
c)	randomly split genome in easy exercises	.3736	.0214	.6065	.0128	-10.352	$<$ .001	1.657

Table 3: Mean and standard error of distances within and between easy and hard exercises.

	Mean	SE
within easy	.3311	.0031
within hard	.3478	.0085
between easy-hard	.4145	.0050

perform comparisons between extremes groups, i.e. *easy* and *hard* complexity levels to extreme the differences. Normality and homogeneity of the variance on pair distances are not met on all levels, thus a non-parametric test is applied. Results of the Krustal-Wallis test shows significant differences between distances within and between levels, $\chi^2(2, N = 1596) = 160.359$, $p < 001$. Mean and standard error of distances within easy, within hard, and between easy and hard groups are shown in Table 3. A Mann-Whitney test is performed to test differences among the levels. Distances within easy exercises (mean rank = 626.16) are significantly smaller than distances between easy and hard exercises (mean rank = 909.77), $z = -12.564$, $p < .001$. Similarly, the distances within hard exercises (mean rank = 277.20) are significantly smaller than distances between easy and hard exercises (mean rank = 383.13), $z = -4.733$, $p < .001$. These results show a clear dependency of the pattern behaviors with the complexity level of the questions. This is reasonable given that hard questions, which need more time, are expected to discourage repetitions.

The impact of exercise complexity on the patterns suggest that the genome is as much impacted by the unique exercise difficulty profile than by individual differences of the students. We re-examine the analysis on Section 5.1 now considering randomly split genome built only from activity on easy exercises, to control for differences of students amount of activity on different complexity exercises. We perform this analysis with 39 students having at least 20 sequences in easy questions. Results shown in last row (c) in Table 2 confirm the stability of patterns: students are more similar to themselves (self distance $M = .3736$, $SE = .0214$) than to others (distances $M = .6065$, $SE = .0128$), $t = -10.352$, $p < .001$, Cohen's $d = 1.6569$, even within exercises of the same complexity.

5.3 Patterns of Success within student groups

Since one of the goals of this paper is using behavior analysis to identify and prevent inefficient patterns, it would be valuable to use the genome to identify which patterns make groups of students more or less successful in the learning process. The easiest approach to do it is to split students into performance-related groups and find unique genome aspects in this group. This simple approach, however, might not work since for students with very different genomes, different behavior patterns might be related to success. In this case, to find a connection between patterns and performance, we should group students into groups with similar behavior and contrast most and least successful students within each group. In this section we perform both kinds of the analysis.

Table 4: Number of students in each predefined performance group (PPG).

	Pretest (total=67)	Posttest (total=65)	Learning gain (total=65)
low	24	22	22
medium	16	19	20
high	27	24	23

5.3.1 Patterns for Predefined Performance Groups

Predefined Performance Groups (PPG) are defined based on pre and posttest scores that we collected. The pre and posttest were highly similar among different semesters (small variation on questions) and the scores were further normalized as score / max_score (min score is 0). Additionally, we compute a normalized learning gain score as (normalized post score) - (normalized pre score). For each of the pretest, posttest, and learning gain measures, students are classified in three groups using the percentiles 33.3 and 66.7: *low*, *medium* and *high*. For example, a student with pretest lower or equal than the percentile 33.3 in the pretest score distribution is classified as *low* pretest student. Summarizing, we have 3 PPG (low, medium, high) for each performance measure (pretest, posttest and learning gain). As explained before, the dataset contains 67 students with pretest and 65 students with both pre and posttest. Table 4 shows the number of students in each PPG.

Do students with similar performances have similar patterns for solving parameterized exercises? Is this similarity, between the students of the same predefined performance group, more than the similarity we can find between the students from different groups? For this analysis we contrast the genome built using all the term activity (all problem solving sequences) of the students classified in the performance groups described before. We sample 50% of all possible pairs of students within and between PPGs and compute the distances (Jensen-Shannon divergence) of all within and between group pairs. Then, we compare the average of distances within and between groups to see if students inside each group are more similar to each other than to students in other groups. Normality and homogeneity of variance is not met for all groups, thus we use Krustal-Wallis non-parametric mean rank test and Mann-Whitney test for single comparisons. We constrained the analysis to PPGs *low* and *high* to see extreme differences.

Results are shown in Table 5. Mann-Whitney comparison is reported only where significant differences among groups were found (pretest). For pretest groups, distances within the low group (mean rank = 222.70) are significantly smaller than distances between low and high groups (mean rank =

Table 5: Statistical tests on differences on distances between pairs of students within low, within high, and between low and high PPGs.

	low		high		low-high		Krustal-Wallis test			Mann-Whitney test		
	M	SE	M	SE	M	SE	Mean Ranks (low,high,low-high)	χ^2	*sig.*	Mean ranks	*z*	sig.
Pretest	.465	.014	.547	.017	.512	.010	294.68, 368.67, 341.51	11.926	.003	222.70, 258.21 low < low-high	-2.537	.011
Posttest	.486	.016	.516	.018	.511	.011	256.41, 271.97, 273.69	1.061	.588	-	-	-
L. Gain	.507	.019	.470	.018	.517	.013	242.32, 216.57, 251.35	5.276	.071	-	-	-

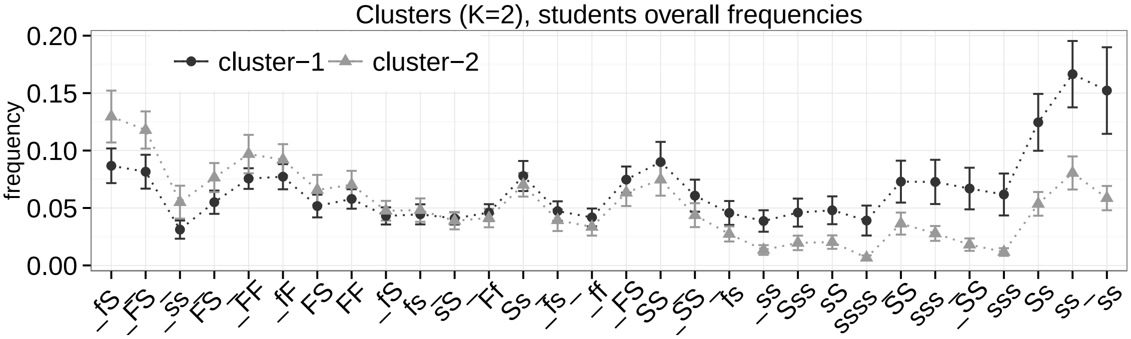


Figure 4: Top 30 patterns and their frequencies in each cluster. Patterns are ordered by the difference on frequencies between cluster two (non-confirmers) and one (confirmers).

258.21), $z = -2.537$, $p = .011$. This suggests that student with no previous experience tend to behave differently than students with stronger background. There is no significant difference between high and low-high distances, though, meaning that the high group behaved more heterogeneously than low group. For posttest and learning gain groups there are no significant differences on distances within and between groups. These results are intriguing, as we will expected to find clear differences among performance groups. Since we could not find those differences, we could hypothesize that specific behavior patterns can't be easily characterized as universally helpful or harmful for student performance, instead, the impact of each micro-pattern on student behavior might depend on the whole profile of micro-patters, i.e., the genome. Thus, to find connections between genome and performance, we need to start from the opposite side: cluster the students based on the genome, characterize the clusters in terms of the distinguishable patterns, and find helpful and harmful patterns within each class. We describe these analyses in the following sub sections.

5.3.2 *Clustering students by their genome*

We use the genome as a feature vector and cluster students using the spectral clustering technique [18] as it gives a better separation of the students. We choose two clusters (K=2) as we observe that two clusters give the largest eigen-gap, suggesting there are two intrinsic groups in the data. Figure 4 shows the top 30 frequent patterns in both of the clusters. Each point represents the average frequency of seeing a particular pattern in the cluster. Error bars are included to indicate significance. We order the patterns in x-axis by the differences between clusters two and one. As we can see in this figure, some of the patterns, such as *_fS_*, *_FS_*, *_ss*, *Sss*, etc., occur with significant frequency difference in the two clusters and some other patterns, such as *_fS*, *fs_*, *Ff*, etc., do not show significant differences. If we look more closely, the sequences that start with failure are mostly related to the students in cluster two and the sequences that start with success are mostly related to the students in cluster one. Also, we can see that the students in cluster one tend to repeat their successful attempts more and more frequently (e.g. the *ssss_* pattern). In other words, even when they get the right answer to the question, they will insist on confirming knowing the question by repeating it again and again. Unlike students in cluster one, the students in cluster two are much less prone to this "confirmation" behavior. Instead, they are more prone to stop working with an exercise early, frequently right after figuring out the first right answer to the question, even if they have struggled for the correct answer in their previous attempts (e.g. *_fS_*, *_FS_*, and *FS_* patterns). Thus, using the student genome, we can identify two major types of student behaviors in solving parameterized exercises. Based on these observations above, we call the first cluster of the students *the confirmers* and the second cluster *the non-confirmers*.

5.3.3 *Performance differences among clusters*

Once two clusters of students that are similar in their overall behavior are identified, we can re-examine the connection between student success and behavior patterns on the cluster level. We study pattern by pattern differences between different PPGs within each cluster and describe the patterns that distinguish them. Both of the clusters have students from all PPGs. As a result, we cannot say that the student's genome has a direct impact on the performance of the student. Both *confirmers* and *non-confirmers* can have high or low performance. To look at the clusters deeply and to see if there are any differences in the patterns, within each cluster, that can drive students' performance, we repeat the first analysis within each cluster looking at the learning gain. For each of the clusters, we look at the patterns and the difference between their average frequencies for the students

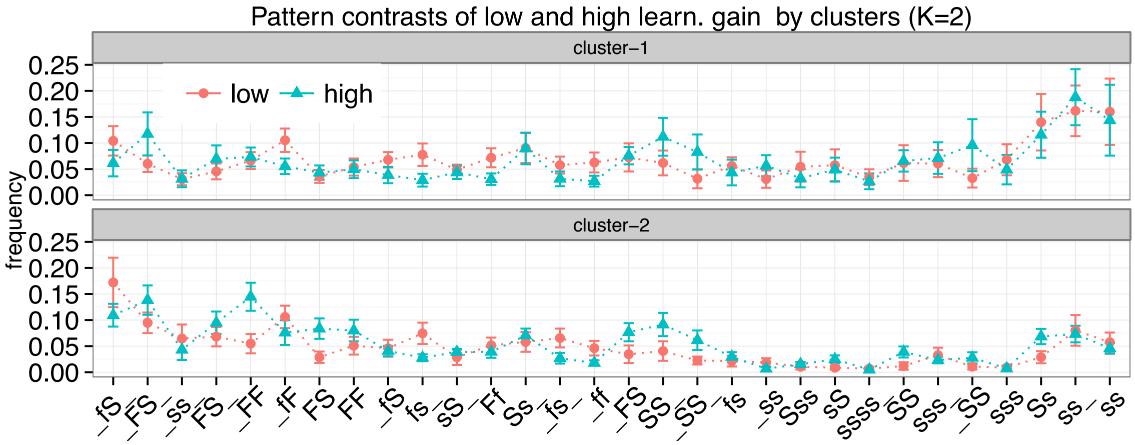


Figure 5: Top 30 patterns and their frequencies for low and high learning gain PPG by cluster.

with low and high learning gain. The result is shown in Figure 5. The upper diagram shows the students in cluster one (the confirmers) and the lower diagram shows the students in cluster tow (the non-confirmers). The red line with round markers show the pattern frequencies for low learning gain students and the blue line with the triangle marker is representative of high learning gain students.

If we look at the patterns in cluster one (the confirmers), we can see that there are some patterns that show significant difference between the low and high learning gain students. Each of these patterns starts with a failure: *_FS_* and *Ff* have long failures in the beginning of the patterns and *_fF*, *fs_*, and *_ff*, have short failures at the beginning of the patterns. Among these patterns, only *_FS_* is practiced more by the high learning gain students. This indicates that, among the confirmer students, the ones that put a good amount of effort to answer a question correctly after a long failure and stop repeating the same question learn more. The low learning gain group shows more frequent use of the *Ff*, *_fF*, *fs_*, and *_ff* patterns. The common element of all of these patterns is short failure (f). If we look at Figure 5 for confirmers, we can see that all of the patterns that include a short failure, are practiced more by the low gain students. This can indicate that the low gain confirmer students do not spend enough time and thought on the questions to which they do not know the answer.

The non-confirmers show more pattern differences between the low and high learning gainers. We can see that the high learning gain group follow the patterns of *_FF*, *FS*, *_FS*, *SS_*, *_SS*, *SS*, and *Ss* more frequently. This means that the high learning gain, non-confirmer students tend to continue trying a non-parameterized exercise and spending time on it after they failed in it or it took them a long time to get to the correct answer for that exercise. In this sense, these students are closer to the confirmer group of students (cluster 1) but only at the times that they are not sure if they have learnt the solution to an exercise. On the other hand, the low learning gain group tend to develop the *fs_*, *_fs_*, and *_ff* patterns in their sequences. The first two indicates that they give up practicing the exercise after having a short success that comes after a short failure. Also, they tend to repeat short failures on the same exercise more often.

Comparing beneficial and harmful patterns for the two clusters, we can make an interesting observation that the increased use of several beneficial patterns for each cluster make students more similar to the opposite cluster. For example, while confirmers have a generally low tendency to stop after first hard success *_FS_*, successful confirmers demonstrate this pattern much more frequently. On the other hand, while non-confirmers generally tend to stop after first hard success, successful non-confirmers have higher tendency to continue after hard success as shown by significantly increased frequencies of such patterns as *SS_*, *_SS_*, and *Ss*. In other words, while the two clusters are considerably different by their behavior overall, the "centrist" students that are closer to the opposite cluster tend to be more successful, while the extreme behavior that distinguishes the cluster is frequently related to less successful performance.

6. CONCLUSIONS AND FUTURE WORK

In this paper we explored patterns of student repetitive work with parameterized exercises for Java programming domain. The goal of this work was to understand the connections between micro- and macro-level behavior patterns and factors that might be responsible for this behavior such as exercise difficulty, student personality, level of knowledge, or position in the course. In turn, we hoped that this understanding could help us predict how a specific student would work with an exercise and prevent inefficient behavior such as repetitive successful attempts when the exercise become too easy to contribute to student knowledge growth. To explore the impact of students' personal features on their work with exercises, we built the student *problem solving genome*, a compact representation that encapsulates the specifics of individual behavior patterns. To build the genome, we started with micro-patterns (genes) that describe small chunks of repetitive behavior based on correctness and duration of each attempt. We then constructed a genome as a frequency profile that shows the dominance of each gene in the student behavior.

Using the genome approach we analyzed the stability of behavior patterns for students and groups and explored their connection with student success in the course. The most interesting finding was a considerable stability of the genome on individual level. As our analysis showed, the genome uniquely identifies a user among other users over the whole duration of the course despite a considerable growth of student knowledge over the course duration. While the problem complexity does affect the behavior patterns as well, we demonstrated that the genome is defined by some inherent characteristics of the user rather than a difficulty profile of the problems she solves.

To find a connection between the problem-solving genome and student performance, we examined genomes for various groups of students. Since a direct attempt to associate genome with performance-related groups (a typical way to group students in educational contexts) was not successful, we started from the opposite side and formed student groups on the basis of their genome similarity. As it appears, all students could be most reliably split into just two cohorts that differ considerably by their behavior. After that split, we were able to contrast successful and less successful learners by their behavior and identify "beneficial" and "harmful" genes for each cohort. In particular, it was interesting to observe that the behavior of successful learners in one cohort was somewhat closer to the behavior of the opposite cohort.

Note that the reported finding are limited to a specific context - non-mandatory work with Java programming exercises. It is not clear whether problem solving behavior patterns in other domains or the same domain with mandatory exercises will exhibit the same properties. We also believe, however, that the "genome" approach provides a new way for exploration of student problem solving behavior and plan to explore to the stability of the "genome" and the presence of behavior cohorts in other domains and contexts. In addition, we would like to proceed to our ultimate goal of recognition and prediction of inefficient behavior. The discovery of a stable genome provides a good ground for developing a recognition engine and the presence of behavior cohorts indicates that some good guidance (encouraging "beneficial" patterns and discouraging "harmful" ones) could be provided even in the early stage of student work when it might be harder to build a reliable genomic profile.

7. ACKNOWLEDGMENTS

Julio Guerra is supported by Chilean Scholarship (Becas Chile) from the National Commission for Science Research and Technology (CONICYT, Chile), and the Universidad Austral de Chile.

8. REFERENCES

[1] Ayres J, Flannick J, Gehrke J, Yiu T (2002) Sequential pattern mining using a bitmap representation. KDD 2002, 429-435

[2] Bouchet, F., Kinnebrew, J. S., Biswas, G., and Azevedo, R. (2012). Identifying Students' Characteristic Learning Behaviors in an Intelligent Tutoring System Fostering Self-Regulated Learning. In EDM (pp. 65-72).

[3] Brusilovsky, P. and Sosnovsky, S. (2005). Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack. ACM Journal on Educational Resources in Computing, 5(3):Article No. 6, 2005.

[4] Herold, J., Zundel, A., and Stahovich, T. F. (2013). Mining Meaningful Patterns from Students' Handwritten Coursework. In EDM 2013 (pp 67-73).

[5] Ho, Joshua, Lior Lukov, and Sanjay Chawla. Sequential pattern mining with constraints on large protein databases. In Proc. of COMAD 2005b, pp. 89-100. 2005.

[6] Hsiao, I-H. and Brusilovsky, P. (2012) Motivational Social Visualizations for Personalized E-learning, In: Proc. of ECTEL 2012, Springer-Verlag, Volume 7563/2012, pp.153-165.

[7] Hsiao, I. H., Sosnovsky, S. , and Brusilovsky, P. (2009). Adaptive navigation support for parameterized questions in object-oriented programming. In ECTEL 2009, volume 5794 of LNCS, pages 88-98. Springer-Verlag.

[8] Kashy, D. A., Albertelli, G., Ashkenazi, G., Kashy, E., Ng, H.-K., and Thoennessen, M. (2001). Individualized interactive exercises: A promising role for network technology. In 31st ASEE/IEEE Frontiers in Education Conference. IEEE, 2001.

[9] Kashy, E., Thoennessen, M., Tsai, Y., Davis, N. E., and Wolfe,S. L. (1997). Using networked tools to enhanse student success rates in large classes. In 27th ASEE/IEEE Frontiers in Education Conference, volume I, pages 233-237. Stipes Publishing L.L.C., 1997.

[10] Kinnebrew, J. S., and Biswas, G. (2012). Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution. In EDM (pp. 57-64).

[11] Kinnebrew, J. S., Loretz, K. M., and Biswas, G. A. U. T. A. M. (2012). A contextualized, differential sequence mining method to derive students' learning behavior patterns. Journal of Educational Data Mining.

[12] Kortemeyer, G., Kashy, E. , Benenson, W., and Bauer, W. (2008). Experiences using the open-source learning content management and assessment system lon-capa in introductory physics courses. American Journal of Physics, 76(438), 2008.

[13] Martinez, R., Yacef, K., Kay, J., Al-Qaraghuli, A., and Kharrufa, A. (2011). Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In EDM 2011 (Vol. 13, No. 2, pp. 111-120).

[14] Mayer, R. E. (1998). Cognitive, metacognitive, and motivational aspects of problem solving. Instructional science, 26(1-2), 49-63.

[15] McAdams, D. P. (1995). What Do We Know When We Know a Person? Journal of Personality, 63(3), 365-396.

[16] Schunk, D. (1991). Self-efficacy and academic motivation. Educational Psychologist 26: 207-231.

[17] Sosnovsky, S., Brusilovsky, P., Lee, D. H., Zadorozhny, V., and Zhou, X. (2008) Re-assessing the Value of Adaptive Navigation Support in E-Learning. In proc. of AH 2008, Springer Verlag, pp. 193-203

[18] Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing, 17(4), 395-416.

[19] Weiner, B. (1986). An Attributional Theory of Motivation and Emotion. New York: Springer-Verlag.

Trading Off Scientific Knowledge and User Learning with Multi-Armed Bandits

Yun-En Liu[1], Travis Mandel[1], Emma Brunskill[2], and Zoran Popović[1]
[1]Center for Game Science, Computer Science & Engineering, University of Washington
[2]School of Computer Science, Carnegie Mellon University
{yunliu, tmandel, zoran}@cs.washington.edu, ebrun@cs.cmu.edu

ABSTRACT

The rise of online educational software brings with it the ability to run experiments on users quickly and at low cost. However, education is a dual-objective domain: not only do we want to discover general educational principles, we also want to teach students as much as possible. In this paper, we propose an automatic method for allocating experimental samples, based on multi-armed bandit algorithms, that balances between learning each experimental condition's effectiveness and users' test performances. Our algorithm, UCB-Explore, allows the experimenter to explicitly specify the tradeoff these two objectives. We assess the performance of our algorithm in a simulated experiment with parameters drawn from a real-world data. In this simulation, our algorithm is better able to navigate this trade off compared to other multi-armed bandit algorithms such as UCB1 and ϵ-greedy. As an example application, we show how a researcher could use the generated samples to identify strong and weak interaction effects, and confirm these findings on a separately-collected dataset.

Keywords

Multi-armed bandits, automatic experimentation, scientific discovery

1. INTRODUCTION

The rise of online educational software has greatly increased the amount of data available to researchers in the educational data mining community. Indeed, both the games and e-commerce industries have already been transformed by the introduction of A/B testing to understand how users react to different software designs and incentives. These tests are traditionally done in a staged manner: run a test, then choose the best option to deploy to all users. But why separate the test phase and the optimized software? Just as systems like Assistments promise to both educate and assess simultaneously [20], we should both educate and experiment at the same time. With programmatic control of educational material and automatic data collection, such systems could provide ever more effective educational experiences while also generating scientific knowledge as they collect data about the comparative effectiveness of different representations of knowledge or teaching strategies. This knowledge could then be used to inform educational theories, potentially producing new and better options to be investigated by the system.

However, this scientific objective is complicated by the fact that in many educational contexts we are working in a high-stakes domain: users of educational software should learn from the system, so testing ineffectual conditions can cause real harm. We want an approach that allows the experimenter to explicitly specify the relative worth of teaching players (by giving out the best conditions) against the gain of scientific knowledge (by giving out sub-optimal conditions to better assess their worth). Then the algorithm can sample the different experimental conditions, with a bias in favor of finding and exploring the better ones depending on the specified weighting. As the algorithm obtains better estimates of the effects of different conditions, it should eventually converge to placing all students in the most effective condition. This problem fits nicely into a multi-armed bandit formulation, where the arms are conditions and the reward is user learning, so we will attack it from this angle.

Our contributions are as follows. First, we formulate a dual-objective bandit in which we want to optimize a weighted combination of the 95% confidence interval sizes around the condition means, and user test performance. We then introduce UCB-Explore, a modification of an existing multi-armed bandit algorithm that tries to directly optimize for this user-specified tradeoff. Second, we analyze the performance of our algorithm in a 64-condition simulation with parameters learned from a real-world experiment involving different ways of displaying number lines. UCB-Explore better optimizes the weighted objective compared to existing bandit algorithms, and in our simulation appears fairly robust to changes in its parameters. Finally, we show how to use the samples generated from our algorithm to identify likely and unlikely two-way interactions between factors, and validate our hypotheses on a separate dataset.

2. RELATED WORK

2.1 Scientific Discovery, Massive Experiments

Researchers in AI have been working for years to develop systems capable of automatically generating useful scientific knowledge. This field, known as scientific discovery, has generated many such systems [14]. For example, Lee et al. used a feedback loop between the RL rule induction program and expert knowledge to identify potentially carcinogenic compounds [15]. Perhaps the most comprehensive example of such a system is Robot Scientist Adam, a fully automated robot capable of the full loop of hypothesis generation, experimental design, and data analysis in yeast genomics [10]. These systems are quite general, and try to automate many aspects of scientific behaviors; we are primarily concerned with the automation of certain kinds of experimentation in high-stakes domains.

We can also be considered related to massive (educational) testing, made possible in recent years through web-based software. As noted by Stamper et al., online experiments can be considered a new source of data with fundamentally different properties than ones conducted at the lab or school district level [24]. Experiments run online can have many more users, and better measure true task engagement; however, it may be difficult or impossible to collect rich data such as interview data. As examples, Lomas et al. conducted two large-scale experiments with many conditions on the effect of challenge on motivation and learning in an educational game [18]. Lindsey et al. used Gaussian process regression and an active learning framework to reduce the number of samples required to learn functions of user responses [16]. And in previous work, we proposed a greedy, hierarchical algorithm that ran sequences of multivariate tests in multi-factor settings [17]. In this work, however, we want an algorithm that adapts incoming samples to maximize an experimenter-specified weighted sum user learning and confidence in the estimated condition means.

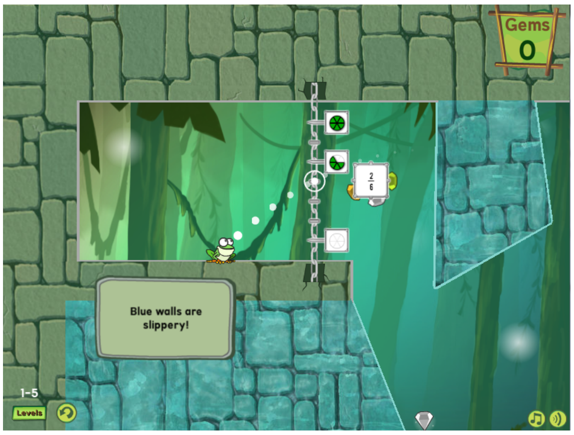
2.2 Adaptive Trials

Clinical trials are another example of high-stakes domains where it may be undesirable or unethical to assign patients to certain experimental conditions. Over the years, researchers have developed different methods for minimizing patient harm while trying to identify the best treatments. Some examples include play-the-winner, drop-the-losers, sample size re-estimation, adaptive treatment-switching, and so on; for a review, see [8] or [4]. The adaptive randomization designs are closest in spirit to our work: they bias the randomization in favor of successful conditions and away from failed ones [29] [26].

These strategies are often heuristic and offer no guarantees. However, clinical trials can also be formulated as multi-armed bandit problems, for which algorithms with theoretical performance guarantees are known. The closest work in this space is perhaps by Kuleshov et al., who propose the use of existing multi-armed bandit algorithms for the allocation of users to experimental conditions and show simulations suggesting that more patients are successfully treated [12]. Similar empirical investigations of bandits have been undertaken in the domain of web content retrieval by Vermorel et al. [25]. They focus on the standard bandit formulation in which the only objective is to maximize reward; in this paper, we also care about scientific knowledge.

3. MULTI-ARMED BANDITS

Throughout this paper, we will be using data drawn from an experiment carried out in the educational game Treefrog Treasure, seen in Figure 1. We give out different types of number lines to players and study how accurately they answer a randomized test number line; the full experimental design is described in Section 5.1. Importantly, we have two competing objectives: teach as much as possible to players in the game (by only giving out the best number lines), but identify the effectiveness of different fraction representations, hinting systems, or other number line properties (by testing different kinds of number lines). The relative weight of these two objectives will vary by application, so we will later allow the experimenter to set the weight explicitly.



Blue walls are slippery!

Figure 1: A screenshot of Treefrog Treasure, our source of users. Players navigate through a physics-based world, solving number line problems along the way. Notice that the number line has full tick marks, pie chart labels on the line, and a symbolic (ex. $\frac{a}{b}$) target representation. In our experiment, these are a few of the parameters we allow our system to automatically explore to determine which types of number lines lead to maximal near-transfer.

If our goal were solely to maximize player learning, then this problem reduces to a multi-armed bandit (MAB). MAB problems consist of K probability distributions, $D_1, \ldots, D_K$, with expected values $\mu_1, \ldots, \mu_K$. The D_i and μ_i are not known at the start. These distributions are classically viewed as wins or losses (1 or 0) from various arms of a slot machine, but in continuous formulations can be any (bounded) user-defined function. In our scenario, we can think of the arms as different types of numberlines to be given as practice, and the reward as player success on a randomized test number line.

In a bandit problem, the experimenter tries to pull arms in order to collect as much reward as possible (e.g. assign players to conditions to maximize test performance). At each turn, $t = 1, 2, ...$, we select an arm $j(t)$ and receives some reward from that arm's reward distribution $r(t) \sim D_{j(t)}$. If the D_i were known, the optimal strategy would be to choose the arm with the highest expected reward, $j(t) = \arg\max_i \mu_i$: that is, pick the most effective number line and give it to every player. Unfortunately, the D_i and μ_i are hidden. Successful bandit algorithms must navigate an *exploration-exploitation* tradeoff to discover information about the D_i while also generating high reward.

In general, we will not be able to give everyone the best intervention. Define the *total expected regret* at some fixed timestep T as the loss of reward from playing non-optimally,

$R^T = T\max_i \mu_i - \sum_{t=1}^{T} \mu_{j(t)}$. Lai and Robbins show that regret must grow at least logarithmically in time [13], developing a lower bound of $R^T = \Omega(log(T))$.

Not all strategies that work well in practice meet this bound. One such heuristic strategy is ϵ-greedy, which for any $0 < \epsilon < 1$ plays a random arm with ϵ probability and otherwise plays the arm with the highest empirical mean. This strategy has linear regret because it has a constant chance to play suboptimal arms. One could consider allowing ϵ to decrease over time to eliminate this linear regret term; however, this adds another parameter and does not always help in practice [25].

A different, popular class of theoretically-motivated strategies which meet the logarithmic regret bound are the Upper Confidence Bound strategies (UCB) [3]. These algorithms exemplify the principle of *optimism under uncertainty* by pulling the arm which has the highest estimated upper confidence bound on its mean. The simplest, UCB1, works in the following way. Assume that all rewards are in the range $[0, 1]$. To initialize, pull each arm once. Then at each subsequent timestep, if the number of times an arm i has been pulled is n_i, choose the arm $j(t) = \arg\max_i \hat{\mu_i}^t + c\sqrt{\frac{2\ln t}{n_i^t}}$. In this formula, the exploitative first term is the estimate of the arm mean, while the exploratory second term represents an uncertainty that grows slowly as other arms are pulled but decreases sharply when this arm is pulled. UCB1 provably incurs logarithmic regret when $c = 1.0$, though c is often set to be smaller for better empirical performance.

In this paper, we will focus on ϵ-greedy and UCB; for other algorithms, see [6]. Note,however, that all of these algorithms are focused on maximizing reward: bandit strategies will try to only allocate enough samples to sub-optimal arms to tell that they are indeed sub-optimal. Unfortunately, this may leave uncertainty about the exact values of each arm, which was our second goal. Furthermore, algorithms such as ϵ-greedy and UCB can be quite sensitive to the settings of their parameters.

On the opposite extreme, researchers have also studied the case where the only goal is to learn something about the arm distributions D_i, such as their means, the μ_i. Antos et al. introduce an algorithm for this problem, GAFS-MAX, which attempts to minimize squared error of the worst estimated μ_i by sampling the "most under-sampled" arm or the arm with greatest empirical variance [2].

Our definition of "scientific knowledge" is similar: we wish to minimize the sizes of our estimates of the 95% confidence intervals around the arm means. To the best of our knowledge, algorithms exist only for the extreme cases where we maximize only for reward or only for estimation of arm properties. In this paper, we propose allowing the experimenter to set a tradeoff between these goals, and introduce an algorithm which smoothly interpolates between these extremes by maximizing for the specified tradeoff.

4. UCB-EXPLORE

Our algorithm, called UCB-Explore, is a variant of UCB1. As noted earlier, changing the scaling factor c on UCB1's confidence bounds often leads to improved performance in practice: thus, the key idea behind UCB-Explore is to self-adjust c in response to mistakes. Our algorithm takes as input a set of arms with unknown reward distributions D_i, a function CI to calculate confidence interval sizes, a weight w controlling the tradeoff between reward and confidence interval sizes around the arm means, and a multiplier m that controls how quickly c changes. Good choices of CI in general depend on the shape of the reward distributions D_i, though methods such as the centered percentile bootstrap [23] allow reasonable estimates in most situations. In our example, the arms are Bernoulli processes generating "success" or "failure" depending on whether the student answers a test question correctly, so a reasonable choice for CI is the Wilson score interval [27].

Let N be the number of arms, r^t be the reward received on pull t, and Δ_j^t be the size of the 95% confidence interval of arm j on round t. For any timestep T, our goal is to maximize the expression $w\sum_{t=1}^{T} r^t - (1-w)\sum_{j=1}^{N} \Delta_j^t$. That is, we want to maximize the total reward received, but minimize the sizes of the 95% confidence intervals sizes, with some weight w between both goals. This type of goal makes sense if the experimenter is able to assign "worth" to confidence interval sizes, in terms of reward. For example, an educational institution might be given funding based on students' standardized test scores, and be willing to pay a certain amount of money for smaller confidence intervals about certain educational interventions. As an alternative interpretation, note that reward grows without bound while the confidence interval sizes cannot go below 0, so w can be thought of as a rough "switching point" after which the algorithm will aim primarily to gather more reward. Say that the experimenter knows a reasonable reward is 0.6, would roughly like the reward term to dominate after n pulls, and calculates that after this many pulls the confidence intervals can be expected to decrease by about 0.3 per pull. Then setting $w = 0.33$ causes the reward term to overtake the confidence intervals after about n pulls.

Note that this objective is difficult to optimize directly. This can be seen by considering the case $w = 1.0$, where we are focused on reward: it is computationally intractable to optimally pull arms to optimize the objective [19]. It must therefore be similarly intractable to optimize in the general case. As such, we propose UCB-Explore as a heuristic algorithm which lacks guarantees but seems to work well in our scenario. Our algorithm is shown in Algorithm 1. Let $c^1 = 1.0$ be our initial scaling factor, as in UCB1. We first pull each arm once; at subsequent times t, we choose the arm $j(t) = \arg\max_i \hat{\mu_i}^t + c^t\sqrt{\frac{2\ln t}{n_i^t}}$. When c^t is very large, $\hat{\mu_i}^t$ has little effect, and we will tend to choose arms that have fewer pulls (more exploratory). When c^t is very small, $\hat{\mu_i}^t$ dominates, so that we tend to pull only the arms with highest empirical means (more exploitative). So far, then, our algorithm is simply a tuned variant of UCB1.

The key change in our algorithm is that we increase or decrease c if we choose the wrong arm to pull. Say that arm i has been pulled at times $t_1, t_2, \ldots$. Then the rewards of all pulls of arm i up until this time are $R_i^t = r_i^{t_1}, r_i^{t_2}, \ldots$. Let $s_i^t = w\hat{\mu_i}^t + (1-w)(CI(R_i^t) - \mathbb{E}_{r_i^t}[CI(R_i^t + r_i^t)])$: that is,

Algorithm 1 UCB-Explore

```
Require: a tradeoff $w$, multiplier $m$, bandit arms $A_{1,\ldots,N}$
  $c = 1.0$
  for $j = 1$ to $N$ do
    $r_j = \textsc{Pull}(A_j)$
    $\mu_j = r_j$
    $n_j = 1$
    $R_j = \{r_j\}$
  end for
  for $t = N + 1$ to $\infty$ do
    for $k = 1$ to $N$ do
      $b_k = \mu_k + c\sqrt{\frac{2\ln t}{n_k}}$
      $c_j = \textsc{CalculateCI}(R_j)$
      $e_j = \textsc{CalculateExpectedNextCI}(R_j)$
      $s_k = w\mu_i + (1 - w)(c_j - e_j)$
    end for
    $u = \arg\max_i b_i$
    $v = \arg\max_{i \neq u} b_i$
    if $s_v > s_u$ then
      if $\mu_u > \mu_v$ then
        $c = mc$
      else
        $c = \frac{c}{m}$
      end if
    end if
    $r_j = \textsc{Pull}(A_u)$
    $\mu_j = n_j\mu_j + r_j$
    $n_j = n_j + 1$
    $R_j = R_j \cup r_j$
  end for
```

s_i^t is the weighted combination of the expected reward and the expected decrease in confidence interval size if we pull arm i. The calculation of $\mathbb{E}_{r_i^t}[CI(R_i^t + r_i^t)]$ in full generality requires a posterior estimate of D_i; since we are working with Bernoulli trials, we can estimate $p(r_i^t = 1) = \hat{\mu_i}^t$ and calculate the expectation directly.

When should we adjust c? In UCB-Explore, we ask whether we should have picked the arm with the second-highest upper confidence bound. Let $b_i^t = \hat{\mu_i}^t + c\sqrt{\frac{2\ln n}{n_i^t}}$ for each arm i. Without loss of generality, assume that $b_1^t >= b_2^t >= b_i^t$, $i = 3, \ldots, K$. If $s_2^t > s_1^t$, then our algorithm has made an error: it could have (greedily) obtained a better tradeoff respecting the researcher's decision of w by pulling the second best arm. If $\hat{\mu_1}^t > \hat{\mu_2}^t$, then the algorithm was exploiting too much, so we set $c^{t+1} = mc^t$. If $\hat{\mu_2}^t \leq \hat{\mu_1}^t$, then the algorithm was exploring too much, so we set $c^{t+1} = c^t/m$. We then pull the arm $j(t)$ and continue. It is important to note that this algorithm is heuristic in nature, but seems to work well in our simulation. It may be possible to develop a more theoretically-motivated algorithm to maximize for this weighted goal, which we leave to future work. In either case, if the algorithm respects w in its behavior and seems relatively robust to the choice of m, then we will have achieved our goal. We will see in Section 6 that this is the case in our scenario.

5. EXAMPLE APPLICATION

We will examine the performance of our algorithm with a 64-arm simulation whose parameters are drawn from real-world data. This will demonstrate the feasibility of our approach in a real-world situation. In this simulation, we will try to identify how the appearance of a "practice" number line affects player performance on a randomized "test" number line, a particularly challenging problem since we expect the effect sizes to be small given that the intervention is one number line long. We choose number lines as they are a well-studied and commonly-used pedagogical tool, and a fair amount of evidence suggests that much whole and rational number knowledge is organized around mental number lines [1], [22]. We will first describe the game from which we collected our data, as well as the factors that vary between number lines.

5.1 Treefrog Treasure

Treefrog Treasure is a platformer game that involves jumping through a jungle world and solving number line problems to reach an end goal. Number line problems serve as barriers that the player must solve by hitting the correct target location, as shown in Figure 1. It has been played by over 10 million players worldwide on various websites; data for this experiment is drawn from BrainPOP [5], an educational website aimed at school-aged children. Our dataset consists of 34,197 players, who played from June 3, 2013 to June 20, 2013.

We consider each player as a sequence of many pairs of number lines, and treat each pair as an experimental unit. This gives us 361,738 pairs. This potentially violates independence assumptions in classical statistical tests, but greatly increases the amount of available data we can use to estimate our arm reward distributions. We will strictly adhere to the correct assumptions when we attempt to generate hypotheses and validate them on a new dataset, later.

The appearance of the first number line in each pair constitutes the experimental condition - the full set of factors is specified in Table 1, with illustrations in Figure 2. We care primarily about Ticks, Animations, Backoff Hints, Target Representation, and Label Representation; the Fraction and Initial Labels are randomly chosen, so that our results are meant to generalize for different settings of these factors. This gives us 64 separate conditions, one for each combination of factor settings.

There are additional complexities in the sampling distributions in this dataset that are not relevant to this work; for a more thorough explanation, refer to [17]. The important point is that we can obtain an unbiased estimate, for each experimental condition, of the probability that a player receiving a number line with those parameters will reach and solve the randomized next "test" number line correctly on the first try. For our simulation we will assign each arm the associated mean estimated from our data, and draw simulated samples from the arms by flipping coins with the specified probability of success. These probabilities range from 0.38 to 0.47, with the vast majority falling in $[0.41, 0.45]$. Since the arms are Bernoulli in nature and the probabilities are close to 0.5, the variance is nearly the maximum possible for distributions in $[0, 1]$.

Parameter	Settings	Interpretation
Fraction	Any $\frac{a}{b} \in (0,1), b \leq 9$	The target fraction the player must hit
Initial Labels	[0,1]	For target $\frac{a}{b}$, the proportion of labels of $\frac{n}{b}$ fractions shown at the start.
Target Representation	Symbolic, Pie	How the target fraction is displayed.
Label Representation	Symbolic, Pie	How fraction labels on the number line are displayed.
Ticks	Present, Absent	For target $\frac{a}{b}$, we can display tick marks for each fraction $\frac{n}{b}$.
Animations	Present, Absent	If the player misses a target $\frac{a}{b}$, they might receive an lengthy pie chart animation showing how to divide up the number line into b parts.
Backoff Hints	1, 2, 3, 4	The number of misses for target $\frac{a}{b}$ before the progressive hinting system fills in all labels for $\frac{n}{b}$ and displays the correct answer.

Table 1: The parameters controlling number lines in our experiment. Bolded parameters are factors we are interested in studying; non-bolded parameters are selected randomly.

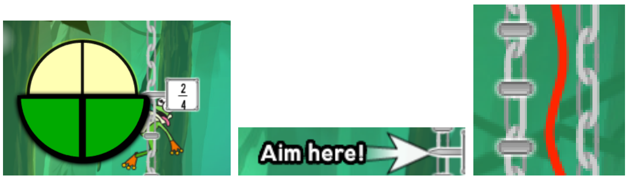


Figure 2: The animation condition on the left shows the player how to divide up the number line. The backoff condition in the middle gradually more direction about where to hit. The ticks condition either divides up the number line into segments when ticks are present, or leaves it empty besides the 0 and 1 labels when ticks are absent.

6. SIMULATION

Empirical MAB research such as [12] and [25] indicates that MAB algorithm performance is very sensitive to the exact values of parameters, and that tuned simple algorithms, such as ϵ-greedy, outperform theoretically-motivated algorithms such as UCB1. One reason that tuning affects reward is that different settings of these parameters can result in a tradeoff between identifying the best mean and exploiting the current best. Although these parameters do not explicitly optimize the tradeoff between confidence interval size and reward, it is often the case that more exploratory parameter settings will do a better job of minimizing confidence interval size. But it is not immediately obvious how to set these parameters for any particular tradeoff - what ϵ should we choose if we want to weight confidence interval size and reward equally? In contrast, UCB-Explore allows us to explicitly set this trade off and optimize for it more directly. We will compare how UCB-Explore trades off reward and scientific knowledge compared to MAB algorithms ϵ-greedy, UCB1, and UCB1-Tuned, ignoring the fact that the UCB-Explore parameter is given by the objective while the other parameters may be more difficult to choose.

ϵ-greedy is a simple and straightforward method for balancing the dual goal of learning about the arm means and also maximizing reward. It has the additional advantage of ϵ being easy to interpret: the proportion of players who will be devoted to exploring the non-optimal arms. UCB1 is also capable of trading off between learning and reward by scaling the bounds: making them very large causes the algorithm to prefer exploration, while making them small causes the algorithm to focus on the highest empirical mean and prefer exploitation. UCB1-Tuned replaces the loose bounds of UCB1 with ones that depend on empirical variance of the arms, which usually works better in practice [3].

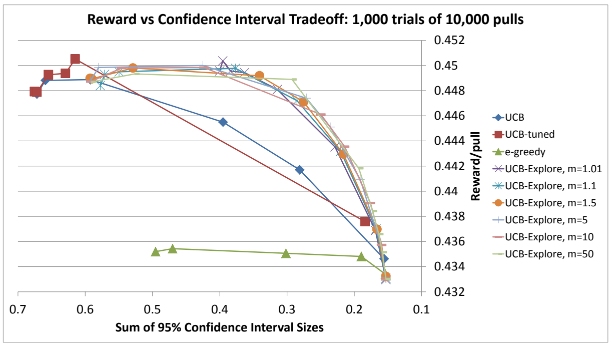


Figure 3: Reward vs confidence interval sizes. Up and to the right is ideal; we see that UCB-Explore is typically better at generating high reward and learning the various arm means than other algorithms. ϵ-greedy performs poorly overall.

Parameter	Values (right to left)
ϵ-greedy, ϵ	0.3, 0.03, 0.01, 0.001, 0.0001
UCB1, c	1.0, 0.2, 0.15, 0.1, 0.03, 0.01
UCB1-Tuned, c	1.0, 0.2, 0.15, 0.1, 0.03, 0.01
UCB1-Explore, w	0, 0.01, 0.03, 0.06, 0.1, 0.3, 0.6, 1

Table 2: The parameter settings generating the tradeoff graphs in Figure 3. All UCB1-Explore variants in the graph, which have different values of the scaling multiplier m, are generated from the same group of w.

We generate a tradeoff curve between average reward per pull and the sum of the sizes of the 95% confidence intervals around each estimated arm mean, shown in Figure 3. Each point for each algorithm is the average reward and interval size for 1000 trials of 10,000 pulls, for the parameters shown in Table 2. Points that are up and to the right are better. We see both that UCB-Explore tends to have superior performance, especially when one does not care entirely about reward, and also that ϵ-greedy appears much worse than both strategies once we scale the bounds calculated by UCB1. In addition, the different UCB-Explore curves are generated by different values of m - in our problem, it appears that the choice of m has little impact within a wide range, with perhaps the exception of $m = 1.01$.

To gain some intuition about how UCB-Explore behaves and how one should choose w, it is useful to examine the behavior of the scaling factor c that controls the size of the bounds. This is shown in Figure 4. As $w \rightarrow 0$, the algorithm

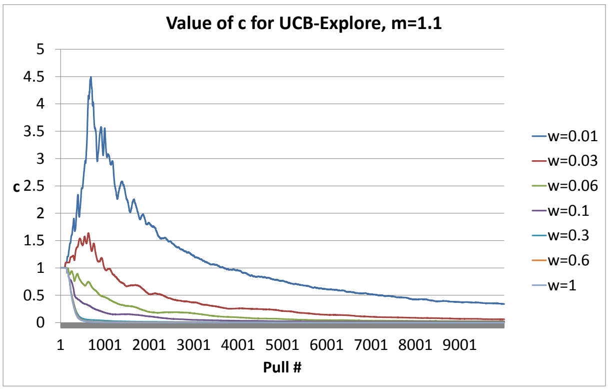


Figure 4: The value of the scale factor on the confidence bounds in UCB-Explore as the algorithm pulls more arms. When $w = 0$, c increases quickly and does not stop, reflecting the fact that the algorithm cares only about exploration.

cares less and less about generating reward, so we expect it to explore more: as expected, c increases. Furthermore, the gains from reward are constant over time, but the gains from reducing estimated confidence interval sizes shrink as we continue to sample. Thus as time passes we expect UCB-Explore to focus more and more on reward as long as $w > 0$, and indeed this is the behavior we observe by the shrinking values of c. In the case of $w = 1.0$, c actually shrinks so fast that the algorithm exploits too quickly, explaining the small dips in performance of both UCB-Explore and UCB1 when tuned to be very exploitative.

7. VALIDATION

7.1 Interaction testing

Our algorithm is a method of allocating samples that respects the tradeoff between encouraging player learning and getting more accurate estimates of experimental condition means. Had this experiment been run online, a researcher could directly analyze the data gathered. However, we have been running our algorithm in a simulated environment with parameters drawn from real-world data, leaving us vulnerable to overfitting. To demonstrate how one might use data gathered from our algorithm, we will instead use these samples to generate hypotheses to validate on a separate dataset: this is similar in principle to how a researcher might run multifactorial experiments in an online setting to find promising hypotheses to test in a more focused setting, as suggested by Stamper et al. [24].

UCBExplore can be thought of as a biased method of drawing samples from different experimental conditions. One natural analysis would be to ask if one condition is significantly better than another. Many of our factors are specific implementation choices in our game, so such comparisons may not lead to very generalizable insights. Instead, we will attempt to identify likely interactions using the samples generated by our proposed method. In particular, we will search for two-way interactions which seem relatively large or small; we study interactions instead of main effects because we already examined main effects in previous work [17]. Then we will use the likelihood ratio test to determine if models learned with these interaction terms fit our validation data significantly better than models where these terms are set to zero. We stress that samples generated from ϵ-greedy and UCB1 could also be used in the same way and would likely lead to the same results, though in light of our simulation results either more samples would be required or more damage would be done to players in that case.

	Target: Pie	Target: Symbolic
Label: Pie	0.431	0.410
Label: Symbolic	0.388	0.422

Table 3: Proportion of players in the validation set able to reach and answer the randomized test number line correctly on the first try. Our simulation results suggested that these parameters might interact; in fact, they interact very strongly.

To do this, we run UCB-Explore with m=1.1, w=0.001, and 100,000 pulls. We then calculate all main effects and two-way factor interactions as is done in the ANOVA test [28]. Our data is not normally distributed and the variances are unequal, violating ANOVA assumptions, but we can still consider which interaction effects seem relatively large or small. In our case, for each pair of factors, we can calculate the average magnitude of the interaction effects between all combinations of settings for those factors. We see that Target and Label have the largest average magnitude at 0.007, while Animation and Ticks have the smallest average magnitude at 0.0003. Thus, we suspect that Target and Label are much more likely to interact than Animation and Ticks.

To test these hypotheses, we will use a held-out validation dataset. This dataset consists of 9,675 players of Treefrog Treasure from June 20, 2013 to July 9, 2013. Unlike the dataset used to estimate parameters of our simulation in the previous section, we will consider only the first three number lines for each player: the first two are treated as the intervention, and the independently and randomly chosen third as the assessment of learning. For any given pair of factors, we can attempt to fit a model with only main effects, or a model with main effects and two-way interaction effects. Since these models are nested, we can use the likelihood ratio test if the interaction model is a significantly better fit, given the increase in degrees of freedom.

For Target and Label, we have that $\chi^2(1, N = 9675) = 7.555, p < 0.006$. For Animation and Ticks, we have that $\chi^2(3, N = 9675) = 0.204, p = 0.977$. Thus, it is very likely that the effects of Target and Label representation on numberlines should not be considered separately, while we have no evidence that our Animation and Ticks hinting systems need to be modeled simultaneously.

7.2 Target/Label representation

In this paper, our goal is to advocate the creation of algorithms which allow experimenters to trade off user learning and scientific knowledge, and the introduction of such an algorithm. The validation is meant to show that this approach generates samples that can lead to interesting hypotheses, so we do not claim them to be mature educational results. We will, however discuss them briefly.

The presence of significant interaction terms means that the factors involved should only be interpreted together. As a reminder, the Target factor refers to the representation of the fraction the player is asked to hit on the number line, while the Label factor refers to the representation of the fractions on the number line itself. The proportions of successful players for the different representation combinations can be seen in Table 3. The nature of the interaction is immediately apparent: players are more likely to reach and answer the next number line correctly if the target and label have the same representation, and Pie chart targets and Symbolic labels are much worse than other conditions. Even when we ignore players who quit before reaching the second number line, these effects persist.

We do not know why this is the case. In our game, player ability to answer number line questions correctly is mostly a function of knowing where to hit, as the game informs the player where they will intersect the number line before they click to jump. In addition, the representation has no effect on game mechanics. Thus, the difference is most likely due to how players perceive the different fraction representations. One possibility is that number lines in classrooms are generally presented with symbolic labels only, so that the mix of familiar and new combinations of representations is particularly confusing to players. As with nearly all online experiments, we do not have access to players' thought processes, only their actions, so a more carefully designed study or a think-aloud in a classroom might be profitable. Regardless, our algorithm was able to generate samples that we could analyze for interesting factor interactions, suggesting that it is a viable method of adaptively randomizing experimental conditions.

8. LIMITATIONS

While our results seem promising, there are some limitations. It is extremely unlikely that UCB-Explore is ideal for all bandit arm configurations; it seems to perform well on many-arm Bernoulli distributions with similar means, but some simulations suggest that it may not perform as favorably in very different cases. One example where our approach fails outright, as do ϵ-greedy and UCB1, is in the case that there are more arms than subjects - the algorithm expends all its subjects on the initialization phase when pulling each arm once. This problem occurs most obviously in the presence of continuous factors, in which case there are an infinite number of arms. It is less likely to occur in standard categorical experimental designs, though the limit can still be reached if researchers want to study something like the exponential space of all possible problem sequences.

Furthermore, while the reward portion of our dual objectives can be any measurable function of subject behavior, there may be other ways to define "scientific knowledge" or navigate the tradeoff between the two. For example, "scientific knowledge" might be the probability that the ordering of arms is correct. Or instead of assigning some weighting between reward and knowledge, an experimenter might have constraints of the form "maximize reward subject to at least x knowledge." Some of these are relatively easy to incorporate into our framework. The example constraint might be handled by forcing the scale factor to stay at 1.0 until enough information has been collected, for instance. Other types of constraints or tradeoffs may require entirely different algorithms: knowing the number of subjects in advance, for example, leads to very different bandit algorithms than the infinite horizon variant we have presented here.

9. FUTURE WORK

UCB-Explore appears to outperform UCB1, UCB1-Tuned, and ϵ-greedy in our problem for most tradeoffs between reward and knowledge. However, our modifications remove any theoretical performance guarantees. It may be possible to alter the algorithm in a principled way to maintain its good performance and guarantee logarithmic regret. Extensive simulation on other problems with more or fewer arms and different reward variances would also be useful to understand when it or another method of allocating samples is preferable. We would also like to test if this algorithm is robust to unusual continuous reward distributions.

In addition, there are other problem formulations that would be interesting to investigate. In many practical cases (such as delayed rewards), fully online learning is infeasible; for these, we could adapt Bayesian techniques such as probability matching [21], which do not depend on online performance. Also, in cases where there is a finite budget of users, the algorithm will need to be modified based on work in mortal bandits to exploit more as the experiment draws to a close [7]. Lastly, in reality the arm distributions may be nonstationary, which might require adaptation of work in dynamically changing bandits [9].

More generally, UCB-Explore only makes sense in situations where we have enough users to get substantial information about each condition. If we have many factors, we may not be able to get information about each specific condition, but may still be able to determine the best settings of the most important factors. A similar problem arises when we want to explore sequences of interventions, in which case techniques from Monte Carlo Tree Search may be most applicable. And if one of the factors is continuous, the algorithm will not be able to make progress: here, it could be useful to adapt work on bandits in general metric spaces [11], or modifying a function approximation scheme as in [16] to incorporate the reward-knowledge tradeoff.

10. CONCLUSION

The rise of online educational software with massive numbers of users promises to change the experimental paradigm in educational research. With access to so many users and individualized control over what educational experiences they receive, it is now possible to automatically run complicated, multi-factor experiments quickly and at relatively low cost. However, education is a high-stakes domain: in many situations we have the ability to cause harm by placing students in sub-optimal conditions. Because of this we want to automatically put less students into harmful conditions while simultaneously discovering which are harmful and which are beneficial.

In this paper, we propose allowing researchers to explicitly weight subject welfare against the amount of generalizable knowledge gained from the experiment. We show how the problem of allocating subjects to experimental conditions can be thought of as a multi-armed bandit problem with a dual objective of gaining maximum reward and minimizing the sizes of 95% confidence intervals around the arm means.

We propose a new algorithm, UCB-Explore, which takes a user-specified weight on the relative value of reward and confidence interval size, and adaptively adjusts its optimistic bound estimates to explore or exploit more when it makes a mistake with regards to this weight function. We analyze the behavior of our algorithm and compare it to tuned versions of other common bandit algorithms in a 64-arm simulation with parameters drawn from real-world data, showing that our algorithm is able to interpolate between these two goals much more effectively than standard algorithms. We use the simulated results of running our algorithm to generate some hypotheses about factor interactions, and confirm these results on a separate validation dataset, showing that the generated samples are useful from a research perspective.

11. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship grant No. DGE-0718124, as well as the Office of Naval Research grant N00014-12-C-0158, the Bill and Melinda Gates Foundation grant OPP1031488, the Hewlett Foundation grant 2012-8161, Adobe, and Microsoft.

12. REFERENCES

[1] D. Ansari. Effects of development and enculturation on number representation in the brain. *Nature Reviews Neuroscience*, 9(4):278–291, 2008.

[2] A. Antos, V. Grover, and C. SzepesvÃąri. Active learning in multi-armed bandits. In Y. Freund, L. GyÃűrfi, G. TurÃąn, and T. Zeugmann, editors, *Algorithmic Learning Theory*, volume 5254 of *Lecture Notes in Computer Science*, pages 287–302. Springer Berlin Heidelberg, 2008.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[4] D. A. Berry. Adaptive clinical trials in oncology. *Nature reviews Clinical oncology*, 9(4):199–207, 2012.

[5] BrainPOP. http://www.brainpop.com/.

[6] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

[7] D. Chakrabarti, R. Kumar, F. Radlinski, and E. Upfal. Mortal multi-armed bandits. In *NIPS*, pages 273–280, 2008.

[8] S.-C. Chow, M. Chang, et al. Adaptive design methods in clinical trials-a review. *Orphanet J Rare Dis*, 3(11), 2008.

[9] A. Garivier and E. Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008.

[10] R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova, A. Sparkes, K. E. Whelan, and A. Clare. The automation of science. *Science*, 324(5923):85–89, 2009.

[11] R. Kleinberg, A. Slivkins, and E. Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, STOC '08, pages 681–690, New York, NY, USA, 2008. ACM.

[12] V. Kuleshov and D. Precup. Algorithms for the multi-armed bandit problem. *Journal of Machine Learning*, 2000.

[13] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[14] P. Langley. *Scientific discovery: Computational explorations of the creative processes.* MIT press, 1987.

[15] Y. Lee, B. Buchanan, and J. Aronis. Knowledge-based learning in exploratory science: Learning rules to predict rodent carcinogenicity. *Machine Learning*, 30(2-3):217–240, 1998.

[16] R. Lindsey, M. Mozer, W. J. Huggins, and H. Pashler. Optimizing instructional policies. In *Advances in Neural Information Processing Systems*, pages 2778–2786, 2013.

[17] Y.-E. Liu, T. Mandel, E. Brunskill, and Z. Popović. Towards automatic experimentation of educational knowledge. In *CHI*, 2014.

[18] D. Lomas, K. Patel, J. L. Forlizzi, and K. R. Koedinger. Optimizing challenge in an educational game using large-scale design experiments. In *CHI*, 2013.

[19] W. B. Powell and I. O. Ryzhov. *Optimal learning*, volume 841. John Wiley & Sons, 2012.

[20] L. Razzaq, M. Feng, G. Nuzzo-Jones, N. T. Heffernan, K. Koedinger, B. Junker, S. Ritter, A. Knight, E. Mercado, T. E. Turner, R. Upalekar, J. A. Walonoski, M. A. Macasek, C. Aniszczyk, S. Choksey, T. Livak, and K. Rasmussen. Blending assessment and instructional assisting. In *Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology*, pages 555–562, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press.

[21] S. L. Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

[22] R. S. Siegler, C. A. Thompson, and M. Schneider. An integrated theory of whole number and fractions development. *Cognitive Psychology*, 62(4), 2011.

[23] K. Singh and M. Xie. Bootstrap: A statistical method, 2008.

[24] J. C. Stamper, D. Lomas, D. Ching, S. Ritter, K. R. Koedinger, and J. Steinhart. The rise of the super experiment. In *EDM*, pages 196–200, 2012.

[25] J. Vermorel and M. Mohri. Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005*, pages 437–448. Springer, 2005.

[26] L. J. Wei and S. Durham. The randomized play-the-winner rule in medical trials. *Journal of the American Statistical Association*, 73(364):pp. 840–843, 1978.

[27] E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

[28] B. J. Winer. *Statistical principles in experimental design.* McGraw-Hill Book Company, 1962.

[29] M. Zelen. Play the winner rule and the controlled clinical trial. *Journal of the American Statistical Association*, 64(325):pp. 131–146, 1969.

Vertical and Stationary Scales for Progress Maps

Russell G. Almond
Florida State University
Educational Psychology and
Learning Systems
Tallahassee, FL
ralmond@fsu.edu

Ilya Goldin
Center for Digital Data,
Analytics, and Adaptive
Learning
Pearson Education
ilya.goldin@pearson.com

Yuhua Guo & Nan Wang
Florida State University
Educational Psychology and
Learning Systems
Tallahassee, FL
[yg07c,nw13]@my.fsu.edu

ABSTRACT

Students and instructors would benefit from a graphical display of student proficiency throughout a course. However, valid and reliable proficiency estimates based on modern statistical techniques require data that are not usually collected in traditional instruction. For example, problems that students solve on tests and homeworks may not be properly equated to a vertical scale; building a true vertical scale requires that overlapping anchor items be administered in way that supports the estimation of student growth between assignments. This paper suggests an alternative, a *stationary* scale in which the expected student growth is subtracted out so that a student making normal progress remains at the zero point in the scale. We define the stationary scale model and validate it on a real-world data set of student answers to homework items. We further produce a Progress Map, a visualization of student proficiency throughout a course.

Keywords

Ability estimation, Homework, Item Response Theory, Kalman Filter, Smoothing, Partially Observed Markov Decision Processes, Progress Maps, Graphical Displays, Vertical Scales

1. INTRODUCTION

All students in a course usually have the question,[1] "Am I on track to master the objectives listed in the course syllabus?" Good students will revisit this question throughout the course and adjust their studying strategy if they are at risk of not mastering the objectives. Instructors have two related questions: "Are my students (as a class) on track to master the objectives?" Again, good instructors will monitor the answers to these questions and "Which students are at risk to not master the objectives?" and adjust the instruction as needed. This is an issue of measurement.

Optimal measurement is not the only consideration when instructors choose items for assignments and quizzes. Instructors primarily choose items to practice objectives recently introduced in class. Although using multiple items per instructional objective produces more reliable measurement, the instructor must balance the test length with student fatigue. If students do not all work on the same items, as may be the case if items are adaptively selected or automatically generated by an Interactive Learning Environment (ILE), then despite item differences, instructors rarely attempt to put the scores onto a common scale. In particular, ensuring that there are sufficient overlapping items between forms to do any kind of common item equating is usually a low priority when choosing items for an assignment.

The lack of a proper equating design in the assignments complicates estimating student *growth* over time from homework assignments. Most procedures for estimating growth require all of the assignments to be linked to a common *vertical scale* [13]. One method of constructing a vertical scale requires overlapping items between adjacent assignments. This is a problem for the instructor because the overlapping items will either cover problems from prior or future topics. As the instructor's goal is maximizing the time spent practicing the current topics, such review and preview items are seldom included on assignments. A second method of constructing a vertical scale requires administering items from throughout the course at a common time point to a group of students who have been subject to a standard set of instruction. Usually courses offer limited opportunities (e.g., pretest, midterm, final) to do such calibration.

To address the problem of tracking student progress across time using homework assignments that have not been vertically scaled, this paper introduces an alternative to the vertical scale called a *stationary scale*. The basic assumption of the stationary scale is that the average student ability and the assignment difficulty increases at the same rate; in other words, the instructor designs each assignment to match the expected ability distribution of the students at the time when the assignment is due. On the stationary scale, the expected ability of a student who is on track remains at 0 throughout the course. This is equivalent to a stationary time series. We use the stationary scale to construct a unified model for multiple assignments over time, incorporating at once all the observations about each student over the duration of a course. This has two benefits: First, it lowers the error of measurement for each assignment

[1] The actual question is something more like "Will I get a good grade?" Good grades, however, should follow from mastering the objectives.

and each student. Second, it puts the assignments on a common scale so that growth can be interpreted as deviations from expected growth.

The next section lays out a time series framework for assignments and talks about previously developed models for growth and observed outcomes on assignments. The following section describes the stationary model and the calibration of models to the stationary scale. The fourth section explores the application of the stationary model to a database of online homework results. The last two sections explore some possible graphical displays and offer suggestions for improvement and future work.

2. MODELS FOR STUDENT CHANGE OVER TIME

Figure 1 shows a general Markov decision process framework for integrating information from multiple assignments over time [2]. Here S_t represents the latent ability of the student, and O_t represents the observed outcomes of the assignment offered at Time t (both of these quantities could be multidimensional). The nodes marked *Activity* represents what activity the instructor chooses for the students between sessions. If the problem is to choose an optimal strategy for selecting activities, then Figure 1 is a partially observed Markov decision process (POMDP)[3]. To simplify the problem, assume that all students get the same action, "continue with the next lesson according to the syllabus," at each time point. This reduces the problem from a POMDP to a hidden Markov model (HMM).

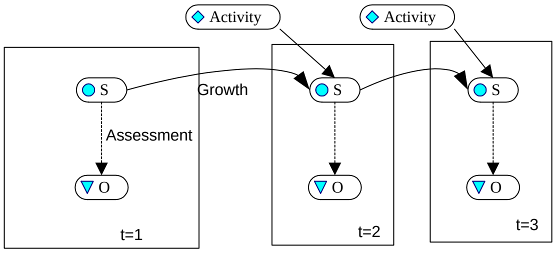


Figure 1: Generic Framework for Accumulating Assessments over Time [2].

The POMDP/HMM framework decomposes the modeling problem into two pieces: what happens within a single vertical time slice (i.e., within an assignment) and what happens between time slices (i.e., the growth model). In a single time slice, familiar models such as item response theory (IRT) apply. For the growth model, the Brownian motion process provides a simple starting point. The biggest drawback of this framework is that the scale of the latent variable, S_t is not identified; this causes difficulty in estimating model parameters from data [1]. The common solution is to put the latent variables onto a vertical scale.

2.1 IRT Models for observed outcomes

For simplicity, assume that the latent state of Student i at Time t, S_{it}, can be represented with a unidimensional, continuous random variable. If every student is in the same course with the same syllabus, the distance along the common path through the multidimensional space defined by the course syllabus that the student has traveled [14] can appear like a single dimension. Care must be taken when comparing the stationary estimates from different courses as different paths through the multidimensional space will give different meaning to the same value on the unidimensional scale.

Again for simplicity, assume that observable outcome from Student i interacting with the assignment given at Time t is a binary vector $\mathbf{O}_{it} = \{O_{i1t}, ..., O_{iJt}$, where $O_{ijt} = 1$ if Student i got a correct answer to the jth item given at Time t, and zero otherwise. Item response theory (IRT) models the likelihood of each observation as conditionally independent given the latent state of the student [11]. There are several possible models; this paper uses a one-parameter logistic (1PL) or Rasch model:

$$P(O_{ijt} = 1|S_{it}) = \text{logit}^{-1}(S_{it} - b_{jt}) \qquad (1)$$

b_{jt} is an item-specific difficulty parameter.

In an IRT model, the scale and location of the latent variable S_{it} is not identifiable from the data. One conventional way to resolve this, as we do here, is to set the average of the difficulties, b_{jt}, to 0. If we calibrate the IRT model using the data from a specific class taking a specific assignment, then this will produce a set of *class-specific IRT parameters*.

IRT has mostly been applied in the world of high-stakes testing, where each examinee attempts a problem exactly once. However, in online learning, most of the assignments will be homework and other lower-stakes assessments. Online homework systems frequently allow multiple attempts at an item, and access learning aids (e.g., tutorials and worked solutions to similar problems) when difficulty arises. Furthermore, the course policy may allow multiple attempts at the whole assignment, even for tests and quizzes which only allow a single attempt at each item within the assignment.

In the world of online homework systems that allow multiple attempts it becomes difficult to define "correct". Two possible definitions are *Correct-on-first-try*—solving the item on the first item-level attempt in the first assignment-level attempt without the use of learning aids,—and *Eventually-correct*—solving the item on any attempt with or without learning aids. The two different definitions of correct will give slightly different meanings of proficiency. These correspond to Falmagne's *inner* and *outer fringes* of the learning space [8]): correct-on-first-try corresponds to the inner fringe—those things that the student can do without assistance,—and eventually-correct corresponds to the outer fringe—those things the student can do with assistance.

2.2 Brownian Motion Growth Model

As the goal of this paper is to produce quick estimates of proficiencies that can be used to track student progress, the simplest growth model discussed in [2] provides a good starting point. Assume that between Times $t-1$ and t, the average expected growth following the syllabus is δ_t, and let Δt be the time (either in calendar time, or some measure of progress through the course as number of chapters of the textbook covered) between the assignments at Times $t-1$ and t. Further assume that the growth for an individual student over one time period is normally distributed around the class average, let ω^2 be the variance over a unit time interval. The variance of the growth between two assignments is $\omega^2\Delta t$, that is, the variance of the growth is proportional to the elapsed time. Thus, $S_{it} \sim N(S_{i,t-1} + \delta_t, \omega^2\Delta t)$. This is a non-stationary Brownian motion process.

The Brownian motion model implies that the less frequently the student is assessed, the more uncertainty there is about the student's ability. Almond [2] suggests that when the variance of the growth increments, $\omega^2\Delta t$, is small with respect to the standard error of measurement, τ_t, the ability estimates can be smoothed across time to have a lower mean squared error. Almond suggests a number of techniques for smoothing: a simple exponential filter (down weighting each prior observation by a factor λ), the Kalman filter [10] (this assumes that O_{it} is approximately normally distributed given the latent ability) and the particle filter [7] (which supports many models for both within and between time slices). These models can also forecast future ability states.

Attempting to estimate the within-time slice (observable outcome) and between time-slices (growth) models at the same time can cause difficulty [1]. In particular, either the average ability increase δ_t or the average difficulty of the items at Time t cannot be identified from the data. The usual approach in these circumstances is to put the assessments given on each time slice onto a vertical scale.

2.3 Vertical Scales

Educators frequently want to measure student learning using tests administered at different times and covering different but overlapping content. In order for differences in the test scores to be meaningful, the tests must be placed onto a common or *vertical* scale [13]. Vertical scales can be challenging to develop [15] and require difficult to verify assumptions [9].

In particular, constructing a vertical scale usually requires *anchor items*, items that are placed in two adjacent tests in the series. By assuming the anchor items have the same psychometric properties in both administrations (an assumption which is open to question [12]) the adjacent test forms can be equated. While anchor items are included in high-stakes testing programs, they are seldom included in homework assignments, where the focus is maximizing practice of the most recently introduced material.

An alternative is to place the anchor items into a separate test which is administered at a single time, so the students see the items covering many parts of the curriculum at the same time. Pretests and final exams provide natural experiments of this type. Even so, the number and quality of the anchor items controls the quality of vertical scale (in particular, the standard error of the equating that underlies its development). It is unusual to have enough anchor items to properly build a vertical scale for homework data.

Assume that a number of anchor items have been assigned difficulty parameters on the vertical scale. Let J_t be the subset of items in the assignment given at Time t that have associated difficulty parameters on the vertical scale, b^*_j. To equate the current assignment to the scale defined by the vertical scale [11], replace the difficulties in the assignment, b_{jt} with the equated difficulties:

$$b''_{jt} = b_{jt} - c_t \ , \tag{2}$$

where

$$c_t = \sum_{j \in J_t} b_{jt} - b^*_j \ .$$

Note that the quality (i.e., standard error) of this equating will depend on the number of anchor times in each assignment. As homework assignments are typically short, the number of available anchor items is typically small and hence the quality of the vertical scale is questionable.

3. STATIONARY SCALES

Consider the problem of estimating students' abilities as they progress through a course. Assume that homework, quizzes and tests are given online, so that the course learning management system has a record of each item from each assignment, as well as which items were and were not attempted. Furthermore, assume that the class size is large enough that parameter estimates from calibrating the IRT model given in Equation 1 will have reasonable standard errors. The instructor would like proficiency estimates for each student at the time of each assignment, as well as end-of-course forecasts.

If the item parameters are not already available, the difficulty of each item must be estimated from the course data. However, the instructors usually assign items when they make sense according to the syllabus of the course, i.e., shortly after the objectives covered in the item were covered in class. This pattern of item assignment does not usually produce the kind of overlap needed to support the construction of a vertical scale.

The alternative we are proposing is a *stationary scale*. Let $\overline{b_t}$ be the average difficulty of the items given at Time t on the vertical scale. Then the stationary scale is defined by assuming $\delta_t = \overline{b_t} - \overline{b_{t-1}}$. In other words, the difficulty of the assignments grows at the same rate as the ability of the students. To put the item parameters and ability estimates on the stationary scale, set:

$$S^0_{it} = S_{it} - \sum_{s=1}^{t} \delta_t \ , \tag{3}$$

$$b^0_{ijt} = b_{ijt} - \sum_{s=1}^{t} \delta_t \ . \tag{4}$$

On this scale, the ability of a person moving through the course at the pace determined by the syllabus will be a stationary (zero mean) time series. In particular, the growth model of the previous section will be a *stationary* Brownian motion process, $S_{it} \sim N(S_{i,t-1}, \omega^2\Delta t)$. Stationary time series are simpler to work with than non-stationary time series. In fact, many books on time series (e.g., [4]) recommend differencing the time series to make it stationary before analyzing the data. Similarly, many filtering techniques which could be used to smooth the observed ability estimates assume stationary time series.

The key assumption of the stationary scale is that the instructor, in the process of constructing the assignments, has taken care of the vertical scaling problem. In particular, we assume that the instructor is picking items so that the expected percent correct, hence the average difficulty, is approximately the same on each assignment. (The first author has found that after 2 or 3 times teaching an elementary statistics course, the median score on the midterm exam is usually close to the target score of 85%.)

Unfortunately, this assumption is difficult to test in practice. A convincing test would require a data collection design similar to the one required for a true vertical scale. The following section explores some more superficial checks which can be done using an arbitrary set of homework data from a large class. The remainder of this section looks at two operations which are now possible under the stationarity assumption: comparing a class to a database of similar classes, and smoothing ability estimates over time.

3.1 Classroom Level Estimates: Equating a class to a database

Assume that the IRT parameters have be calibrated using data from a single class, and that the scale has been identified by setting the average difficulty of each assignment to zero. Care must be taken in the interpretation, because a sudden drop in the average class ability could mean that the assignment was poorly designed rather than the class is not meeting expectations.

For many instructors, it would be useful to compare the performance of their class to similar classes: perhaps the same class offered in different years, or similar classes offered by other instructors. In particular, if we have a database of homework results from different classes using the same text, and if we assume that all of the instructors who use this database are also assigning items according to the stationary scale, then we can equate each class to the scale defined by the database using Equation 2.

Again, a version of the stationarity assumption allows a meaningful interpretation of item difficulties averaged across different courses. If we assume that each instructor introduces the item when it is instructionally relevant, that is when its difficulty matches the average student ability in the class, then the average difficulties across all classes in the database will also be on a stationary scale of sorts. This is a stronger version of the stationarity assumption than the single class version. If historical homework results are drawn from one textbook and pool of items across any instructors and syllabi, this will result in differences in which book sections are covered, their relative emphasis, and timing of delivery. Applying the stationarity assumption to the whole database assumes that the variability in the item difficulties produced by the variations in context are ignorable.

3.2 Smoothing the Ability Estimates

The real advantage of the stationary scale is that we can use the model of Figure 1 to smooth the proficiency estimates. Thus, at any time point the best estimate for a student ability is a weighted average of the student's ability estimate at the previous time point and the estimate from the current assignment. The weights depend on the relative size of the standard error of measurement for the assignment and the variance between time steps in the growth process [2]. Assuming that the growth model is normal process and that the ability estimates are normally distributed around the true ability allows the Kalman filter to be used to smooth the proficiency estimates. Although the observations are binary, the shape of the likelihood for the ability in the IRT model is approximately normal with a mean corresponding to the point estimate and a standard deviation corresponding to the standard error of measurement.

Implementing the Kalman filter requires knowledge of the variance of the growth increments, ω^2. If the Brownian model model holds, then the variance of the ability variable should increase linearly with time. We estimated a Rasch model using a regression with a random student effect [6] for each assignment. This produces a variance for the student ability variable at each time point. The slope of the line for the ability variances regressed against time provides an estimate of ω^2. This provides all of the necessary information to smooth the ability estimates using the Kalman filter.

Smoothing across time points should reduce the standard error of the ability estimates. In particular, the ability estimate at each time point will have a standard error that depends on both the direct evidence from the current assignment and the indirect evidence from the past history and the typical trajectory of student abilities. This will given a pattern of constant ability (on the stationary scale) and shrinking standard errors for students who are progressing normally. It helps answer one question instructors often have: when is a low assignment score a one-off fluke and when is it an indication of a problem which requires attention. In the former case, the filter will smooth the estimate towards the student's usual performance; in the latter case, the instructor will see the student trend line drifting away from the average trend of the class.

The biggest problems for instructors are not the students who progress normally, but the students who do not, especially students who do not complete assignments. Following the Brownian motion growth model, student ability will have an expected increase of δ_t on the vertical scale for each assignment, or on the stationary scale, expected ability will stay the same. The standard error of that estimate should increase. In particular, the variance of the estimate will be $\omega^2(t-t_0)+\sigma^2_{t_0}$, where $\sigma^2_{t_0}$ is the standard error of the ability estimate at the last time t_0 for which work is available for the student. If the student later returns to a normal completion pattern, the standard error will once again decrease and the proficiency estimate will track the student's ability. If the student continues to not submit assignments, the uncertainty will grow steadily larger.

4. MODEL VALIDATION

In practice, the stationary assumption is difficult to test: a rigorous test requires overlapping items in same kind of pattern as is used to construct a vertical scale. There are, however, three consequences of the model that we can test. First, if we separate the data into two pieces the mean ability on the stationary scale should change at the same rate for both groups. Second, the smoothed estimates of ability should have lower standard errors than estimates using data only from the most recent assignment. Third, the filter should produce reasonable predictions for the current assignment based on past assignments.

We fit the model to a data set of 578 students, spread across 9 sections, enrolled in an Intermediate Algebra course in the same semester. Students completed homework assignments online using the MathXL system[2]. There were 18 assignments taken from 3 chapters, Chapters, 3, 6 and 9 (with time elapsed between chapters). The assignments ranged in length from 12 to 62 scorable item parts[3], with most assignments having around 20. The number of students active in the course declined over time ranging from 567 attempting the first assignment to only 408 student attempting the penultimate assignment.

We randomly chose 10% of the students as test data and used the remaining students for training data. Using only the training data, we fit a Rasch model to all of the items in each assignment using a logistic regression with a random effect for student (ability estimate) and a fixed effect for items (difficulty) [6]. We then put the item parameters for that assignment onto the stationary scale by subtracting the average difficulty for each assignment. Once we had the final item parameters, we constructed expected a posteriori (EAP) estimates and the corresponding standard errors for the ability of each student in both the training and the test samples. Because the EAP estimates would later be combined with prior information about the student ability in a filter, a flat prior was used for the EAP estimates.

Figure 2 shows the average EAPs for the training and test samples. Note that the two estimates track each other closely. The correlation between them is $r = 0.92(n = 18)$. While this does not prove that the stationary assumption holds, it does demonstrate that if it holds for the training sample, it holds for the test sample as well.

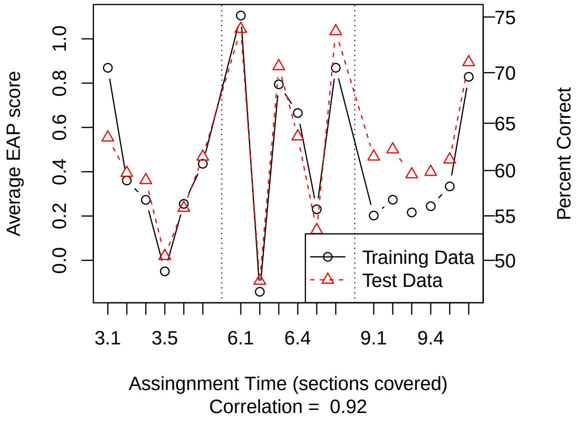


Figure 2: Average EAP ability estimates over time

If the Brownian motion model holds, the variance of ability estimates (from the random effects logistic regression) against time (sections of the book completed) should rise with a slope of ω^2. Figure 3 shows the observed variances. For the first two chapters, where the variance is decreasing, the participation rate dropped by about 100 students. During the third chapter (Chapter 9) the sample size was more stable, and the population variance shows the expected linear increase. Consequently, we used the slope of the increase during the final chapter as the estimate of ω^2.

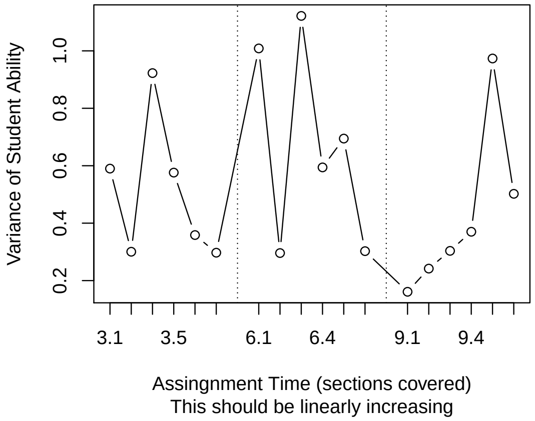


Figure 3: Population ability variance over time

Next, the ability estimates from the IRT calibration were smoothed using the Kalman filter. Because the smoothed estimates take both current and historical evidence into account, the standard errors for the smoothed estimates should be lower than that the standard errors of the original IRT estimates (which only include the current time point). Figure 4 verifies this is the case, plotting the average[4] standard error. The standard errors for the filtered estimates get better over time as the filter incorporates more data. Also, the standard errors increase during the intervals between chapters when some time elapses without measurements of student progress. It is also possible to run the filter backwards to get improved estimates for earlier time points (incorporating later data), but that was not done.

Validating the quality of the forecasts is difficult because there is no baseline to compare it against. As a weak form of validation, we look at the size of the average difference between the forecast from the filter and the EAP estimate from the IRT data (with no smoothing). Let $\hat{\theta}_{n,r}$ be the EAP estimate using only data from the current assignment at the time of Assignment r, and let $se(\hat{\theta}_{n,r})$ be its standard error. Further, let $\theta^*_{n,r}$ be the one step ahead forecast from the filter incorporating only data from past assignments. Let $z_{n,r} = (\theta^*_{n,r} - \hat{\theta}_{n,r})/se(\hat{\theta}_{n,r})$ be the standardized difference between the forecast and the current data only estimate.

Figure 5 shows the root mean squared standardized difference between the filter forecast and the IRT estimate using

[2] http://mathxl.com

[3] Some items had multiple scorable parts. We treated those parts as separate items, ignoring the dependence between the parts of the items.

[4] The average used here is the root mean square standard error, that is the square root of the mean prediction variance. Data points with missing assignment data are excluded.

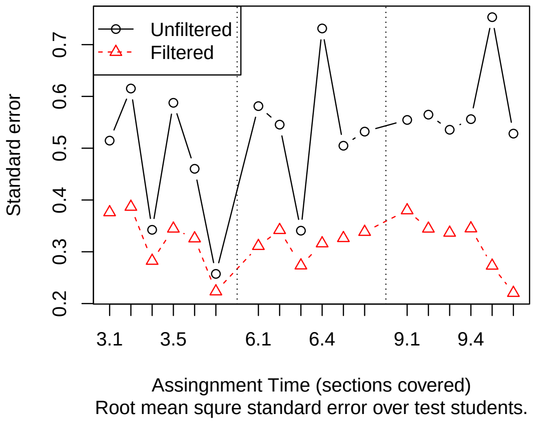


Figure 4: Average standard error for test set

only the current assignment data. The average (over the test set) difference is less than one standard error (of the IRT estimate) for all the time points, indicating the filter is doing reasonably well. Note, however, that the filter is doing fairly well even at time point zero where it is simply predicting the class mean for every student. Therefore, the positive result speaks more to the low information from the relatively short homework assignments than it does to the quality of the forecasts from the filter.

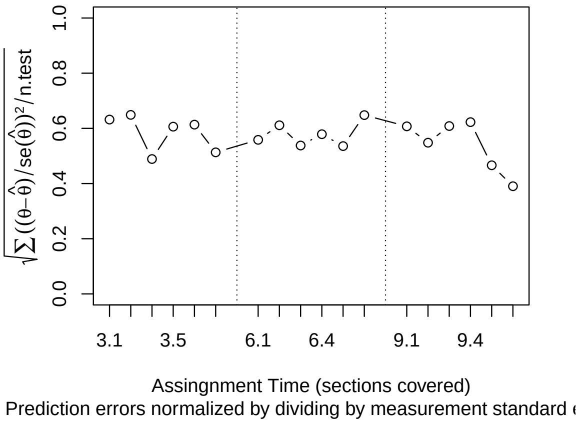


Figure 5: Root Mean Squared prediction difference

5. PROGRESS MAPS

To illustrate how the stationary scale can be displayed to an instructor or student, we display the progress of an arbitrarily chosen student. One simple graph plot the ability estimate from the IRT model against the time. We connect the measurements with a line; a dotted line indicates that one or more intermediate assignments is missing. There are a number of possible time scales to use. The method we found provided the clearest displays was to simply use the sequence number for each section, adding an extra step between the chapters. Figure 6 shows the result.

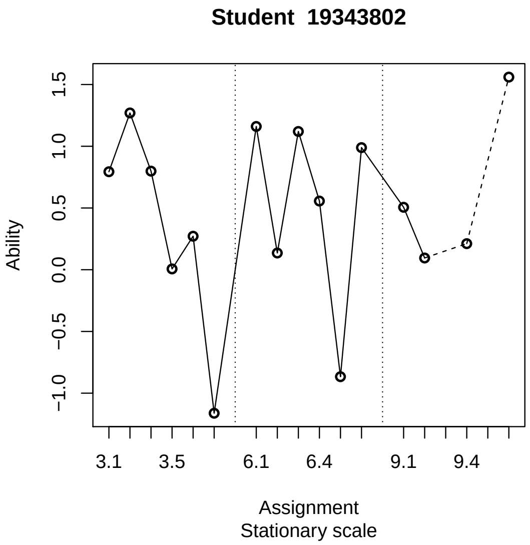


Figure 6: Stationary Map: Graph of student progress on a stationary scale

There are two additions we would like to make to this progress map. First, we would like the scale of the graph to give the visual impression that students making normal progress are increasing in proficiency. Second, we would like visual indications of the standard errors and expected performance standards.

5.1 The Progress Scale

Although the stationary scale is mathematically convenient, the ability estimates for students making normal progress will remain flat. Students and instructors would prefer to see progress towards a goal, i.e., rising ability. If the ability increases, δ_t, were known at each time point, then the data points could be put back on the vertical scale by inverting Equation 3 or 4. However, learning the ability increases is equivalent to the problem of learning the vertical scale. In particular, it requires the existence of a set of anchor items assigned in a pattern that supports the establishment of a vertical scale.

An alternative is to simply pick a convenient value, d_t, and set $\delta_t = d_t$. We call this scale the *progress scale*. There are now three possible scales (related through Equations 3 and 4):

Vertical Scale The values of δ_t are estimated from data. The quality of the scale will depend on the available anchor items.

Stationary Scale This defines $\delta_t \equiv 0$. Item parameters can be put on this scale by calibrating each assignment separately.

Progress Scale This defines $\delta_t \equiv d_t$ as an arbitrary series of constants. It can be readily produced from the stationary scale to make an increasing scale for display.

In our preliminary experience with graphical displays we have found letting $d_t = 1/K_t$, where K_t is the number of sections in the chapter administered at time t works well. This corresponds to an increase in one standard deviation in the population ability for each chapter covered in the text. Figure 7 shows an example. Here d_t is set so that progress through 1 chapter is the equivalent of 1 point on the scale; the d_t for a section is the corresponding fraction of the chapter. Note that the progress scale is slightly different scale from the section count scale used for the x-axis; that is why there appear to be sharp rises between the end of each chapter and the beginning of the next.

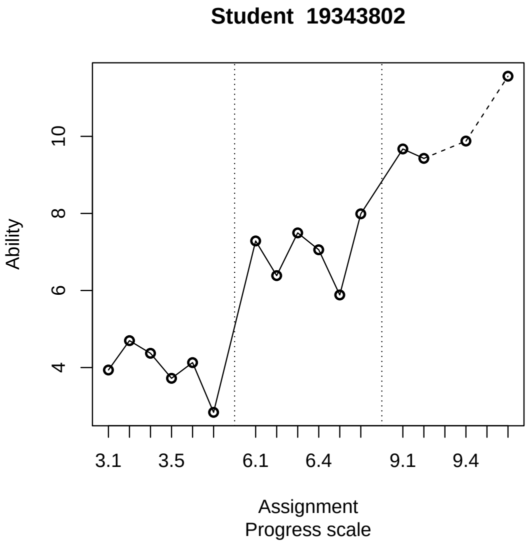


Figure 7: Progress Map: Graph of student progress on a progress scale

5.2 Error Bars and Control Limits

One drawback of the progress maps in Figure 6 and 7 is that they provide no information about the precision of the ability estimates. Figure 8 adds error bars to each point estimate extending ±2 standard errors from the point estimates. Both the point estimates and the standard errors are the unfiltered estimates from the IRT analysis.

Figure 8 also adds control limits to the display. The progressively darker shaded regions on the graph indicate areas of increasing concern for the student and instructor. When the point estimates pass the control limits, the size of the plotting symbol is changed to make the problematic data points more visible. The control limits are wavy instead of smooth because two different time scales are used, one (number of chapters completed) for adjusting the ability and one (assignment count) for plotting the time.

There are several ways of coming up with the control limits. Figure 8 uses a simple idea based on the IRT model. The instructor chooses a proportion correct for the assignment. To get the control limits, solve the IRT equation (Equation 1) for the ability that leads to that proportion for a zero difficulty item. This provides a roughly interpretable limit. The probabilities used in Figure 8 are .4, .2, .1, and .05.

The big advantage of the stationarity assumption is that it allows smoothing ability estimates using the Kalman filter. Figure 9 shows the filtered time series for the first student. Note that the error bars in Figure 9 are smaller than the error bars in Figure 8. This is an effect of the smoothing.

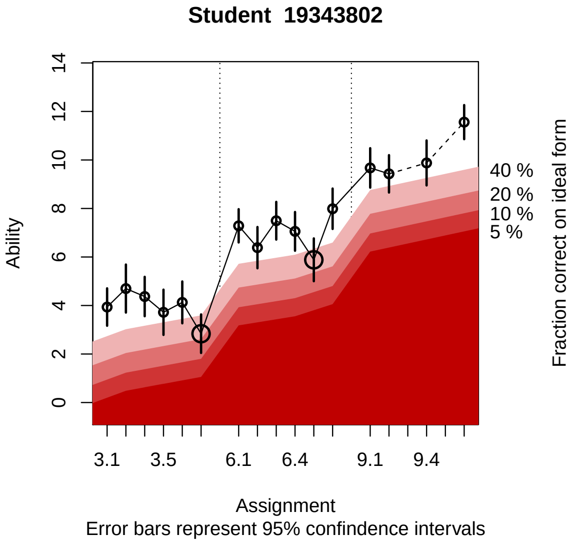


Figure 8: Progress Map with error bars and fixed percentile limits

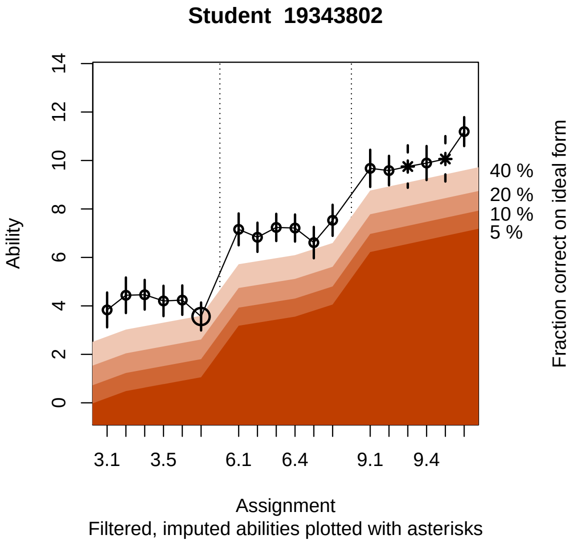


Figure 9: Filtered Progress Map

The filter automatically imputes ability estimates for missing assignments. The student shown in Figure 9 is missing data for assignments 9.3 and 9.5 (plotted with asterisks). The filter imputes abilities based on the previous scores. Note that the standard error grows larger for the imputed values (dashed error bars) but shrinks down when assignment data are again available. Adding thermometer plots across the top showing the completion percentage of each assignment would improve the utility of this display.

6. LIMITATIONS AND IMPROVEMENTS

Building rigorous psychometric models for homework is problematic because homework items are seldom selected with an eye to building a true vertical scale. By assuming that the items are assigned according to a stationary scale, we gain consistency in interpreting estimated student abilities. This allows a growth model to be used to smooth the estimates over time. The additional assumption that growth is similar between any two sections allows the ability estimates to be placed on a progress scale, which has some of the same visual appeal as a true vertical scale.

Stationarity is very much an assumption of convenience. This is both a strength, in that it allows analysis to proceed without a true vertical scale, and a weakness, in that without verification it adds an unknown bias to the ability estimates. Consequently, we only recommend the use of unverified stationary scales for low-stakes purposes, such as student ability tracking by student or instructor. High-stakes uses of the stationary scale would require verification of the stationarity assumption.

A key limitation is that the verification of the stationarity assumption requires the same kind of anchor item design as building a true vertical scale. This is part of a fundamental model identification issue: if the scale at each time slice is not identified, then neither is δ_t, the average growth between time slices [1].

The stationary scale supports a variety of techniques, such as the Kalman filter, for smoothing ability estimates. We imagine that instructors will find smoothed graphs (e.g., Figure 9) more useful than unsmoothed graphs (e.g., Figure 8). A logical next step would be to evaluate Progress Map usability with instructors.

This paper modeled student ability with a single continuous variable, but stationarity generalizes in a straightforward way to the multidimensional case. The Kalman filter works as well for multidimensional normal models of proficiency. Other models, for example replacing the model of Figure 1 with a dynamic Bayesian network [5], fit into the general framework. The Kalman filter is no longer appropriate for smoothing, but the particle filter is adaptable to a wide variety of representations.

Further work is required in estimating the parameters of the growth model. The scale identification problem is insurmountable without the stationarity assumption (or a true vertical scale). But even under stationarity, estimating the growth model variance of the innovations $\omega^2\Delta t$ can be tricky. Although the Brownian motion model implies that the population variance should increase over time, that was not the case for our data set. This may be due to student attrition; the number of students actively completing assignments dropped from approximately 550 to 450 over the semester, and it is likely that the drop-outs were predominantly lower-ability students.

7. ACKNOWLEDGMENTS

We thank Tom McTavish, Quinn Lathrop, Simcha Knif, Brett van de Sande and Kurt VanLehn for feedback and helpful suggestions.

All opinions expressed in this paper are those of the authors and do not represent official positions of Florida State University or Pearson Education.

8. REFERENCES

[1] R. Almond, U. Tokac, and S. Al Otaiba. Using POMDPs to forecast kindergarten students reading comprehension. In J. M. Agosta, A. Nicholson, and M. J. Flores, editors, *The 9th Bayesian Modelling Application Workshop at UAI 2012*, Catalina Island, CA, August 2012.

[2] R. G. Almond. Cognitive modeling to represent growth (learning) using Markov decision processes. *Technology, Instruction, Cognition and Learning (TICL)*, 5:313–324, 2007.

[3] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[4] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and control.* Holden-Day, 1976.

[5] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computer Intelligence*, 5:142–150, 1989.

[6] H. Doran, D. Bates, P. Bliese, and M. Dowling. Estimating the multilevel rasch model: With the lme4 package. *Journal of Statistical Software*, 20(2):1–18, 2007.

[7] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice.* Springer, 2001.

[8] J.-C. Falmagne. A latent trait model via a stochastic learning theory for a knowledge space. *Psychometrika*, 54:283–303, 1989.

[9] H. Huynh and C. Scheider. Vertically moderated standards: background, assumptions, and practices. *Applied Measurement in Education*, 18(1):99–133, 2005.

[10] R. E. Kalman and R. S. Bucy. New results in linear prediction and filtering theory. *Transactions ASME, Series D, J. Basic Eng.*, 83:95–107, 1961.

[11] M. J. Kolen and R. L. Brennan. *Test equating, scaling, and linking: Methods and practices.* Springer, 2004.

[12] R. J. Mislevey and R. Zwick. Scaling, linking, and reporting in a periodic assessment system. *Journal of Educational Measurement*, 49(2):148–155, 2012.

[13] R. Patz. *Vertical scaling in standards-based educational assessment and accountability systems.* Council of Chief State School Officers, 2007.

[14] M. D. Reckase, T. A. Ackerman, and J. E. Carlson. Building a unidimensional test using multidimensional items. *Journal of Educational Measurement*, 25(3):193–203, 1988.

[15] L. Wise and M. Alt. *Assessing vertical alignment.* Council of Chief State School Officers, 2005.

Visualization and Confirmatory Clustering of Sequence Data from a Simulation-Based Assessment Task

Yoav Bergner, Zhan Shu, and Alina A. von Davier
Educational Testing Service
Princeton, NJ 08541
{ybergner,zshu,avondavier}@ets.org

ABSTRACT

Challenges of visualization and clustering are explored with respect to sequence data from a simulation-based assessment task. Visualization issues include representing progress towards a goal and accounting for variable-length sequences. Clustering issues focus on external criteria with respect to official scoring rubrics of the same sequence data. The analysis has a confirmatory flavor; the goal is to understand to what extent clustering solutions align with score categories. It is found that choices related to data preprocessing, distance metric and external cluster validity measures all impact agreement between cluster assignments and scores. This work raises key issues about clustering of educational data, especially in the presence of multidimensionality. Different clustering protocols may lead to different solutions, no one of which is uniquely best.

Keywords

Sequence mining, clustering, visualization, simulation-based tasks, assessment

1. INTRODUCTION

Complex tasks in educational environments are intended to be more engaging for learners and more reflective of real life challenges than traditional test items [22]. In an assessment context at least, the additional time it takes to administer such tasks comes at a certain cost. One hopes therefore that data relating to the process provide more information than the outcome of the task alone. Examples of such process data range in complexity: they may include simple measures like response time [21], multiple attempt records [2], and use of hints [20]; or more expansive processes such as referencing a range of online learning resources [37], keystroke-level writing data [1], or actions taken in a simulation- or game-based task [18, 25]. One broad characterization of the data at the latter end of this list is that they comprise sequences of observable states. Temporal information about the duration of each state may be included or not.

Clustering sequences is a way to detect similar patterns of behavior. In an educational context, the hope is that this structure is informative of some underlying characteristic, perhaps style, perhaps ability. From the perspective of learning scientists and instructional designers, it is important to understand both of these aspects, and from an assessment perspective, it is important to distinguish between them. In other words, patterns in the structure of responses may detect both construct-relevant and construct-irrelevant variance, and the distinction is critical for validity in the interpretation of scores [16].

We consider sequence data from a particular simulation based task, the *Wells* task used by the National Assessment for Educational Progress (NAEP) as part of the Technology and Engineering Literacy (TEL) Assessment [26]. The sequence data from *Wells* are only modestly complex as sequence data go, but their analysis introduces a number of operational choices. To map out the challenges to the data analyst, we organize some of these loosely into challenges of visualizing sequence data (and associated frequency or summary data) and challenges of clustering the sequences.

The *Wells* task is scored along two cognitive dimensions by separate rubrics. Our goal is not to reproduce the results of the rubrics after the fact by alternate means. Instead we ask whether a bottom-up search for patterns in the data comes close to approximating the top-down scoring design in the scores of the sequences, and if not, why not? The analysis thus has a *confirmatory* flavor. We fully expect the two approaches not to meet in the middle, but hope that there may be insights to gain about principles of scoring and/or clustering from the concordance of scores with cluster assignments.

The organization of the paper is as follows: in section 2 we describe related work on clustering and sequence mining. In section 3, we describe the NAEP task, scoring design, and the sequence data. Section 4 introduces two operational choices that affect the visualization of the data, while section 5 addresses choices with respect to clustering. Section 6, describes resulting measures of external validity when comparing cluster assignments to score categories. Section 7 includes a discussion of the results with extensions to future work.

2. RELATED WORK

Clustering student actions is a common approach to various types of educational data. Some recent applications include reading comprehension tasks [28], discussion forum behavior [4, 23], collaborative learning sessions [29], automated speech act detection [33], and strategies in educational games [18, 25]. Most clustering studies operate on feature vectors from logs, not on sequences of states themselves. This is an important distinction. Whether feature vectors are numerical in nature (counts, ratios, etc.) or coded as binary indicators (e.g. [18]), such vectors are all the same length and permit straightforward metrics such as Manhattan, Euclidean, or cosine distance functions. Though their clustering analysis used only feature vectors, exploratory sequential pattern mining also figured in [29]. Time-series data and an agglomerative approach similar to ours was used in [4], but with a key difference. That analysis mapped each possible action type (e.g., reading or writing) to its own binary-valued time-series using an indicator, and only clustered students based on one type of action at a time. Such binary data types do not invoke the same sequence matching issues.

Many of the issues encountered here arose in the context of clustering web sessions from online learning environments in [37], namely: the desire to mine categorical activity sequences, rather than sets or counts of actions; the need to introduce appropriate similarity measures on these sequences; and lastly, the challenges of cluster validation. This work introduced several new algorithms and reported on the performance and scaling of these, but did not say much about cluster interpretation issues. Our approach and toolset is similar to those used by [7] to explore study patterns and identify groups, although there again the analysis was exploratory.

In fact, most if not all of the studies mentioned above used clustering either in an exploratory fashion, or to examine correlation, for example with levels of answers to reading questions [28] or to course pass-fail rates [23]. In our application, by contrast, the student sequences are actually scored by an operational rubric (along two dimensions, as will be discussed). By looking closely at the external validity of our clustering results with respect to rubric-based scores, our analysis has a more confirmatory flavor. This paper thus contributes both a new application of sequence clustering methods in an assessment context and an extension of the discussion on cluster validity with respect to expert-based measures.

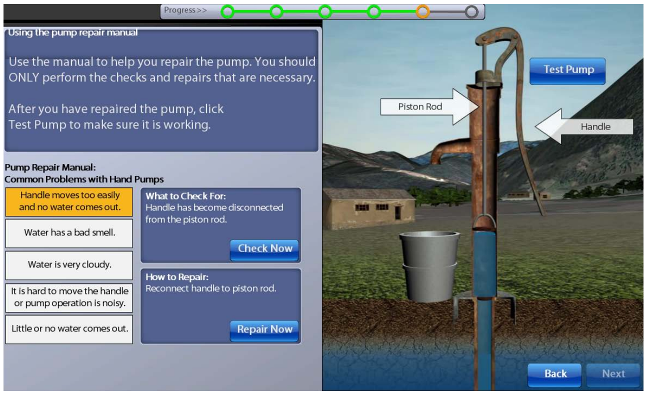
The first part of this paper also concerns ordering a set of actions in a sequence with regard to proximity to the end-goal. Sequences of actions are not always goal oriented, for example when they describe web sessions or studying behaviors. Even in the cases where the activity itself has a goal, it is not always straightforward to tell whether the user activity represents movement towards or away from the goal, especially when the state space is large. Estimating the probabilistic distance to solution in computer programming exercises was the subject of [36]. Networks of states and actions in a logic tutor were analyzed using a novel data structure in [8], and social network methods were used to identify both solution sub-goals and conceptual problem areas. In our application, this task is much easier because the state space is small. However, one can imagine generalizations of the task or other simulation-based task applications, in which these probabilistic methods would be quite useful.

Finally, alternatives to clustering in analysis of sequential data include approaches such as differential sequence mining [24] or the use of hidden or dynamic Markov models [19, 35] to distinguish successful sequences from unsuccessful sequences. Complex feature engineering, as in the design of affect detectors [3, 6], can also account for many of the salient features of sequential data. All of these approaches may be more applicable to open-ended group or individual problem solving sessions than to a task such as ours where success depends deterministically on certain actions and all of the sequences are ultimately successful.

3. DATA FROM THE *WELLS* TASK

The data for our analysis (sample size *N*=1318) come from a pilot administration of TEL tasks by NAEP in 2013. The *Wells* task has been publicly released on the NAEP website [27].

Briefly, *Wells* is designed to elicit efficient and/or systematic behaviors in the diagnosis and repair of a groundwater well in a rural village. Extensive scaffolding is of course provided, as students are not expected to know already how such pumps work or what makes them fail. Through direct instruction and by leading the student to ask a simulated villager certain questions, information is communicated that the well is exhibiting two problems. Eventually the student is presented with an animated view of the well and a set of action choices (buttons) that will ultimately lead to its successful repair.



Using the pump repair manual

Use the manual to help you repair the pump. You should ONLY perform the checks and repairs that are necessary.

After you have repaired the pump, click Test Pump to make sure it is working.

Pump Repair Manual:
Common Problems with Hand Pumps

Handle moves too easily and no water comes out.

Water has a bad smell.

Water is very cloudy.

It is hard to move the handle or pump operation is noisy.

Little or no water comes out.

What to Check For:
Handle has become disconnected from the piston rod.

How to Repair:
Reconnect handle to piston rod.

Figure 1: Screenshot of diagnosis/repair stage in *Wells* task

A screenshot of the diagnosis and repair stage is shown in Figure 1. The student is prompted to consider five common problems. Corresponding to each are buttons to either check the possibly malfunctioning part of the pump or, independently of checking, repair the part. There are thus five check actions (C1, C2, …) and five repair actions (R1, R2, …), and each is allowed at most once. In addition, the student can test whether the pump as a whole is functioning normally. This pump test (P) is the only action that can be repeated.

The two problems that need repair appear (always) in positions four and five, i.e., C4, R4 check and repair one of them, while C5, R5 check and repair the other. Once the broken parts of the pump have been repaired, a pump test ends the task with a success message. As the action set is small and students are allocated plenty of time to complete the task, all students reach the end goal of the task, even if they do so by random guessing.

For example, a student sequence during the diagnosis and repair might be recorded as follows:

C1, C2, C3, C4, R4, P, C5, R5, P

Since the only problems with the well correspond numerically to problem 4 and 5, this sequence might correspond to a student who has not gained (or acted on) the prior knowledge about the problems exhibited by the well. Because such knowledge is possible from the information provided, the sequence C4, R4, P, C5, R5, P is very common. The sequence C5, R5, P, C4, R4, P should presumably be equally good. We will return to this point.

In practice, the sequence of observed actions by the student generates two scores (Efficiency and Systematicity) using two separate rubrics. For the purpose of this analysis, we maintain a semblance of agnosticism about the rubrics themselves. Thus we will refer to these as F-score and Y-score going forward.

4. VISUALIZATION

The classic visual representation of a state-sequence is a graph in which each state is represented by a node and a transition between states is represented by a directed edge (arrow) between nodes. These graphs have some relation to spatial maps, if the location of the node corresponds to the location in space, but the location of nodes in state-space graphs can be more abstract. The formal similarity between state-space graphs and (social) network graphs has invited more than one application of methods of network analysis to sequence mining [8, 34].

State-space or network graphs can represent an accumulation of data from many sequences, for example by using thicker arrows to represent more transitions. But a potential shortcoming arises in cases where a sequence represents progress towards a goal, as illustrated in Figure 2 for the case of the *Wells* task. As shown, randomly placed nodes remove any visual sense of progress toward the end-state goal, and this problem is not easily solved. States that are equally productive or counterproductive and states that recur (for example, the pump test action P) make ordering the node locations impossible.

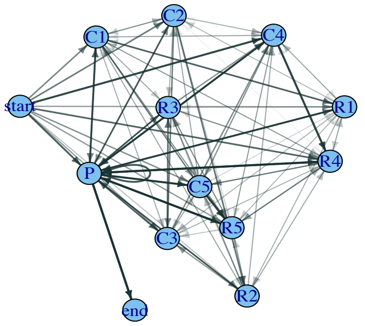


Figure 2: a state-space graph showing all 1318 sequences

4.1 Ordering and Degeneracy of States

One approach is to consider a remapping of the state-space into a new state space that permits such an ordering. This mapping is by design not one-to-one, since two equally good moves can reduce to the same state. It is also not without subjectivity, as will soon be clear. The first step might be to group together favorable moves and unfavorable moves. Actions in the set {C4, C5, R4, R5} are favorable moves in that they either provide confirmation of a failure point (good) or remediation of same (better). Actions in the set {C1, C2, C3} are unfavorable in that it is knowable beforehand that the pump does not have these problems. In that sense {R1, R2, R3} are arguably even worse. Pump testing P is difficult though: it is a good move to test the pump following a (needed) repair, but otherwise it is not particularly useful. Based on these observations, we could collapse all valid checks (VC) and repairs (VR) and invalid checks and repairs (IC, IR), which appears to shrink the state space.

To be sure, making more valid repairs is better (and necessary to reach the goal), while making more invalid checks or repairs is counterproductive. Thus one should probably keep count (*n*VC, *n*VR, for the *n*th valid check, etc.) In fact, this reasoning applies to pump test P, though here is probably where the choice gets most subjective. There are two times that P is called for (after each of the needed repairs; denote these valid pump tests 1VP, 2VP). Other times, pump tests are at best neutral (1IP) or even counterproductive, for example testing the pump more than once in a row or testing it after an invalid check (2IP). With these (subjective) rules in mind, it is possible to map sequences in the original state space {C1, …, R1, …, P} to a new set of sequences, which we call *remap*. The new state space is actually larger (14 states instead of 11), but the states are now ordered with respect to the end-goal:

$$3IR < 2IR < 1IR < 3IC < 2IC < 1IC < 2IP < 1IP < 1VC \\ < 1VR < 1VP < 2VC < 2VR < 2VP$$

With an ordering in hand, visualizing the sequences is as easy as drawing a plot of state position (on the ordered scale) by step number in the sequence.

The results are shown in Figure 3. Each sequence is drawn in partially transparent grey so that the accumulation of multiple overlapping sequences forms darker lines. Students who use no extraneous actions do not dip below the starting point (dashed line) and complete the task in 3-6 steps. A large number of inflection points for a sequence visualized this way might suggest haphazard guessing.

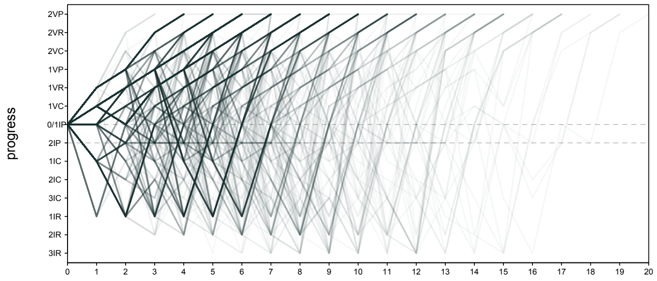


Figure 3: Ordered state position by sequence step (1318 seqs)

The R package TraMineR [10], a toolbox for categorical sequence data originally designed for life trajectory modeling in the social sciences, can be used to generate sequence frequency and state-distribution visualizations such as in Figures 4-5. Because an ordering can be associated with a color-palette, choosing "hotter" colors for negative states and green and blue shades for productive moves makes it possible to read information easily from the plots. It is clear from the frequency plot (Fig 4) that a large group of students complete the task using only the valid check-repair-test actions. Note that only the ten most frequent sequences are shown in the figure; over 500 unique sequences were observed.

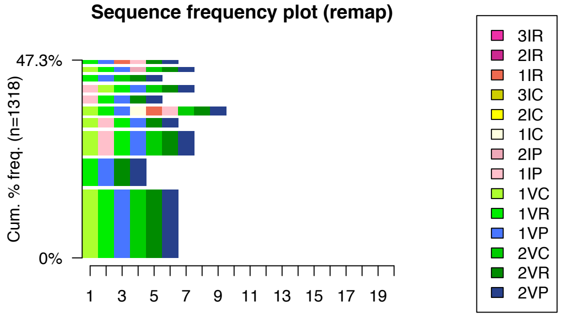


Figure 4: Frequency plot showing 10 most frequent sequences. Colors correspond to the redefined states (see legend).

4.2 Variable Sequence Length

The state-distribution plot in Figure 5 raises an important issue concerning the variable length of sequences in our data set; it is a familiar issue from survival analysis [12]. Consider for example the vertical slice through the plot at step 10. This slice gives the impression that roughly 30% of sequences are entering the final correct state (2VP), another 20% completing the second valid repair (2VR), and the remaining half divided among states behind these in the progression. But it is important to remember that this is the breakdown *only for sequences that continue out to this step number*. In fact, a great number of respondents have already finished the task by this point and so have dropped out of the distribution. The plateau at steps 7-13 belies this fact.

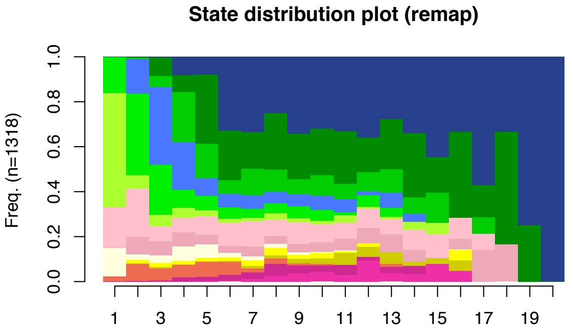


Figure 5: state distribution plot; for color legend, see Fig. 3

An alternative is to coerce the sequences to be of equal length by persisting in the final state until the maximum length is reached. The new state distribution, shown in Figure 6, now reflects the accumulating population of completers, and the plateau at steps 7-13 is not a plateau at all. As we shall see, this manipulation of the sequence data also has a significant effect on clustering results, because it alters the similarity measure between two sequences when standard edit distances are used to compare them.

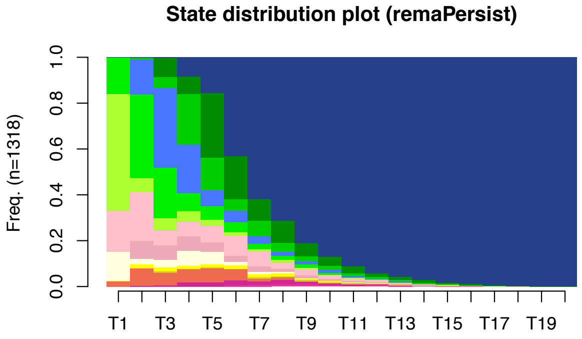


Figure 6: State distribution when the final state is maintained out to a fixed length

We have so far considered two operational choices for pre-processing and visualizing the sequence data in *Wells*: remapping the original sequences using a new ordered set of states, *remap*, and padding out the sequences to a fixed length by repeating (persisting) the final state, *remapP*. Both of these choices can have significant informational impact on the visual representations data. We now turn to the question of whether the sequence data themselves can be seen to self-organize in a structure that is reflected in the scoring designs of the *Wells* task.

5. CLUSTERING

In a taxonomy of data clustering methods [15], the first branch point separates agglomerative from partitional methods. Briefly, partitional approaches start with one large cluster and divide it once according to some algorithm and similarity measure. A canonical example is *k-means* clustering, but fuzzy clustering or expectation-maximization based mixture resolving are also partitional schemes. An agglomerative approach on the other hand starts with each datum as its own cluster and then groups them progressively in a nested structure (dendrogram) until one cluster is obtained. One advantage of this approach is that a single dendrogram can be cut at various levels, resulting in a deterministic refinement of clusters. This approach is suitable for our purposes, because we wish to compare cluster assignments in a confirmatory sense to categorical scores, and these categorical score levels can also be agglomerated based on cut scores. For example, we will make a case in Section 6.1 to agglomerate a five level F-score into either three levels or two.

Hierarchical agglomerative clustering (we used the agnes method in the R package cluster) requires the specification of both a metric (or equivalently, a dissimilarity matrix) and a linkage algorithm, e.g., single-link, complete-link or Ward's method [15].

5.1 Defining (Dis)similarity

A distance defined between two sequences is highly related to the notion of string edit distance. Using the TraMineR package [9], we consider longest common subsequence (LCS), longest common prefix (LCP), optimal matching (OM) and simple Hamming distance (HAM), which are described in detail in [10]. LCS distance (not to be confused with the *LCS problem*) is equivalent to Levenshtein distance with only insertions and deletions (indel cost 1), no substitutions. In optimal matching, one also specifies a substitution matrix. For example, the substitution cost may be computed based on transition rates, in order to accentuate rare events. With a fixed substitution cost of 2 and indel cost of 1, the OM distance metric is equivalent to LCS. We used OM with a fixed indel cost of 3 to distinguish this metric, as illustrated below. Consider the sequences defined in Table 1 in both their original and remapped representation:

Table 1: Example sequences under original and remap states

Sequence	Original	*remap* representation
S1	C4,R4,P,C5,R5,P	1VC,1VR,1VP,2VC,2VR,2VP
S2	C5,R5,P,C4,R4,P	1VC,1VR,1VP,2VC,2VR,2VP
S3	C4,R4,P,C1,C5,R5,P	1VC,1VR,1VP,1IC,2VC,2VR,2VP
S4	C4,R4,P,P,C5,R5,P	1VC,1VR,1VP,2IP,2VC,2VR,2VP

Sequences S1 and S2 differ only by "equivalent" choices between whether to operate on issue 4 or issue 5 first. Compared with S1, S3 and S4 each insert one extra action, either an invalid check or an extra test. The distances between each pair of example sequences in Table 1 are shown for selected combinations of data representation and distance metric in Table 2.

Table 2: Distances between sequences under different metrics

Distance	S1-S2	S1-S3	S1-S4	S2-S3	S2-S4	S3-S4
orig.LCS	6	1	1	7	7	2
remap.LCS	0	1	1	1	1	2
remapP.LCS	0	2	2	2	2	2
remap.OM	0	3	3	3	3	2

Note that the *remap* representation erases the difference between the S1 and S2, by design. Note also that in *remapP.LCS*, all single insertions (e.g. S1 to S3) have the same cost as a substitution (S3 to S4), because in a fixed-length sequence, one cannot insert an element without removing one of the persisting states at the end. The choice of representation (*remap* vs. *remapP*) thus has an effect on the distance, even when the same LCS metric is used. To counterbalance this effect, we introduce a higher indel cost for OM (see last row of Table 2). Since we have considered a persisting variant of the *remap* sequences, we also included an *origP* representation, in which the original states are used but padded out to fixed length.

One can thus form all possible combinations of representations (*orig, origP, remap, remapP*) and dissimilarity measures and finally choose a linkage algorithm for hierarchical agglomerative clustering. Including single-linkage, complete-linkage, and Ward's methods gave a total of 42 combinations. Each combination results in a dendrogram, which can be cut to produce a cluster assignment for any target number of clusters.

Clustering may be evaluated using internal criteria—essentially how meaningful is the partition—or external criteria, such as how well does the partitioning agree with some ground-truth label. We are interested in external criteria with respect to operational scores in the task. This raises a set of issues we describe next.

5.2 Relating Cluster Analysis to Scoring

Comparing cluster assignments with rubric-based scores on the *Wells* task is complicated by several factors: multidimensionality of the score, number and ordering of categories to be matched, and chosen measure of comparison. We outline the issues here and address them further in the results section.

The multidimensionality issue arises because cluster analysis does not result in a multidimensional assignment, whereas the rubric assigns to each sequence both an F-score (five levels) and a Y-score (four levels). Although canonical correlation analysis [13] and MANOVA options exist, a reasonable first step is to take the scores one at a time and compare the clustering assignments to each. As we shall show, a single clustering algorithm may not be separately optimal for both scores.

Cluster labels are inherently nominal, while the rubric scores are ordered categories. We may of course discard the ordering information in the scores themselves and use a purely nominal association measure, such as Goodman and Kruskal's τ [11]. But while a clustering algorithm has no way of rank-ordering the clusters, we believe *a priori* that an underlying ordering exists if both the clusters and the scoring rubric have any validity. One way to derive an ordering of the clusters is by the mean score of the cluster members. We thus consider a set of measures that treat the cluster label as either nominal or ordinal.

A standard ANOVA yields a measure of score variance explained by nominal cluster label, namely R^2. If we order the clusters first by mean score, a linear regression model on the ordered categories also yields an R^2. Along with τ, these measures have the advantage that the number of clusters does not have to match the number of score categories.

For completeness, and to make contact with standard approaches in classifier performance, we also consider "agreement" types of measures. In particular, we add Cohen's weighted κ [5] (using squared off-diagonal weights), Precision, and Recall. A detailed discussion of the merits, biases, and internal relationships of many classifier evaluation measures can be found in [31]. In any case, use of these measures requires that the number of clusters be selected to match the number of score categories. This is acceptable, since in our confirmatory approach, we do not try to identify the optimal number of clusters.

There are some *post hoc* justifications for "agglomerating" some of the F-score levels, based on the pilot data, before choosing the number of clusters. The distributions for both scores in our pilot data are shown in Figure 7. F-score levels 1 and 4 are very sparsely populated (around 5% in each). The rationale is that if the level 4 data are construed as boundary cases between levels 3 and 5, rather than genuine categories, then looking for a cluster assignment that correctly identifies them is stacking the deck against the clustering algorithm. Moreover, starting out with smaller numbers of categories for F-score has the further benefit of simplicity, especially in visualizations.

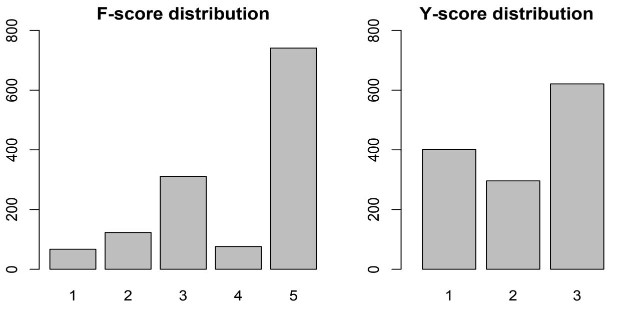


Figure 7: Distribution of scores in the pilot data (N=1318)

We consider partitions of F-score into two and three levels. The two-level F-score introduces a single cut, F ≤ 3, F > 3, while the three-level version introduces a second cut, F < 3, F = 3, F ≥ 4. We examine agreement measures with repartitioned F-score for two- and three-cluster solutions. For Y-score, we consider only the three-cluster case. Because hierarchical agglomerative clustering is deterministic, the cluster assignments that result from different cuts of the dendrogram are stable.

The subject of cluster validity is covered in many references, for example [14, 30, 32]. Our brief treatment of the subject here is meant only to highlight some examples of the issues that arise in our application.

6. RESULTS

6.1 Alignment of Clustering with F-score

Table 3: 2-cluster cuts with 2-level F-scores (sorted by R^2)

	Method	R^2	τ	κ	Prec	Recall
1	ward.remapP.LCS	0.73	0.44	0.88	0.89	0.91
2	ward.remapP.OM	0.57	0.26	0.71	0.84	0.76
3	ward.origP.LCS	0.55	0.23	0.67	0.85	0.72
4	ward.orig.OM	0.47	0.20	0.62	0.81	0.71
5	ward.origP.OM	0.47	0.21	0.61	0.82	0.70
6	ward.remap.OM	0.42	0.14	0.49	0.84	0.63
7	ward.remapP.HAM	0.37	0.17	0.52	0.76	0.66
8	complete.orig.OM	0.24	0.07	0.35	0.92	0.55
…	…	…	…	…	…	…
11	ward.remap.LCS	0.22	0.19	0.26	0.86	0.63
…	…	…	…	…	…	…

Results for two-cluster comparison with two-level F-score are shown in Table 3, ordered by R^2. In this simple case, the R^2 from ANOVA and from a linear model are necessarily the same, and weighted κ is identical to unweighted κ. We point out a few salient features of Table 3. First, the Ward's method algorithm leads to the top seven clustering assignments. Second, it is apparent by inspection that the measure variables are almost perfectly monotonic; the rank correlations are high. Ward clustering of the *remapP* sequences using the LCS metric scored highest in all measures. Interestingly, it scored much higher than the same method used on the *remap* sequences, which differ only in the persistence of the final state. As we saw in Table 2, the computed distance between two sequences does change depending on the representation. In Figure 8 we examine the concordance effects visually.

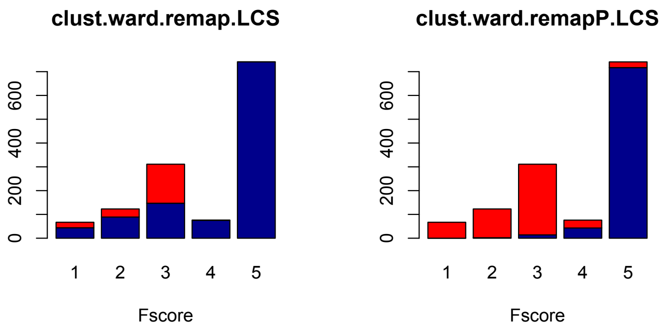


Figure 8: Comparison of cluster assignments (red/blue) with F-score for different representations using the same method.

Even though F-scores were aggregated for the calculation of agreement measures, we have left all of the original levels in Figure 8 for illustrative purpose. The effect of data preprocessing is quite noticeable. Using *remap* (no persistent final state), the two-cluster solution does not achieve good separation in the lower F-score levels, though no high F-scores fall into the red cluster as false positives. On the other hand, a small number of high F-scores are misclassified by the assignment using *remapP*. Those sequences typically contained many extra pump tests (P), often in a row. From the clustering algorithm's "point of view," these sequences had more in common with other extraneous moves, though from the task designer's point of view, extra pump testing was not penalized on efficiency.

The *ward.remapP.LCS* dendrogram is still the best performer at three clusters (R^2 = 0.78) and five (R^2 = 0.78), not shown. However the monotonicity, or rank-correlation, among the measures degrades as cluster number increases. This issue arises in the Y-score results and is discussed in the next section.

6.2 Alignment of Clustering with Y-score

We now consider the second score dimension for *Wells*. Although the rubric describes four levels, the pilot test data, as shown in Figure 7, only contain three levels in a slightly U-shaped distribution. The association and agreement measure table for three-cluster dendrogram cuts with Y-score is shown in Table 4.

Table 4: 3-cluster cuts with 3-level Y-scores (sorted by R^2)

	Method	R^2_{anova}	R^2_{linear}	τ	κ	Prec	Recall
1	ward.origP.LCS	0.45	0.44	0.30	0.66	0.55	0.58
2	complete.remap.LCP	0.38	0.38	0.23	0.61	0.60	0.54
3	ward.remap.LCP	0.35	0.34	0.22	0.54	0.50	0.48
4	ward.remapP.LCP	0.34	0.34	0.24	0.51	0.61	0.39
5	ward.orig.LCP	0.33	0.32	0.21	0.51	0.50	0.46
6	ward.origP.OM	0.31	0.31	0.21	0.55	0.48	0.51
7	ward.origP.LCP	0.30	0.30	0.22	0.47	0.62	0.38
…	…	…	…	…	…	…	…
12	ward.remapP.LCS	0.08	0.08	0.06	0.22	0.54	0.41

Besides the fact that overall external agreement is worse in the case of Y-score, a few other details are worth noting. Our ordering heuristic, i.e. using the cluster means, appears to be reasonable, given the near-perfect correlation of the two R^2s. The *remap* sequences are no longer clear winners; in fact the best performing dendrogram with respect to F-scores placed 12th in this table. Also there is no longer monotonicity between the measures, which is especially clear from looking at the τ and Precision columns. This is important, as is illustrated in Figure 9, in which three different cluster assignments are compared side by side. The plots from left to right correspond to dendrograms of rows 1, 2, and 4 in Table 4.

The leftmost assignment is the one with the highest R^2, τ, and κ score. One of the clusters here, shown as dark blue areas, does not discriminate at all between levels of Y-score. The yellow areas correspond to a cluster that reasonably captures the top Y-score, whereas the red cluster comprises mostly level 1 and 2. This assignment would probably have scored even better if the lowest two levels of Y-score were combined into one. Indeed, none of the solutions shown appear to identify three clusters that associate convincingly with each of the three score categories.

The middle plot in Figure 9 shows a clustering solution with

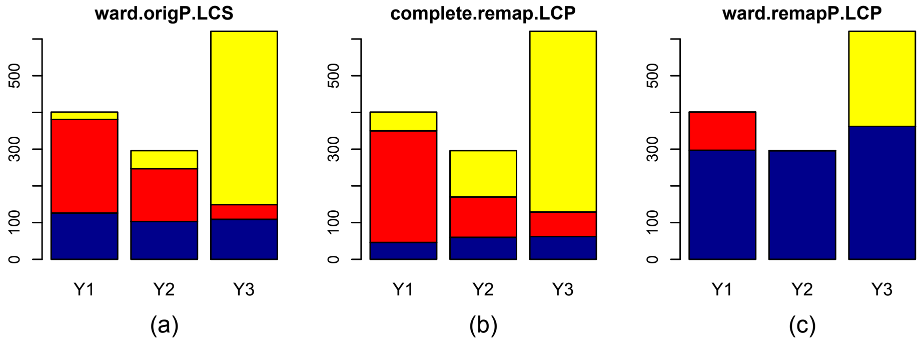


Figure 9: Visualization of the cross-tabulation from three cluster assignments with Y-score. Plots (a), (b), and (c) correspond to dendrograms of rows 1, 2, and 4 in Table 4, respectively.

lower R^2, τ, and κ, but a higher Precision. It also has a non-discriminating cluster, though here it is smaller. The yellow and red clusters are more or less equally split on the mid-level score.

Finally the rightmost clustering has slightly higher precision, but low recall, R^2 and κ. According to τ, it is the second best match to the scores. One is tempted to say that this cluster assignment "avoids getting it wrong." Although many more sequences are put into the non-discriminating blue cluster, including all of the mid-level Y-score, the red and yellow clusters have no false positives at all. Depending on the purpose, for example routing in a multi-stage assessment [38], it might be argued that this "diagnostic" clustering is preferable.

After varying the data representation, distance metric and even linkage function, examining alignment of clustering solutions with Y-score turns out to be rather subtle.

6.3 Alignment of Clustering with Both Scores

The best cluster dendrogram for F-score is a poor performer with respect to Y-scores (and vice-versa). Using MANOVA with both scores simultaneously, *ward.remapP.LCS* is still the winner (likewise if a combined six-level FY-score—two F-score levels and three Y-score levels—is matched to a six-cluster cut). It wins despite not resolving the Y-scores well, just on account of resolving the F-score as well as it does. The logical conclusion to draw from this is that, in the case of multidimensional scores, there is no one best clustering assignment. The appropriate clustering method and preprocessing of the data indeed depend on the intended purpose.

7. DISCUSSION AND FUTURE WORK

We have tried to show some of the operational issues that arise in characterizing sequence data from a simulation-based task, specifically visualization and clustering choices. With respect to visualization, we were concerned with representing progress as unambiguously as possible, and we explored the consequences of both mapping the original sequences to an ordered set of states and padding out the sequences to a fixed length.

With respect to clustering, we were most interested in measures of external agreement with the two-dimensional scoring rubric designed for the task. We found that the clustering dendogram that worked best in terms of one score did not necessarily work at all in terms of the other. Both data preprocessing and selection of the between-sequence distance metric had an impact, though the best agglomerative linkage algorithm was almost always Ward's method. In the case of Y-score, while none of the solutions were great, we found it ambiguous to tell which was even the best among the mediocre. Whether this reflects a feature of the Y-scoring that is ambiguous or just difficult to capture via sequence clustering is something we wish to investigate further.

This work raises important issues about clustering of educational sequence data in the presence of multidimensionality: two different clustering protocols may reach different solutions, both of them valid. Furthermore, brute force search among clustering solutions for a best fit according to one particular external criterion may exclude solutions of interest. In practice, what this of course suggests is that the use of sequence clustering methods for inference needs to be handled with care.

Without a doubt, much prior knowledge goes in to preprocessing educational data already. For example, we often simply exclude events we are not interested in. In the context of sequential data, an alternative might be to assign selective weights to particular insertions, deletions and substitutions of states. The web-page similarity index in [37] is designed to address this issue, because substituting one web page with a very different one should be treated distinctly from substituting similar pages. In our case, for example, a variable insertion cost for pump test actions P would have affected the agreement of cluster assignment with F-score.

F-score and Y-score indeed stand for real constructs in the rubric design: efficiency and systematicity. We found that grouping by efficiency can be discovered through sequence clustering, but systematicity was not as well matched. If sequences with the same score do not self-group under edit distance, these discrepancies may merit closer examination. This is the confirmatory value of performing such an analysis.

The edit-distance similarity measures that we used here do not embed sequence data in a multidimensional coordinate space, whereas feature-vector descriptions of sequences do. The latter approach might have several advantages when the external measure is also multidimensional, as in the case of our expert-based scores. Methods like canonical analysis [13] may be brought to bear on such multivariate data. Standard distance metrics also make internal cluster quality analysis more straightforward. While we did not delve here into such measures, it turns out that many internal cluster criteria are not well suited to the use of generalized dissimilarity, for example because a cluster "centroid" is not easily defined. Some authors have cautioned against using Ward's method with non-Euclidean distances because of interpretability problems [17], although this prohibition would have removed the best clustering solutions—by external criteria—in our study. The lack of appropriate internal indices is an unsolved problem that we plan to investigate further.

We note that ordering effects in the data were likely introduced by the fixed order of presentation in the task itself. The five sets of buttons corresponding to possible problems with the well were presented vertically and always in the order corresponding to the codes numbered 1-5. Within the sequence data, the two valid checks, C4 and C5, occurred in that order nine times as often as the reverse, and unnecessary check C2 preceded C3 almost four times as often as the reverse. A randomized order of presentation would have produced more balanced sequence data, and this might have enlarged the effect of remapping the sequences.

We did not look at specific time or duration in this investigation at all. From the perspective of trying to understand student behavior, it might make a significant difference whether the student clicked through options quickly in a task or deliberated before a decision. Such behaviors are part and parcel of sequence mining efforts in, for example, affect detectors [3, 6] or keystroke analysis [1]. The inclusion of temporal variables to sequence clustering and validation is a natural extension of this work.

8. REFERENCES

[1] Almond, R.G., Deane, P., Quinlan, T., Wagner, M. and Sydorenko, T. 2012. A Preliminary Analysis of Keystroke Log Data from a Timed Writing Task. *ETS Reseach Report RR-12-23*. November (2012).

[2] Attali, Y. 2010. Immediate Feedback and Opportunity to Revise Answers: Application of a Graded Response IRT Model. *Applied Psychological Measurement*. 35, 6 (Oct. 2010), 472–479.

[3] Baker, R.S.J.D., Corbett, A.T., Roll, I. and Koedinger, K.R. 2008. Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction*. (2008), 1–36.

[4] Cobo, G., García-Solórzano, D., Santamaria, E., Moran, J.A., Melenchon, J. and Monzo, C. 2011. Modeling students' activity in online discussion forums: a strategy based on time series and agglomerative hierachical clustering. *Proceedings of 4th International Conference on Educational Data Mining*. (2011).

[5] Cohen, J. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*. 70, 4 (1968), 213–220.

[6] D'Mello, S.K., Craig, S.D., Witherspoon, A., McDaniel, B. and Graesser, A. 2007. Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction*. 18, 1-2 (Dec. 2007), 45–80.

[7] Desmarais, M. and Lemieux, F. 2013. Clustering and Visualizing Study State Sequences. *Proceedings of 6th International Conference on Educational Data Mining*. (2013).

[8] Eagle, M., Johnson, M. and Barnes, T. 2012. Interaction Networks: Generating High Level Hints Based on Network Community Clustering. *Proceedings of the 5th International Conference on Educational Data Mining*. (2012), 164–167.

[9] Gabadinho, A. 2011. Analyzing and visualizing state sequences in R with TraMineR. *Journal Of Statistical Software*. 40, 4 (2011).

[10] Gabadinho, A., Ritschard, G., Studer, M. and Muller, N.S. 2010. *Mining sequence data in R with the TraMineR package*. University of Geneva.

[11] Goodman, L. and Kruskal, W. 1954. Measures of Association for Cross Classifications. *Journal of the American Statistical Association*. 49, 268 (1954), 732–764.

[12] Hosmer, D.W., Lemeshow, S. and May, S. 2011. *Applied Survival Analysis: Regression Modeling of Time to Event Data*. John Wiley & Sons.

[13] Hotelling, H. 1936. Relations between two sets of variates. *Biometrika*. 28, 3/4 (1936), 321–377.

[14] Jain, A.K. and Dubes, R.C. 1988. *Algorithms for clustering data*. Prentice Hall.

[15] Jain, A.K., Murty, M.N. and Flynn, P.J. 1999. Data clustering: a review. *ACM computing surveys (CSUR)*. 31, 3 (1999), 264–323.

[16] Kane, M.T. 2001. Current Concerns in Validity Theory. *Journal of Educational Measurement*. 38, 4 (2001), 319–342.

[17] Kaufman, L. and Rousseeuw, P.J. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley & Sons.

[18] Kerr, D., Chung, G. and Iseli, M. 2011. The feasibility of using cluster analysis to examine log data from educational video games. *CRESST Report 790*. April (2011).

[19] Köck, M. and Paramythis, A. 2011. Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*. 21, 1-2 (Jan. 2011), 51–97.

[20] Lee, Y.-J., Palazzo, D.J., Warnakulasooriya, R. and Pritchard, D.E. 2008. Measuring student learning with item response theory. *Physical Review Special Topics - Physics Education Research*. 4, 1 (Jan. 2008), 1–6.

[21] Van der Linden, W.J. 2009. Conceptual Issues in Response-Time Modeling. *Journal of Educational Measurement*. 46, 3 (Sep. 2009), 247–272.

[22] Lombardi, M. 2007. Authentic learning for the 21st century: An overview. *Educause learning initiative*. (2007).

[23] López, M.I., Luna, J.M., Romero, C. and Ventura, S. 2012. Classification via Clustering for Predicting Final Marks Based on Student Participation in Forums. *Proceedings of the 5th International Conference on Educational Data Mining*. (2012).

[24] Martinez-Maldonado, R. 2013. Data mining in the classroom: Discovering groups strategies at a multi-tabletop environment. *Proceedings of the 6th International Conference on Educational Data Mining*. (2013), 121–128.

[25] Mislevy, R.J., Oranje, A., Bauer, M.I., Von Davier, A., Hao, J., Corrigan, S., Hoffman, E., DiCerbo, K.E. and John, M. 2014. Psychometric considerations in game-based assessment. *GlassLab Report*. (2014).

[26] NAEP TEL - Technology and Engineering Literacy Assessment: *http://nces.ed.gov/nationsreportcard/tel/*. Accessed: 2014-02-20.

[27] NAEP TEL - Wells Sample Item: *http://nces.ed.gov/nationsreportcard/tel/wells_item.aspx*. Accessed: 2014-02-20.

[28] Peckham, T. and McCalla, G. 2012. Mining Student Behavior Patterns in Reading Comprehension Tasks. *Proceedings of the 5th International Conference on Educational Data Mining*. (2012).

[29] Perera, D., Kay, J., Koprinska, I. and Zaiane, O. 2009. Clustering and sequential pattern mining of online collaborative learning data. *IEEE Transactions on Knowledge and Data Engineering*. (2009), 1–14.

[30] Powers, D. 2007. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. *Technical Report SIE-07-001*. December (2007).

[31] Powers, D.M.W. 2007. Evaluation : From Precision , Recall and F-Factor to ROC , Informedness , Markedness & Correlation. December (2007).

[32] Reilly, C., Wang, C. and Rutherford, M. 2005. A rapid method for the comparison of cluster analyses. *Statistica Sinica*. 15, (2005), 19–33.

[33] Rus, V., Moldovan, C. and Graesser, A.C. 2012. Automated Discovery of Speech Act Categories in Educational Games. *Proceedings of the 5th International Conference on Educational Data Mining*. (2012).

[34] Shreim, A., Grassberger, P., Nadler, W., Samuelsson, B., Socolar, J. and Paczuski, M. 2007. Network Analysis of the State Space of Discrete Dynamical Systems. *Physical Review Letters*. 98, 19 (May 2007), 198701.

[35] Soller, A. and Stevens, R. 2007. Applications of Stochastic Analyses for Collaborative Learning and Cognitive Assessment. *Institute for Defense Analyses*. IDA D-3421 (2007).

[36] Sudol, L.A., Rivers, K. and Harris, T.K. 2012. Calculating Probabilistic Distance to Solution in a Complex Problem Solving Domain. *Proceedings of the 5th International Conference on Educational Data Mining*. (2012), 144–147.

[37] Wang, W. and Zaïane, O. 2002. Clustering web sessions by sequence alignment. *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*. (2002).

[38] Yan, D., von Davier, A.A. and Lewis, C. eds. 2014. *Computerized Multistage Testing: Theory and Applications*. Chapman and Hall/CRC.

Who's in Control?: Categorizing Nuanced Patterns of Behaviors within a Game-Based Intelligent Tutoring System

Erica L. Snow
Arizona State University
Tempe, AZ, USA
Erica.L.Snow@asu.edu

Laura K. Allen
Arizona State University
Tempe, AZ, USA
LauraKAllen@asu.edu

Devin G. Russell
Arizona State University
Tempe, AZ, USA
Devin.Russell@asu.edu

Danielle S. McNamara
Arizona State University
Tempe, AZ, USA
Danielle.McNamara@asu.edu

ABSTRACT

The authors use dynamical analyses to investigate the relation between students' patterns of interactions with various types of game-based features and their daily performance. High school students (n=40) interacted with a game-based intelligent tutoring system across eight sessions. Hurst exponents were calculated based on students' choice of interactions with four types of game-based features: generative practice, identification mini-games, personalizable features, and achievement screens. These exponents indicate the extent to which students' interaction patterns with game-based features are random or deterministic (i.e., controlled). Results revealed a positive relation between deterministic behavior patterns and daily performance measures. Further analyses indicated that students' propensity to interact in a controlled manner varied as a function of their commitment to learning. Overall, these results provide insight into the potential relations between students' pattern of choices, individual differences in learning commitment, and daily performance in a learning environment.

Keywords

Intelligent Tutoring Systems, dynamical analyses, strategy performance, game-based learning

1. INTRODUCTION

Students' behaviors during learning tasks vary both as a function of the student and the task. Some students approach learning tasks in a decisive manner, revealing a plan and purpose. These students are controlling and regulating their behavior: a crucial skill for academic success [1 – 6]. However, other students can approach the same task in an impetuous manner, showing little discernible schemes or methods. These students are failing to take control of their own learning behaviors; consequentially, their academic success often suffers [1, 7 – 8]. The emergence of students' ability to set decisive goals, plans, and make decisions during a task is often referred to as self-regulation [5].

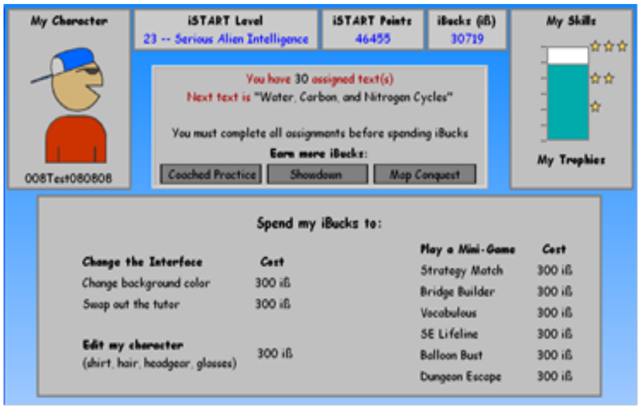
One important component of self-regulation is students' ability to control their choices and behaviors during learning tasks [8]. To gain a deeper understanding of how self-regulated learning manifests in students' choices, scientists have begun to examine patterns that emerge in students' behaviors while they engage with adaptive environments [9 – 10]. These environments produce log data (e.g., keystroke or mouse click data) that are rich in information about what students choose to do while engaged with the system. Analyzing patterns that emerge within log data has been shown to shed light upon the amount of agency exerted during tasks. The utilization of log data is especially useful for researchers interested in examining how students engage with game-based systems, which typically offer students high levels of agency. As students engage with game-based environments, they are frequently provided with multiple choices and trajectories. These variations allow students to exhibit several levels of control, which influence the interaction patterns that manifest during their time within the system. Consequently, these environments provide researchers a unique opportunity to examine students' ability to control their learning experience and the ultimate impact this skill has on learning outcomes.

The ability to effectively self-regulate is challenging for many students, as they often struggle to set their own learning goals and control their behaviors during learning tasks. As a result, self-regulation skills (e.g., ability to control behaviors) tend to vary widely among students [11]. Thus, it is critical to understand what individual differences drive various interaction patterns that may be indicative of students' ability to control their behaviors. Historically, individual difference researchers have shown that students vary in the way that they learn and interact in the classroom [12 – 14]. More recently, it has been shown that individual differences, such as expectations of technology, prior reading ability, and commitment to learning, similarly influence students' interactions and performance within adaptive learning environments [15 – 17].

The current study builds up upon this work by investigating the extent to which students' patterns of interactions display deterministic and controlled properties, and how those properties ultimately impact daily performance outcomes. Additionally, we investigate whether these interaction patterns vary as a function of individual differences in students' commitment to learning, prior reading ability, or expectations of technology. By investigating students' propensity to interact in controlled (i.e., deterministic) patterns within learning environments, our goal is to enhance theoretical understandings of self-regulation and its ultimate impact on learning gains.

1.2 iSTART-ME

The context of this study is iSTART-ME (Interactive Strategy Training for Active Reading and Thinking-Motivationally-Enhanced), a game-based Intelligent Tutoring System (ITS) designed to improve students' reading comprehension skills by providing them with instruction and practice on how to use self-explanation and comprehension strategies [18]. This game-based tutoring system was built upon a traditional ITS (i.e., that was not game-based) called iSTART [19]. iSTART and iSTART-ME are similar in that they both introduce students to self-explanation strategies, demonstrate the use of these strategies, and allow students to practice applying self-explanation strategies to science texts. This scaffolding is conducted in three separate modules (for more detail about these modules and the original iSTART system, please see [20 -21]).


My Character

008Test080808

iSTART Level
23 -- Serious Alien Intelligence

iSTART Points
46455

iBucks (iß)
30719

My Skills

My Trophies

You have 30 assigned text(s)
Next text is "Water, Carbon, and Nitrogen Cycles"

You must complete all assignments before spending iBucks
Earn more iBucks:
Coached Practice | Showdown | Map Conquest

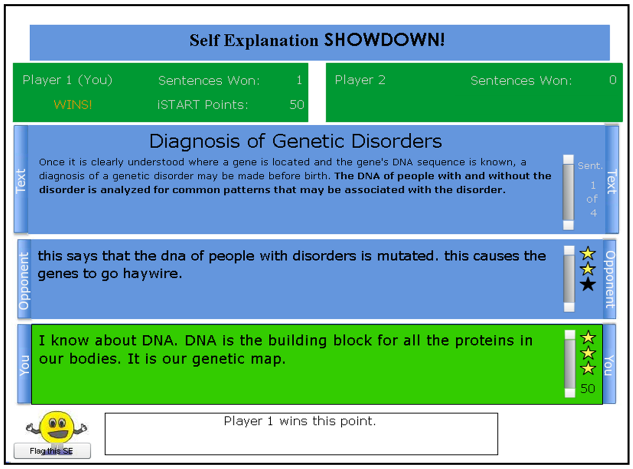
Spend my iBucks to:

Change the Interface	Cost
Change background color	300 iß
Swap out the tutor	300 iß
Edit my character (shirt, hair, headgear, glasses)	300 iß

Play a Mini-Game	Cost
Strategy Match	300 iß
Bridge Builder	300 iß
Vocabulous	300 iß
SE Lifeline	300 iß
Balloon Bust	300 iß
Dungeon Escape	300 iß

Figure 1. Screen shot of iSTART-ME Selection Menu

iSTART-ME builds upon the original iSTART system by adding in game-based practice. This game-based environment provides an opportunity for extended practice and was designed to enhance students' motivation and persistence during extended training sessions (see Figure 1; [21 -22]). Within this game-based practice environment, students can choose to interact with the interface in a variety of ways, such as reading and self-explaining new texts within the context of a game (see Figure 2 for a screenshot of a generative game), personalizing the system interface, or playing identification mini-games (see Figure 3 for screenshot of a mini-game; for a more detailed description of the iSTART-ME system, please see [18]). iSTART-ME presents students with a variety of activities they can choose from. This flexibility puts iSTART-ME in a unique position to assess the agency exhibited within students' patterns of interactions and how those various patterns ultimately impact learning.


Self Explanation SHOWDOWN!

Player 1 (You) Sentences Won: 1
WINS! iSTART Points: 50

Player 2 Sentences Won: 0

Text

Diagnosis of Genetic Disorders

Once it is clearly understood where a gene is located and the gene's DNA sequence is known, a diagnosis of a genetic disorder may be made before birth. **The DNA of people with and without the disorder is analyzed for common patterns that may be associated with the disorder.**

Sent. 1 of 4

Opponent: this says that the dna of people with disorders is mutated. this causes the genes to go haywire.

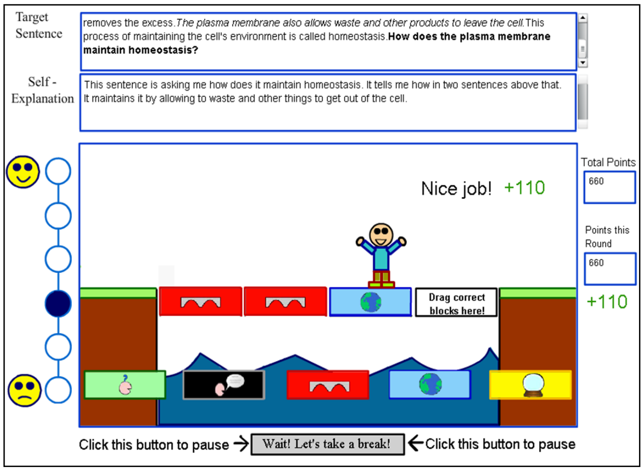
You: I know about DNA. DNA is the building block for all the proteins in our bodies. It is our genetic map. 50

Flag this SE

Player 1 wins this point.

Figure 2. Screen Shot of Generative Practice Showdown

iSTART-ME assesses students' self-explanations through the use of a feedback algorithm [19]. Self-explanations are scored on a scale that ranges from 0 to 3. A score of "0" is assigned to any self-explanation that is composed of irrelevant information or is considered too short. A score of "1" indicates that the self-explanation relates to the sentence but does not elaborate upon the information within the text. A score of "2" is assigned when students' self-explanations incorporate information from other locations in the text beyond the target sentence. Finally, a score of "3" indicates that the self-explanation incorporates information from both the text and students' prior knowledge.


Target Sentence: removes the excess.*The plasma membrane also allows waste and other products to leave the cell.*This process of maintaining the cell's environment is called homeostasis.**How does the plasma membrane maintain homeostasis?**

Self - Explanation: This sentence is asking me how does it maintain homeostasis. It tells me how in two sentences above that. It maintains it by allowing to waste and other things to get out of the cell.

Nice job! +110

Drag correct blocks here!

Total Points: 660

Points this Round: 660

+110

Click this button to pause → Wait! Let's take a break! ← Click this button to pause

Figure 3. Screenshot of iSTART-ME mini-game Bridge Builder

1.3 Current Study

Previous work has provided insight into the way that individual differences influence how students regulate their behaviors. However, there remain questions regarding the influence of these individual differences on students' behavior patterns and learning outcomes. The current study attempts to address this issue by examining how students' behavior patterns within the game-based environment iSTART-ME relate to system performance and vary as a function of individual differences. We investigated two primary questions:

1) Do students' behavior patterns influence their daily self-explanation quality?
2) Do individual differences influence students' patterns of interactions within the system?

2. METHODS

2.1 Participants

Participants in the current study (n= 40) were high school students from the Midwest United States. The students were, on average, 15.9 years of age, with a mean reported grade level of 10.4. Of the 40 students, 50% were male, 17% were Caucasian, 73% were African-American, and 10% reported other ethnicities.

2.2 Procedure

The current work is part of a larger study conducted to compare iSTART-ME, iSTART, and a non-tutor control [18]. The current study solely focuses on the 40 subjects assigned to the iSTART-ME condition, as they had access to the full game-based environment. The study consisted of 11 sessions. Session 1 was a pretest wherein the students answered a battery of questions, including measures of prior ability, commitment to learning, and attitudes toward technology. During sessions 2 through 9, students interacted with the game-based system. Session 10 comprised the posttest portion of the experiment, including measures similar to those in the pretest. Finally, one week after the posttest, students returned for session 11. During this session, students completed a retention test that included similar measures as the pretest and posttest.

2.3 Measures

2.3.1 Pretest reading comprehension

Students' reading comprehension ability was assessed using the Gates-MacGinitie Reading Test [23]. This test is a well-established measure of student reading comprehension (α=.85-.92 [24]). This task consists of 48 questions that ask students to read a passage and then answer two to six comprehension questions about the material in that passage.

2.3.2 Strategy performance

Students' self-explanation ability was assessed at pretest and during training. At pretest, students were asked to read a short science passage and self-explain predetermined target sentences in that text. During training, students' self-explanation ability was assessed through their interactions with the generative practice games. In these games, students were shown science texts and asked to generate their own self-explanations for various target sentences within the texts. All self-explanations were scored using the previously mentioned iSTART algorithm.

2.3.3 System Interaction Choices

Students' recorded interactions with iSTART-ME involved one of four types of game-based features, each representing a different type of game-based functionality within iSTART-ME. Each interaction was classified as involving one of the four categories of game-based functionalities (see Table 1 for descriptions).

Table 1. Interaction Categories within iSTART-ME

Interaction Classification	Description
Generative Practice	Students generate their own self-explanation
Identification Mini-Games	Students identify the self-explanation strategy
Personalizable Features	Students customize some aspect of the system interface
Achievement Screens	Students view their performance within iSTART-ME

2.3.4 Commitment to Learning

Students' commitment to learning was assessed at pretest through two self-report questions. A composite score was calculated that combined the questions related to students' enjoyment of learning and their frequency of reading for enjoyment (see Table 2 for questions).

Table 2. Learning Commitment Questions

Response Statement	Scale*
"How much do you enjoy reading?"	1 - 6
"How much do you enjoy learning about non-scientific material?"	1 - 6
*1 (Strongly Dislike) to 6 (Strongly Like)	

2.3.5 Prior Expectations of Technology

Students' prior expectations of technology were assessed at pretest. This measure was a composite score that combined two self-report measures related to students' expectations of computer helpfulness and their expected enjoyment while interacting with the iSTART-ME system (see Table 3 for questions).

Table 3. Prior Expectations of Technology Questions

Response Statement	Scale*
"Do you expect to enjoy interacting with this system?"	1 - 6
"Do you expect computers to be helpful?"	1 - 6
*1 (Strongly Disagree) to 6 (Strongly Agree)	

2.4 Dynamical Methodologies

Students' interaction patterns were classified using two dynamical methodologies. First, students' sequence of interaction patterns were analyzed using a random walk model. This method has been used in previous work to analyze fluctuations in patterns across time [16, 25]. Random walks create a spatial representation of categorical sequences across time. In the current study, we generated a unique walk for each student by first placing an imaginary particle at intersection of the x and y-axes (0,0). Then using system log-data we examined the patterns of interactions in which students engaged and moved the particle in a manner consistent with a simple set of rules (see Table 4). These rules dictated what direction the particle would "step." For instance, if students played an identification mini-game the particle moved one "step" up along the y-axis. If students chose to play a generative practice game, the particle moved one "step" left along the x-axis. When students chose to interact with an achievement screen the particle moved one "step" down along the y-axis. Finally, when students chose to interact with personalizable feature, the particle moved one "step" right along the x-axis. Notably, the direction of movement is arbitrary (i.e., a certain direction is not associated with the quality of the feature). Figure 4 reveals what a completed walk from the current study looked like for a student with 326 interaction choices.

Table 4. Particle movement assignment

Students' Choice of Interaction	Direction of Movement
Generative Practice Games	1 step left along the X-axis
Identification Mini-Games	1 step up along the Y-axis
Personalizable Features	1 step right along the X-Axis
Achievement Screens	1 step down along the Y-axis

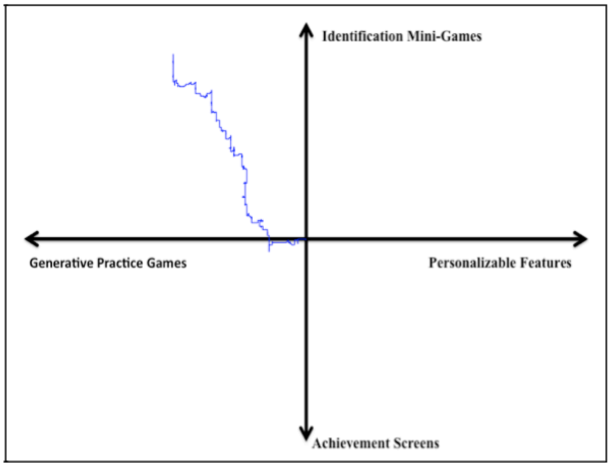


Figure 4. Complete Random Walk

Using students' sequence of categorical choices, we calculated Euclidian distances for each step within their random walk (see Equation 1). The combination of all distance calculations within students' random walk generated a "distance time series." which was representative of the fluctuations in students' interaction patterns across time. These distance time series calculated how far students' choice patterns fluctuated from the origin (0,0). Finally, the classification of each student's interaction pattern was conducted by using the distances time series generated from the random walk analysis and entering them into a detrend fluctuation analyses (DFA). The result of each DFA was a scaling component called the Hurst exponent [26]. The Hurst exponent can classify the tendency of long-term time series as follows: $0.5 < H \leq 1$ indicates persistent (deterministic or controlled) behavior, $H = 0.5$ signifies random (independent) behavior and $0 \leq H < 0.5$ denotes antipersistent (corrective) behavior. Patterns that are classified as persistent are considered to be equivalent to a positive correlation. Time series exhibiting persistence are thought to reflect self-organized and controlled processes [27]. Conversely, when patterns are classified as antipersistent, the pattern is said to be equivalent to negative correlations. This measure has been used in a variety of domains to view fluctuations and the persistence of complex patterns across time [26].

$$\text{Distance} = \sqrt{(y_i - y_0)^2 + (x_i - x_0)^2} \qquad (1)$$

3. RESULTS

3.1 Hurst Exponents

Hurst exponents were used to quantify students' patterns of choices within the iSTART-ME system. In the current study students' Hurst exponents varied considerably from weakly to strongly persistent (range =0.57 to 1.00, M=0.77, SD=0.11).

3.2 Hurst Exponents and System Interactions

Within the current study, students varied in the interaction patterns (Hurst exponents). To provide a visualization of what variability in Hurst scores looks like within the system two probability analyses were conducted. These probability analyses are similar to the ones used by D'Mello and colleagues (2007). This calculation can be described as L[It→Xt+1]. Simply put, we are examining the probability of a student's next interaction (X) with an interface feature given their previous interaction (I). For the current study, we calculated two of these probability analyses. One for a student with a high Hurst score (i.e., deterministic pattern) and one for a student with a low Hurst score (i.e., weakly persistent pattern).

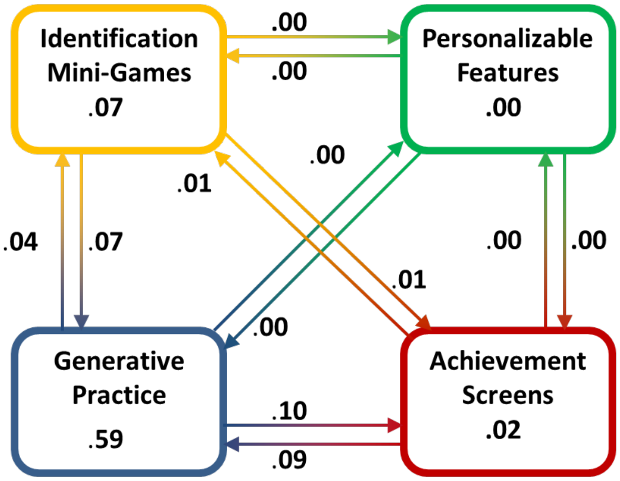


Figure 5. Transitional Probability of a Student with a High Hurst Score

Figure 5 illustrates how a student with a Hurst exponent of .98 interacted with various features in the iSTART-ME system. This student interacted with the generative practice games almost 60% of the time, revealing very little tendency to interact with other features in the system. When this student did engage with another game-based feature, there was a tendency to transition back to the generative practice games afterwards. Thus, this student seemed to be acting in a decisive manner, consistently interacting with generative practice games or transitioning back to generative practice games after engaging with another feature.

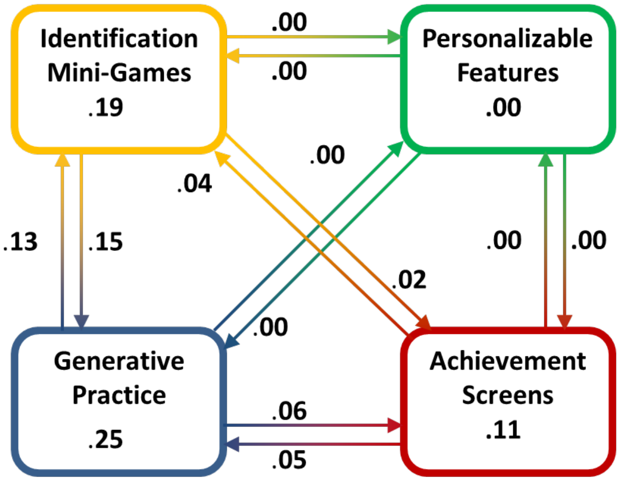


Figure 6. Transitional Probability of a Student with a High Hurst Score

Conversely, Figure 6 illustrates how a student with a Hurst exponent of .60 interacted within the system. This analysis revealed that the student with a low Hurst score explored more of the system interface than the student with a Hurst score of .98 (see Figure 5). However, the interaction pattern was more spread out and less predictable compared to the student with the high Hurst score. Thus, this student was not acting in a decisive manner and as such, may not have been regulating their learning experience within the iSTART-ME system.

3.3 Hurst and Self-Explanation Quality

The current study examined how variations in students' interaction patterns within a game-based environment related to their daily strategy performance. Pearson correlations were conducted (see Table 5) to investigate relations between students' interaction patterns and their daily self-explanation scores. Results from this analysis indicated that there was a positive correlation between students' overall Hurst exponents (regulatory measure) and self-explanation quality on days 1, 2, 3, 4, and 6. Hurst exponents were also marginally related to students' self-explanation quality on days 5 and 7. However, there was no significant relation between Hurst exponents and self-explanation quality on day 8 of training.

To further examine these relations, we conducted separate hierarchal regression analyses on students' self-explanation quality scores for each of the eight training days. These analyses investigated how students' interaction patterns predicted self-explanation scores over and above prior self-explanation ability (i.e., self-explanation scores at pretest). This is reflected by the R^2 change attributable to the variance accounted for by the interaction patterns (i.e., Hurst exponents) after entering prior self-explanation ability in the regression model (see Table 6). These analyses revealed significant models and R^2 change for *session 2*, *F*(1,37)=5.32, *p*<.05, R^2=.21, R^{2Change}=.11 (i.e., see session 2 in Table 6), *session 3*, *F*(1,37)=5.29, *p*<.05, R^2=.29, R^{2change}=.11, *session 4*, *F*(1,37)=9.42, *p*<.01, R^2= 29, R^{2change}=.19, and *session 6*, *F*(1,37)=6.251, *p*<.05, R^2=.19, R$^{2\ change}$=.14. These analyses also reveal a marginally significant R^2 change on *session 1*, *F*(1,37)=3.08, *p*=.08, R^2= .41, R$^{2\ change}$=.05, where prior self-explanation ability accounted for the majority of the variance in performance during that initial session.

Table 5. Hurst Exponents and Daily Self-Explanation Quality

Self-Explanation Quality	Interaction Patterns (Hurst)
Session 1	.325*
Session 2	.387*
Session 3	.391*
Session 4	.477**
Session 5	.296 (M)
Session 6	.405**
Session 7	.282 (M)
Session 8	.054

p=.05*, p<.01**, p<.10 (M)

Table 6. Hierarchal Linear Regressions Predicting Self-explanation Quality from Interaction Patterns (Hurst) and Prior Self-Explanation Ability

Self-Explanation Quality	*β*	ΔR^2	R^2
Session 1			**.41****
Prior Self-Explanation Ability	.56	.36**	
Interaction Patterns	.23	.05(M)	
Session 2			**.21***
Prior Self-Explanation Ability	.26	.10(M)	
Interaction Patterns	.34	.11*	
Session 3			**.29***
Prior Self-Explanation Ability	.36	.18*	
Interaction Patterns	.32	.11*	
Session 4			**.29***
Prior Self-Explanation Ability	.25	.10(M)	
Interaction Patterns	.43	.19*	
Session 5			**.22***
Prior Self-Explanation Ability	.37	.17*	
Interaction Patterns	.23	.05	
Session 6			**.19***
Prior Self-Explanation Ability	.16	.05	
Interaction Patterns	.38	.14*	
Session 7			**.16***
Prior Self-Explanation Ability	.28	.10(M)	
Interaction Patterns	.25	.06	
Session 8			**.15**
Prior Self-Explanation Ability	.38	.14*	
Interaction Patterns	.04	.01	

p<.05 *, p<.01 **, p<.10 (M)

These findings reveal that students' interaction patterns play an important role in students' daily self-explanation performance, particularly after the first session. We further examined whether students' interaction patterns varied as a function of individual differences using pretest measures of reading ability, commitment to learning, and prior expectations of technology (see Table 7). Results from this analysis revealed that commitment to learning was the only pretest measure significantly related to students' interaction patterns. This variable accounted for 15% of the variance among students' interaction patterns as reflected by the Hurst exponent scores. Indeed, when students reported a higher commitment to learning they were more likely to interact with the system in a controlled and deterministic way. Interestingly, students' prior ability level and expectations of technology was not related to their pattern of interactions. Thus, when given agency over a learning task, students learning goals may be one of the primary factors that influence how regulated they behave. These findings support previous work that shows that students' goals are an important contributor to their ability to self-regulate during learning [29].

Table 7. Correlations between Interaction Patterns (Hurst) and Individual Differences

Variable	Interaction Patterns
Reading Ability	.150
Commitment to Learning	.387*
Prior Expectations of Computers	.281(*M*)

*$p < .05$, *M*=Marginal

4. DISCUSSION

The current study investigated how students' behaviors within an adaptive environment impacted their daily learning outcomes and varied as a function of individual differences. The current study utilized a scaling component (i.e., Hurst exponents) to classify students' interactions with game-based features as random or deterministic (i.e., controlled). Previous work has posited that an important aspect of self-regulation is a student's ability to control behaviors and act in a decisive manner [1, 7 – 8]. Thus, patterns that manifest within students' behaviors may reveal one component of self-regulated learning.

The analysis presented here is a potential means of covertly capturing one aspect of self-regulation (i.e., self-control). Students with higher Hurst exponents are said to be engaging in deterministic and controlled behavior patterns. Students with lower Hurst exponents are described as engaging in random behaviors. Random behaviors are associated with less purpose, control, or persistence. Results presented here indicate that these tendencies across time are related to students' daily learning outcomes. When students engaged in controlled behaviors, they were more likely to generate higher quality self-explanations across training. When this analysis was taken a step further, it was revealed that this relation held for the majority of training days when factoring out prior ability in self-explanation.

Although it is important to understand the impact that controlled interaction patterns have on daily learning outcomes, it is also important to identify students who are more inclined to engage in controlled patterns. Understanding how individual differences drive students' patterns of choices within adaptive environments has the potential to contribute to a deeper understanding of self-regulation. Thus, the current study investigated how individual differences in students' reading ability, prior expectations of computers, and commitment to learning was related to their propensity to interact with game-based features in a deterministic manner. The current findings indicated that only students' commitment to learning was positively related to controlled patterns of interactions within iSTART-ME. Hence, when students expressed a desire to learn, they were also likely to act in a decisive, persistent, controlled, and deterministic manner in the system. Self-regulation researchers have postulated that when students are motivated to achieve learning goals they are more likely to regulate their behaviors [30]. These findings support previous research, which reveals that self-regulation is related to students' learning goals [29]. Thus, students' ability to control their behaviors is not necessarily tied to their literacy skills or familiarity with computers. Indeed, students must choose to take an active role in their learning and behave in a manner that supports their learning goals. These findings are preliminary. Clearly, future research will call for better measures of learning orientation to gain a deeper understanding of how students' attitudes influence the nuanced ways in which they approach learning tasks within game-based environments. Nonetheless, these results contribute to theoretical notions of self-regulation by revealing potential relations between students' attitudes and patterns of controlled behaviors.

In sum, these exploratory findings are promising for educational researchers as they reveal how students' behavior patterns influence learning outcomes. The current work also begins to shed light upon the nuanced ways in which scientists may be able to trace and classify students' interactions within adaptive systems. These analyses provide evidence suggesting that dynamical methodologies may afford researchers an online stealth assessment of self-regulation. Future work calls for confirmatory studies focused on demonstrating concurrent validity as well as how these dynamical methods of analysis can be utilized to improve student models within adaptive systems. Namely, real time analysis may offer a useful means of measuring self-regulated behavior patterns without relying on self-report questionnaires. If student models are able recognize optimal vs. non-optimal patterns of interaction for each student, we expect that learning systems will more effectively adapt to students' needs based on students' behavior patterns.

5. REFERENCES

[1] McCombs, B. L. 1989. Self-regulated learning and academic achievement: A phenomenological view. In B. J. Zimmerman & D. H. Schunk (Eds.), *Self-regulated learning and academic achievement: Theory, research, and practice* (pp. 51-82). Springer New York: Springer-Verlag.

[2] Pintrich, P. R., and De Groot, E. V. 1990. Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology, 82*, (1990) 33-40.

[3] Winne, P. H., and Nesbit, J. C. 1995. Graph theoretic techniques for examining patterns and strategies in learners' studying: An application of LogMill. Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, CA (April, 1995).

[4] Winne, P., and Hadwin, A. 1998. Studying as self-regulated learning. In Hacker, D., Dunlosky, J., and Graesser, A. (Eds.), *Metacognition in Educational Theory and Practice,* (pp. 279-306). Hillsdale, NJ: Erlbaum.

[5] Zimmerman, B. J. 1990. Self-regulated learning and academic achievement: An overview. *Educational psychologist, 25*, (1990), 3-17.

[6] Zimmerman, B. J., and Schunk, D. H. 1989. *Self-regulated learning and academic achievement: Theory, research, and practice*. New York: Springer-Verlag Publishing.

[7] Schunk, D. H. 2008. Metacognition, self-regulation, and self-regulated learning: Research recommendations. *Educational psychology review, 20* (2008), 463-467.

[8] Schunk, D. H., and Zimmerman, B. J. 2003. *Self*‐*regulation and learning. Handbook of psychology.*

[9] Hadwin, A. F., Nesbit, J. C., Jamieson-Noel, D., Code, J., and Winne, P. H. 2007. Examining trace data to explore self-regulated learning. *Metacognition and Learning, 2*, (2007), 107-124.

[10] Sabourin, J., Rowe, J., Mott, B., and Lester, J. 2011. When off-task is on-task: the affective role of off-task behavior in narrative-centered learning environments. *Proceedings of Artificial Intelligence in Education* (Auckland, New Zealand June 28 –July 1, 2011). Springer Berlin/Heidelberg, 534-536.

[11] Ellis, D., and Zimmerman, B. 2001. Enhancing self-monitoring during self-regulated learning of speech. In H. Hartman (Ed.), *Metacognition in Learning and Instruction* (Vol. 19). Kluwer, Dordrecht, The Netherlands, 205-228.

[12] Riding, R., and Rayner, S. 1998. *Cognitive styles and learning strategies: Understanding style differences in learning and behavior*. David Fulton Publishers, London.

[13] Satterly, D. J., and Brimer, M. A. 1971. Cognitive styles and school learning. *British Journal of Educational Psychology, 41*, (1971) 294-303.

[14] Witkin, H. A., Moore, C. A., Goodenough, D. R., and Cox, P. W. 1977. Field-dependent and field-independent cognitive styles and their educational implications. *Review of educational research, 47*, (1977)1-64.

[15] Snow, E. L., Jackson, G. T., Varner, L. K., and McNamara, D. S. 2013. Expectations of technology: A factor to consider in game-based learning environments. In K. Yacef et al. (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence in Education* (Memphis, Tennessee, July 9 -12, 2013), Heidelberg, Berlin: Springer. 359-368.

[16] Snow, E. L., Likens, A., Jackson, G. T., and McNamara, D. S. 2013. Students' walk through tutoring: Using a random walk analysis to profile students. In S. K. D'Mello, R. A. Calvo, and A. Olney (Eds*.), Proceedings of the 6th International Conference on Educational Data Mining,* (Memphis, Tennessee, July 6 -9, 2013), Springer Berlin Heidelberg, 276-279.

[17] Jackson, G. T., Varner, L. K., Boonthum-Denecke, C., & McNamara, D. S. in press. The impact of individual differences on learning with an educational game and a traditional ITS. *International Journal of Learning Technology*.

[18] Jackson, G. T., and McNamara, D. S. 2013. Motivation and Performance in a game-based intelligent tutoring system. *Journal of Educational Psychology, 105*, (2013), 1036-1049.

[19] McNamara, D. S., O'Reilly, T., Rowe, M., Boonthum, C., and Levinstein, I. B. 2007. iSTART: A web-based tutor that teaches self-explanation and metacognitive reading strategies. *In Reading comprehension strategies: Theories, interventions, and technologies,* D. S. McNamara, Ed. Erlbaum. Mahwah, NJ, 397-420.

[20] Jackson, G. T., Boonthum, C., and McNamara, D. S. 2009. iSTART-ME: Situating extended learning within a game-based environment. *In Proceedings of the Workshop on Intelligent Educational Games at the 14th Annual Conference on Artificial Intelligence in Education*, (Brighton, UK, July 06 -10, 2009). AIED, IOS Press, 59-68.

[21] McNamara, D.S., Boonthum, C., Levinstein, I.B., and Millis, K. 2007. Evaluating self-explanations in iSTART: Comparing word-based and LSA algorithms. *In Handbook of Latent Semantic Analysis,* T. Landauer, D. S. McNamara, S. Dennis, and W. Kintsch, Eds. Erlbaum. Mahwah, NJ, 227-241.

[22] Jackson, G. T., Dempsey K. B., and McNamara, D. S. 2010. The evolution of an automated reading strategy tutor: From classroom to a game-enhanced automated system. In M.S. Khine & I.M. Saleh (Eds.), *New Science of learning: Cognition, computers and collaboration in education* (pp. 283-306). New York, NY:Springer.

[23] MacGinitie, W., and MacGinitie, R. 1989.*Gates MacGinitie reading tests.* Riverside. Chicago, IL.

[24] Phillips, L. M., Norris, S. P., Osmond, W. C., and Maynard, A. M. 2002. Relative reading achievement: A longitudinal study of 187 children from first through sixth grades. *Journal of Educational Psychology, 94,* (2002) 3-13.

[25] Lobry, J. R. 1996. Asymmetric substitution patterns in the two DNA strands of bacteria. *Molecular Biological Evolution, 13*, (1996) 660–665.

[26] Hurst, H. E. 1951. Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers., 116,* (1951) 770-808.

[27] Van Orden, G. C., Holden, J. G., and Turvey, M. T. 2003. Self-organization of cognitive performance. Journal of *Experimental Psychology: General, 132*, (2003) 331-350.

[28] D'Mello, S. K., Taylor, R., and Graesser, A. C. 2007. Monitoring affective trajectories during complex learning. In D. S. McNamara & J. G. Trafton (Eds.), *Proceedings of the 29th Annual Meeting of the Cognitive Science Society* (Nashville, Tennessee, August 1-4, 2007) Cognitive Science Society, 203-208.

[29] Pintrich, P. R. 2000. An achievement goal theory perspective on issues in motivation terminology, theory, and research. *Contemporary Educational Psychology,* 25, (2000), 92-104.

[30] Bandura, A. 1991. Social cognitive theory of self-regulation. *Organizational behavior and human decision processes, 50*, (1991), 248-287.

Short Papers

Acquisition of Triples of Knowledge from Lecture Notes: A Natural Language Processing Approach

Thushari Atapattu, Katrina Falkner, Nickolas Falkner
School of Computer Science
University of Adelaide, Australia
(+61)883136178
{thushari.atapattu, katrina.falkner, nickolas.falkner}@adelaide.edu.au

ABSTRACT

Automated acquisition of knowledge from text has been utilised across several research areas including domain modeling of knowledge-based systems and semantic web. Primarily, knowledge is decomposed as fragments in the form of entities and relations called triples (or triplets). Although empirical studies have already been developed to extract entities (or concepts), relation extraction is still considered as a challenging task and hence, performed semi-automatically or manually in educational applications such as Intelligent Tutoring Systems. This paper presents Natural Language Processing (NLP) techniques to identify subject-verb-object (SVO) in lecture notes, supporting the creation of concept-relation-concept triple for visualisation in concept map activities. Domain experts have already been invested in producing legible slides. However, automated knowledge acquisition is challenging due to potential issues such as the use of sentence fragments, ambiguity and confusing use of idioms. Our work integrates the naturally-structured layout of presentation environments to solve semantically, syntactically missing or ambiguous elements. We evaluate our approach using a corpus of Computer Science lecture notes and discuss further uses of our technique in the educational context.

Keywords

Triples, lecture notes, relation extraction, NLP, concept map.

1. INTRODUCTION

Automated annotation of unstructured text, which is decomposed as entities and relations, is beneficial for wide variety of applications. Among them, within the educational context, knowledge-based systems such as intelligent tutoring systems benefit from semi- or fully automated domain modeling. Concept map activities such as *skeleton* maps to fill missing nodes or links benefit from adopting concept map mining (CMM) techniques as a way of reducing manual workload.

Although, previous studies focused on entity extraction [1], relation extraction is still challenging, with many techniques adopting pre-defined relations or 'named entities' (e.g. location) [2] and hence, restricted to specific domains. Although supervised learning approaches are more efficient, majority of such algorithms inapplicable to extract undefined relations. Technical disciplines like Computer Science lack named entities or pre-defined patterns and hence, not possible to reuse existing works.

This paper discusses a tool developed to automatically extract triples from lecture notes. Concept map extraction from text books is covered in other works [3]. Domain experts have already been invested in producing legible slides, allowing their expended effort to be applied to more activities that are beneficial for both the teacher and the learners. However, using NLP techniques to extract knowledge is challenging due to the noisiness of the data including use of sentence fragments, idioms and ambiguity. Therefore, we utilise contextual features such as the natural layout of presentation framework to resolve syntactically and semantically missing or ambiguous elements. This includes allocating missing subject or objects of fragments, resolving pronouns using a novel algorithm. Unlike other works which incorporate triple extraction from well-written sentences [5-6] or text books [3], to our knowledge, there are no studies until this which have implemented a full scale triple extraction from ill-written text in educational materials.

Two human experts having knowledge in Computer Science and linguistics were recruited to participate the experiments; 1. pronoun resolution 2. triple annotation. The comparison between machine and human extraction and the agreement between human experts is presented using *accuracy (F-measure)* and the *positive specific agreement* [7] respectively. We hypothesise that our proposed system is effective if *human-to-machine agreement is greater than or equal to human-to-human agreement* [8].

2. RELATED WORK

Triple extraction from Biology text books has been studied in a previous work [3] which presented a drawback of their failure to extract every triple from every sentence which leads to poor coverage of number of triples and therefore, poor pedagogical value. Triple extraction using heuristics [5] compares 3 popular parsers: Stanford/OpenNLP, link parser and Minipar. We reuse their work; however, heuristics proposed are restricted to unambiguous, complete sentences. Authors in [6] extracts all possible ordered combinations of three tokens (i.e. triple candidates) to train SVM using human annotated triples. This work has a limitation of considering all combinations of three tokens which exponentially increase with the length of sentences.

3. CONCEPT MAP MINING

Our core research focus is on automatically extracting concept maps from lecture notes to provide variety of assessment/reflective activities for learners. Initially, noise is automatically reduced using co-occurrence analysis techniques. NLP-based algorithms developed to extract concepts and rank them using structural features such as number of incoming and outgoing links, proximity and typography factors. Finally, the system produces a CXL (Concept map extensible language) file to visualise concept maps using IHMC cmap tools (http://cmap.ihmc.us/). These techniques are broadly discussed in our previous works [1,4]. The nature of presentation framework encourages incomplete, ambiguous sentences and hence, increases the difficulty of the automated knowledge acquisition. Section 4 discusses the contextual features to solve the probable issues.

4. CONTEXTUAL FEATURES

The 'word window model' is a valid approach to solve word sense disambiguation [9]. It considers a window of *n* words to the left and right of the 'ambiguous term' to determine the context of the target word. The window can be several words in same sentence, several sentences in paragraph, or a document. By applying this method to our problem, we utilise contextual information embedded in slides to resolve ambiguity. To support our claim, we assume that the slide heading reflects the content in that particular slide. Further, we assume that each bullet-point shares logical relations with its sub points. However, there is no guarantee that an existence of logical relation between preceding and succeeding sentences in same indentation levels.

1. Syntactic rules (subject or object allocation)

This section proposes an approach to nominate syntactically missing elements in fragments. There are two main types of fragments in English called noun phrases (NP) and verb phrases (VP). Noun phrases contain a noun(s) followed by a verb. Therefore, noun phrases require an 'object' to create subject-verb-object triples. Similarly, verb phrases contain a verb followed by a noun(s) which requires 'subject' to form a complete sentence. More information on the grammatical meanings of tags can be found in http://bulba.sdsu.edu/jeanette/thesis/PennTags.html.

1. NP which contains the pattern [NP VP[VB]] look forward for candidate nouns in 'child' (i.e. sub-indentation) levels to allocate missing 'objects'.
2. VP which contains the pattern [VP[VB NP]] look backward for nouns in 'parent' (i.e. preceding-indentation) levels to allocate missing 'subjects'.

Weights are assigned to candidate nouns based on features such as grammatical structure (e.g. nouns, verbs), distance from 'input fragment' to candidate, number of tokens in the candidate phrase, whether it is an immediate level (backward or forward) or not. The weight calculation finds the subject or object to transform fragments into complete sentences.

2. Semantic rules

Lecture notes consist of semantic ambiguities such as pronouns. The widely used approach for pronoun resolution is utilising 'named entities'. Other works include searching replacement candidates in the same sentence or backward and forward search of preceding and succeeding sentences [11]. Since lecture notes lack logical relations between preceding and succeeding sentences, we propose a new algorithm.

Pronoun resolution

We applied a mechanism proposed in [11] to find replacements when bullet-point contains multiple sentences. Additionally, we find replacements in 'parent-levels', which is the preceding indentation level or heading. We assign weights for each candidate according to features such as 'location' of the candidate, distance from the pronoun, grammatical structure, grammatical number (singular or plural). The most suitable candidate is chosen using weights.

Demonstrative determiners

Lecture notes often contain demonstrative determiners (e.g. *this, these*), a word or phrase that occurs together with a word(s) to express the reference of that word(s) in the context. Our proposed approach to resolve them only considers lexical reiterations (e.g. *these calls-> system calls*). We consider features like grammatical number (singular or plural), number of strings overlaps with the candidate, grammatical structure and the determiner.

5. TRIPLE EXTRACTION

We propose a new set of features to extract entity-relation triples from English sentences. Our feature set is applicable regardless of the pre-defined patterns as in [5]. However, reusing their work might improve the accuracy in specific sentence patterns [5-6]. Our addition of new features is a consequence of broad analysis of approximately 140 lecture slide sets from different courses. This work has the potential for reuse for any knowledge source by eliminating features specific to the presentation framework.

The NLP annotation includes parsing the sentence through Stanford statistical parser [10] and link grammar parser [12]. In order to assist better understanding of the features, we derive a decision tree (Figure 1).

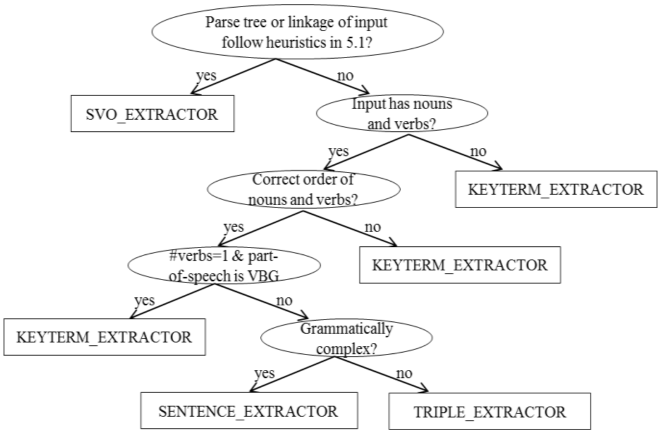

Parse tree or linkage of input follow heuristics in 5.1?
yes
no
SVO_EXTRACTOR
Input has nouns and verbs?
yes
no
Correct order of nouns and verbs?
KEYTERM_EXTRACTOR
yes
no
#verbs=1 & part-of-speech is VBG
KEYTERM_EXTRACTOR
yes
no
KEYTERM_EXTRACTOR
Grammatically complex?
yes
no
SENTENCE_EXTRACTOR
TRIPLE_EXTRACTOR

Figure 1. Decision tree which describes features and actions

1. Linguistic-based heuristics

Syntactic parse tree

Figure 2 illustrates a parse tree based on Stanford parser [10].

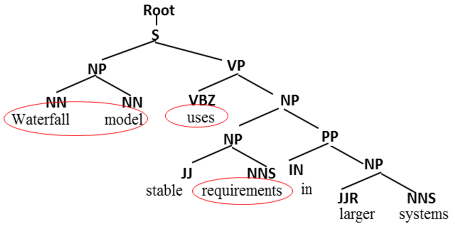


Figure 2. Parse tree of an example sentence

The following heuristics are based on previous work [5]. If the sentence contain the pattern 'Root (S (NP_subtree) (VP_subtree)), it applies following rules to extract SVO triples.

Rule 1 (subject): Perform breadth first search in NP_sub tree and select first descendant of NP_sub tree

Rule 2 (verb): Search in VP_sub tree for deepest verb descendent

Rule 3 (object): Search in PP, NP or ADJP siblings of the VP_sub tree. In NP or PP_sub tree, select first noun or compund noun, or in ADJP sub tree, select first adjective

We extended these heuristics to extract prepositional phrases.

Linkage

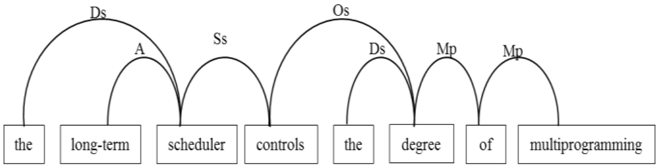


Figure 3. Linkage diagram of an example sentence

Figure 3 illustrates the linkage diagram obtained from link grammar parser [12]. Following heuristics are based on [5].

Rule 1(subject): Selects the word left of S_link

Rule 2(verb): Select first word right of S_link until {Pv, Pg, PP, I, TO, MVi} links found

Rule 3(object): Select links from 'verb' until {O, Os, Op, MVpn} links found

According to the decision tree, if input sentence follows one of the rules above, we simply extract SVO from them. However, there are many variations of sentence patterns found in lecture notes which arise us to exploit new features.

2. Sentence-based features

We extract the part-of-speech of all sentences filtered out from the criteria above. Our previous work implemented a greedy approach to identify nouns, compound nouns with their adjectives and verbs [1]. These extractions are checked against 'order' where a verb should be in-between two noun(s) to form an entity-relation triple. The candidate list should contain at least one none gerund verb. Computer Science domain contains verbs in its *–ing* form (called *gerund* -VBG), which can be used as nouns (e.g. *Software testing*). All the sentences exclude from above criteria might not produce triples and hence, important key terms are extracted from them (KEYTERM_EXTRACTION) [1].

The grammatical complexity is checked in remaining sentences using number of nested sentences (S) and dependent clauses (SBAR). If the sentence is identified as 'complex', but complete, Stanford typed dependency parser [10] splits them into simple sentences (SENTENCE_EXTRACTOR) and repeats all steps in the decision tree. The filtered out sentences consider fallback features (TRIPLE_EXTRACTION) such as number of nouns, number of verbs, numerals and symbols, negative verbs, subject-object distance, subject-verb distance, verb-object distance and headword of the sentence. We determine whether any candidate nouns are emphasised using different font colors, underline. This expresses the importance of terms to be selected as triple candidates. However, this feature is specific to the presentation framework. Finally, the extracted triples are checked against 'redundancy cycles' where the subject is repeated in an object.

6. EVALUATION

We selected lecture slide sets from recommended text books (e.g. *Software Engineering* by Sommerville) and Computer Science courses taught across different undergraduate levels in our University. We demonstrate our work using Microsoft PowerPoint, but our tool is applicable to other formats such as OpenOffice and Keynote with a structured template for header and text. Each selected lecture slide sets contains combination of contents such as text, programming, figures and notations.

Experiment 1 – Pronoun resolution

We observed that pronouns under study include *you, we, us, itself*, addressing students who refer to the course material. Due to lack of replacements in the context, we exclude these pronouns.

Table 1. Statistics of pronouns discovered in our corpus

Pronoun	they	their	it(s)	itself	we	them	you(r)	us
Frequency	57	51	241	17	23	34	94	22

Two human experts were recruited to nominate replacement candidate within a context of the slide. We did not provide replacements proposed by the system since it can influence the human judgment. We compare both of their pronoun resolution with machine's prediction and results are averaged. In table 2, *accuracy* (F-measure) is calculated as the harmonic mean between precision and recall and the *agreement* between participants is calculated using positive specific agreement [7].

Table 2. Accuracy and agreement of pronoun resolution

Lecture	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Frequency	17	16	0	67	54	39	44	50
Accuracy	0.857	0.66	-	0.746	0.923	0.587	0.5	0.571
Agreement	0.8	0.33	-	0.916	0.9	0.727	0.8	0.68

Lecture	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Frequency	5	10	7	7	40	23	4
Accuracy	0.5	0.909	0.19	0.41	0.528	0.857	0.66
Agreement	0.6	1	0.142	0.571	0.384	1	0.5

Table 2 verifies that the use of pronouns in courses vary depends on authors (e.g. L3=0). It is evident that courses which demonstrate grammatically rich, consistent writing styles provides probable replacement candidates, allowing computer algorithm to accurately (accuracy>0.8) resolve pronouns (e.g. L1- *software architecture*). As highlighted in the table 2, in some courses, accuracy is greater than human agreement. This validates our original hypothesis. We observed that the agreement is dropped when one rater suggests a replacement while other flagged it as 'null' when they find it uncertain. Occasionally, some of machine replacements did not overlap with human, reducing the accuracy as shown in L11 and L12. Dependent clauses appeared to be the main cause for this. Besides, some sentences include dummy pronouns (e.g. it is raining) which do not contain a corresponding replacement. Our results cannot be compared with other works since our corpus under study is different (i.e. lecture notes).

Experiment 2 – Triple extraction

This study uses different slide sets from experiment 1, but same Computer Science courses. We extracted 1996 sentences from 15 slide sets with approximately 40 slides per lecture note. The average number of sentence per slide is 3.3. From that, 265 sentences excluded due to 'insolvable' pronouns (highlighted in Table 1). We extracted 1838 triples from rest of the 1731 sentences. A sentence can consists of no, one or more triples.

Similar to experiment 1, two human experts participated to identify subject-verb-object triples. There is no guarantee that human annotations are identical with machine extracted triples since our algorithm mapped sentences into their base form using lemmatisation techniques. Therefore, we calculated string similarity between each subject, verb and object and obtained an average score. An example of similarity calculation between subjects is shown below and more details can be found in [6].

- Computer (sub) – *drawback of waterfall model* (tokens=4)
- Human (sub`) – *waterfall model* (tokens=2)

Sim (sub, sub`) = overlap / (# tokens in x; x= max(sub, sub`))
= 2/4 = 0.5

Verb and object similarity is calculated in the same way. The final similarity between computer and human is ranged between 0-1, stressing 1 is identical and 0 means no overlap. We measured the *precision* by comparing computer extracted triples to human and *recall* when performing the other way around and obtained the mean using F-measure (accuracy).

Table 3. Accuracy and agreement of triple extraction

Lecture	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
# Triples	107	81	24	173	207	108	145	221	82	134	74	180	130	110	62
Accuracy	0.862	0.507	1	0.605	0.872	0.397	0.88	0.944	0.787	0.833	0.465	0.319	0.497	0.858	0.672
Agreement	0.928	0.808	1	0.761	0.930	0.623	0.8804	0.975	0.829	0.792	0.66	0.645	0.72	0.844	0.76

According to Table 3, it is evident that some courses produces acceptable machine performance (accuracy>0.8) (e.g. L8-*Software engineering*). Computer networking slides (L3) from text book (source can be found in http://williamstallings.com/DCC6e.html) achieved an accuracy of 1, resulting in an ideal machine extraction. The accuracy is varying based on the richness of the content. Our algorithm is more effective (accuracy>0.8) for courses categorised as Software engineering, computer architecture, communications (see ACM classification in http://en.wikipedia.org/wiki/Outline_of_computer_science). We recognise these contents are *well-fitted* (e.g. rich grammar, complete sentences with apparent independent clauses) for CMM. Other courses with combinations of good text and notations (e.g. L15-distributed systems) are categorised as *average-fitted* (accuracy>0.5). The courses with low accuracy (<0.5) (e.g. programming languages, data structures) are classified as *ill-fitted* (More information on the classification can be found in [4])

Our results show that accuracy is greater than or equal to inter-rater agreement in some courses which validates our original hypothesis into some extent. The agreement varies when one party (computer or human) extracts modifiers while the others extracts only the exact words. The machine performance is dropped in some occasions (e.g. L6, L11 and L12) mainly due to our failure to handle negations correctly. It is practically challenging for machine to outperform human in a corpus like lecture notes since there is no well-defined structure for writing course materials.

An important aspect of studying concept maps mined from lecture notes is to facilitate students in understanding relationships between concepts allowing effective knowledge organisation which is not supported in the linear nature of lecture notes. The aim of this research is to adapt concept maps according to the learners' problem solving context. In future works, we evaluate our work by measuring students' performance in given tasks while learning through task-adapted concept maps. Besides, CMM techniques support wider concept mapping activities such as providing scaffolding aid and domain modeling of ITS.

7. CONCLUSION

This paper proposed a novel set of features to automatically extract entity-relation triple from lecture notes. While slides may have many potential issues, including incomplete, ambiguous sentences, we introduced a novel approach to resolve syntactically and semantically missing or ambiguous elements using contextual information of the slides. Our results showed that for *well-fitted* courses, machine performance is closer to human predictions (accuracy>0.8). However, our system indicates low accuracy for *ill-fitted* contents such as programming which are undesirable for CMM. The work presented in this paper is restricted to a corpus of Computer Science courses. We plan to conduct cross-disciplinary study to observe the validity of our approach.

8. REFERENCES

[1] Atapattu, T., Falkner, K. and Falkner, N. 2012. Automated extraction of semantic concepts from semi-structured data: supporting computer-based education through analysis of lecture notes. In *proceedings of the 23rd International conference on Database and Expert systems applications*.

[2] Cunningham, H. et al. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th anniversary meeting of the association for Computational Linguistics.*

[3] Olney, A. et al. 2011. Generating concept map exercises from textbooks. In *Proceedings of the 6th workshop on innovative use of NLP for building educational applications*.

[4] Atapattu, T., Falkner, K. and Falkner, N. 2014. Evaluation of concept importance in concept maps mined from lecture notes: computer vs human. In *proceedings of the 6th International conference on computer supported education.*

[5] Rusu, D. et al. 2007. Triplet extraction from sentences. In *Data mining and data warehouses*.

[6] Dali, L. et al. 2009. Triplet extraction from sentences using SVM. In *Data Mining and Data Warehouses (SiKDD).*

[7] Hripcsak, G. et al. 2005. Agreement, the F-measure, and reliability in information retrieval. In *Journal of the American medical informatics association*, 12(3), 296-298.

[8] Hearst, M. 2000. The debate on automated essay grading. In *Intelligent systems and their applications*.

[9] Ide, N. and Veronis, J. 1998. Introduction to the special issue on word sense disambiguation: the state of art. In *Computer Linguistic Journal – Special Issue on word sense disambiguation.* 24(1), 2-40

[10] Klein, D. and Manning, C. 2003. Accurate Unlexicalized parsing. In *Proceedings of the 41st meeting of the association for computational linguistics*, 423-430.

[11] Mitkov, R. 1998. Robust pronoun resolution with limited knowledge. In *Proceedings of the 17th International conference on Computational Linguistics*, 869-875.

[12] Sleator, D. et al. 1993. Parsing English with a links grammar. In *third International workshop on parsing technologies*.

Towards Assessing Students' Prior Knowledge From Tutorial Dialogues

Dan Stefanescu
University of Memphis
365 Innovation Drive
Memphis, TN, 38152
dstfnscu@memphis.edu

Vasile Rus
University of Memphis
365 Innovation Drive
Memphis, TN, 38152
vrus@memphis.edu

Arthur C. Graesser
University of Memphis
365 Innovation Drive
Memphis, TN, 38152
graesser@memphis.edu

ABSTRACT

This paper describes a study which is part of a project whose goal is to detect students' prior knowledge levels with respect to a target domain based solely on characteristics of the natural language interaction between students and a state-of-the-art conversational Intelligent Tutoring System (ITS). We report results on dialogues collected from two versions of the intelligent tutoring system DeepTutor: a micro-adaptive-only version and a fully-adaptive (micro- and macro-adaptive) version. We extracted a variety of dialogue and session interaction features including time on task, student-generated content features (e.g., vocabulary size or domain specific concept use), and pedagogy-related features (e.g., level of scaffolding measured as number of hints). We present which of these features are best predictors of pre-test scores as measured by multiple-choice questions.

Keywords

Intelligent Tutoring Systems, Knowledge Assessment, Tutoring Dialogues

1. INTRODUCTION

One-on-one tutoring is one of the most effective forms of instruction [14, 23, 26] because it opens up the opportunity of maximizing learning gains by tailoring instruction to each individual learner. A necessary step towards instruction adaptation is assessing students' knowledge such that appropriate instructional tasks (macro-adaptation) are selected and appropriate feedback is provided while students are working on a particular task (micro-adaptation or within-task adaptation). Students' knowledge state is a moving target and therefore, continuous monitoring and updating is necessary which makes the assessment task quite challenging. We focus in this paper on assessing students' knowledge at the moment when they first start interacting with an Intelligent Tutoring System (ITS), which is a special case of the large problem of assessing students' knowledge state, i.e. mental model. Students' prior knowledge with respect to a target domain is typically assessed using multiple choice pre-tests although other forms of assessment may be used.

Assessing students' prior knowledge is very important task in ITSs because it serves two purposes: enabling macro-adaptivity in ITSs [14], and, when paired with a post-test, establishing a baseline from which the student progress is gauged by computing learning gains. While the role of pre-test is important for macro-adaptivity and for measuring learning gains, a major challenge is the fact that the pre-test and the post-test usually take up time from actual learning. Pre-test may even have a tiring effect on students. Last but not least, designing good pre- and post-tests requires domain expertise and could be an expensive effort. Being able to infer students' knowledge directly from their performance would eliminate the pre-test thus saving time for more training, getting rid of the tiring effects or testing anxieties, and reducing developers' effort.

In this paper, we focus on identifying the most important dialogue features that best correlated with students' prior knowledge as measured by pre-tests consisting of multiple-choice questions. This work is part of a large effort which has two major research goals: (1) to understand to what extent we can predict students' pre-training knowledge levels from dialogue features and (2) what is the minimum length of dialogue that is sufficient for predicting students' knowledge states/levels with good accuracy.

An interesting aspect of our work is the fact that we assess students' knowledge levels from training sessions collected from two versions of the intelligent tutoring system DeepTutor: a micro-adaptive-only version and a fully-adaptive version (the fully adaptive is both macro- and micro-adaptive). Micro-adaptivity is about within-task adaptation: the capacity of the system to select appropriate feedback at every single step while the student is working on an instructional task. The fully-adaptive condition adds macro-adaptivity on top of micro-adaptivity; macro-adaptivity is about selecting and sequencing an appropriate set of tasks to each individual student based on her knowledge level. We used a macro-adaptivity method based on an Item-Response Theory approach [15]. As such, in the micro-adaptive-only version of DeepTutor, students worked on tasks following the one-size-fits-all approach, while in the fully-adaptive condition, 4 different task sequences were assigned to students based on their knowledge levels: low, medium-low, medium-high, and high. Analyzing the data from both conditions, our goal is to identify dialogue features and models based on these features that are good predictors of students' prior knowledge as measured by pre-test scores.

2. RELATED WORK

The most directly relevant work to ours is the one by Lintean and colleagues [10] who studied the problem of inferring students' prior knowledge in the context of an ITS that monitored and scaffolded students' meta-cognitive skills. They compared student-articulated prior knowledge activation (PKA) paragraphs to expert-generated paragraphs or to a taxonomy of concepts related to the target domain (i.e. human circulatory system). Students' prior knowledge levels were modeled as a set of 3 categories: low, medium, and high mental models. There are significant differences between the two approaches. First, we deal with dialogues as opposed to explicitly elicited prior knowledge paragraphs. Second, we do not have access to gold standard paragraphs or correct answers or a taxonomy of concepts that would allow us to make direct comparisons. Third, we model students' prior knowledge using their score on a multiple-choice pre-test.

Predicting students' learning and satisfaction is another area of research relevant to our work, and one of the earliest and most useful applications of Educational Data Mining [cf. 13]. Forbes-Riley and Litman [6] used the PARADISE framework [24] to develop models for predicting student learning and satisfaction [7]. They used 3 types of features: system specific, tutoring specific, and user-affect-related. They used the whole training session as a unit of analysis, which is different from our analysis, in which the units are instructional tasks, i.e. Physics problems. Also, their work was in the context of a spoken dialogue system, while ours focuses on a text/chat-based conversational ITS. In addition, they focused on user satisfaction and learning, while we are interested in identifying students' prior knowledge.

Williams and D'Mello [25] worked on predicting the quality of student answers to human tutor questions, based on dictionary-based dialogue features previously shown to be good detectors of cognitive processes [cf. 25]). To extract these features, they used LIWC (Linguistic Inquiry and Word Count) [10], a text analysis software program that calculates the degree to which people use various categories of words across a wide array of texts genres. They reported that pronouns and discrepant terms are good predictors of the conceptual quality of student responses. Some of our features are informed by their work.

Yoo and Kim [27] worked on predicting the project performance of students and student groups based on stepwise regression analysis on dialogue features in Online Q&A discussions. To extract dialogue features they made use of LIWC too, but also of Speech Acts [11], a tool for profiling user interactions in on-line discussions. They found that the degree of information provided by students and how early they start to discuss before the deadline are 2 important factors in explaining project grades. A similar research was conducted by Romero and colleagues [13], who also included (social) network related features. Their statistical analysis showed that the best predictors related to students' dialogue are the number of contributions (messages), the number of words, and the average score of the messages.

3. THE DATA

We conducted our research on log-files from experiments with DeepTutor [14], the first ITS based on the framework of Learning Progressions (LPs) [3]. DeepTutor is a conversational ITS based on constructivist theories of learning, which encourages students to self-explain solutions to complex science problems and only offers help, in the form of progressively informative hints, when needed. This type of adaptivity, within a task, is also known as *micro-adaptivity* [21]. Our approach to predict students' knowledge levels relies on the fact that each DeepTutor-student dialogue has its own characteristics, strongly influenced by student's profile.

Our work is based on data collected from students interacting with DeepTutor after-school, outside the lab, in the course of a multi-session online training experiment that took place in the fall of 2013. This was possible because DeepTutor is a fully-online conversational ITS, accessible from any device with an Internet connection. During the experiment, students took a pre-test, trained with the system for about one hour each week for a period of 3 consecutive weeks, and then took a post-test. Each training session consisted in solving a sequence of 8 physics problems with help from DeepTutor. The pre-test and post-test were taken under the strict supervision of a teacher. During the 3 training sessions, students were exposed to 3 different topics, one topic per week, in the following fixed order: force and motion (Newton's first and second laws), free-fall, vectors and motion (2-D motion). In our analysis, we included only 150 the students who finished all sessions in one sitting. They were randomly assigned to one of two *conditions* mentioned earlier: micro-adaptive-only (μA; n=70) and fully-adaptive (ϕA; n=80). In this paper, we only analyze the dialogues corresponding to the first session of training as it was closest to the pre-test. The data consists of a total of 8,191 student dialogue turns (9,256 sentences) out of which 4,587 (5,102 sentences) belong to μA condition, and 3,604 (4,154 sentences) to ϕA condition. Before feature extraction, the dialogues were preprocessed using Stanford NLP Parser [18]. The preprocessing pipeline consisted in 5 steps: tokenization, sentence splitting, part of speech (pos) tagging, lemmatization, and chunking.

4. THE FEATURES

The features we mined from dialogues was inspired by the work mentioned in section 2 of the paper, as well as by studies on automated essay scoring [16] in which the goal was to infer students' knowledge levels or skills from their essays. Also, our set of features is grounded in the learning literature as explained next.

The proposed dialogue features can be classified into 3 major categories: ***time-on-task***, ***generation***, and ***pedagogy***. In general, *time-on-task*, which reflects how much time students spend on a learning task, correlates positively with learning [20]. We measured *time-on-task* in several different ways as: total time (in minutes) or normalized total time (using the longest dialogue as the normalization factor). Additional time-related features were extracted such as the average time per turn and winsorized versions of the basic time-related features. *Generation* features are about the amount of text produced by students. Greater word production has been shown to be related to deeper levels of comprehensions [2, 22]. *Pedagogy* features refer to how much scaffolding a student receives (e.g. number of hints) during the training. Scaffolding is well documented to lead to more learning than lecturing or some other less interactive type of learning such as reading a textbook [22]. Feedback is an important part of scaffolding and therefore we extracted features regarding the type (positive, neutral, negative) and frequency of the feedback [17].

We extracted raw features as well as normalized versions of the features. In some cases, the normalized versions seem to be both more predictive and more interpretable. For instance, the number of hints could vary a lot from simpler/short problems, where the solution require less scaffolding in general even for low knowledge students, to more complex problems which would require more scaffolding as there are more steps in getting to the solution. That is, a normalized feature, such as the percentage of hints, would allow us to better compare the level of scaffolding in terms of hints across problems of varying complexity or solution length. In our case, we normalized the number of hints by the maximum number of hints a student may receive when answering vaguely or incorrectly at every single step during the dialogue. This number can be inferred from our dialogue management components.

We mined a total of 43 features from 1,200 units of dialogue which led to 43×1,200= 51,600 measurements. The unit of dialogue analysis was a single problem in a training session. Because the force-and-motion training session consisted of 8 problems, and we collected 150 sessions from 150 students, we ended up with 8×150=1,200 units. Due to space constraints, we do not provide the full list of features:

Time-on-task features: *total_time* (the time length of the dialogue in minutes), *avg_time_per_turn* (the average length of a student turn in minutes);

Generation features: *dialogue_size* (length of the student dialogue (number of words, no punctuation included)), *avg_dialogue_size_per_turn*, *#sentences* (number of sentences), *#chunks* (number of syntactic constituents), *vocSize* (vocabulary size), *content_vocSize* (vocabulary size of content words), *non_content_vocSize*, *dialogue_length_div_voc* (#words divided to vocabulary size), *%physicsTerms* (percentage of physics related words out of those used), *%longWords* (percentage of words longer than 6 characters), *posDiversity* (number of unique different pos-es divided by vocabulary size), *%puctuation* (percentage of punctuation out of all tokens), *%articles*, *%pronouns*, *%self-references*, *totalIC* (total Information Content of the dialogue: explained below), *totalIC_per_word*, *positiveness* (text positiveness computed based on SentiWordNet: explained below), *negativeness*.

Scaffolding features: *#turns* (number of student turns), *#normalized_turns*, *#c_turns* (number of student turns classified as contributions), *%pos_fb* (percentage of turns for which student received positive feedback), *%neg_fb*, *pos_div_pos+neg* (positive feedback divided by positive plus negative feedback), *vague_div_vague+pos* (neutral feedback divided by positive plus neutral feedback), *#shownHints* (number of shown hints), *#shownPrompts* (number of shown prompts), *#shownPumps* (number of shown pumps).

Next, we discuss on short the dialogue features on the Information Content and positive-negative polarity.

Information Content (IC) was used by Resnik [12] to measure the informativeness of a *concept c*, on the assumption that the lower the frequency of *c*, the higher its informativeness. Resnik made use of Princeton WordNet [5] and its hierarchical taxonomy, where each node is a concept, also called *synset*. The more general concepts are at the top of the hierarchy, while the specific ones, at the bottom. Each synset can be realized in texts by any of the specific senses of certain words (i.e. *literals*), which are considered to be part of that synset. To count the frequency of a nominal synset *s* in a reference text, Resnik sums the frequencies of the literals of *s* and those of all the synsets for which *s* is a parent in the hierarchy. Thus, the estimated probability of occurrence can be easily computed and so is the IC value for that synset.

We replicated Resnik's work on WordNet 3.0 for all pos-es. Starting from synsets, we transferred the IC values to individual words. If a word has various senses associated with different synsets, we assign to it the IC value corresponding to the most non-informative synset, so that high IC values are only associated with informative words. For a word that does not appear in WordNet, our algorithm selects a WordNet literal of the same grammatical category, so that the similarity between the two is sufficiently high according to an LSA model built on the whole Wikipedia [19]. We compute the IC of a text as the sum of the IC values for its words.

To include features on the ***Positive-Negative Polarity*** of the dialogues, we made use of an updated version [1] of *SentiWordNet* [4] in which, each WordNet synset is assigned scores representing the polarity strength on 3 dimensions: Positive, Negative and Objective. Based on SentiWordNet, we extracted two lists: one of positive and the other one of negative words along with computed scores for positivity and respectively, negativity. For each word in WordNet, we summed up the values on all 3 polarity dimensions corresponding to the synsets that contain that word. If the Positive dimension value (p) is at least twice the Negative value (n) and also p is greater or equal to the Objective value (o) or greater than a certain threshold (> 2), than that word is added to the list of positive words with a positivity value computed as p / (p+ n + o). An identical procedure is applied for finding the negative words. Dialogue positiveness (negativeness) is simply computed as the sum of the values assigned to all positive (negative) words found in the text, divided by the total number of words in that text.

5. EXPERIMENTS AND RESULTS

Our larger goal is to understand how various dialogue units, corresponding to one problem in a session, individually and as groups, relate to students' prior knowledge as measured by the pre-test, which is deemed as an accurate estimate of students' knowledge level. The group analysis would indicate after how much dialogue, corresponding to consecutive training problems, one can accurately infer students' pre-test score. We present in this paper only our initial feature analysis, due to space reasons.

5.1 Feature Analysis

We started by extracting all the above-mentioned features for the sub-dialogues corresponding to individual problems. We worked on 1,193 sub-dialogues spanning over 7,927 turns (6.64 on average), 5,441 minutes (4.56 on average), with a total length of 74,036 words (62.05 on average). The next step was to identify the features whose values best correlate with the pre-test scores. We considered both the entire pre-test (an extended version of Force Concept Inventory) [8], which can be seen as assessing students overall knowledge with respect to Newtonian Physics, but also the *pre-testFM*: the portion of the pre-test containing questions directly related to the force-and-motion training session. Table 1 shows correlations of features with the pre-test scores for μA condition.

Table 1. Correlations values with pre-test (top) and pre-testFM (bottom) for interesting features on each of the 8 problems in the μA condition.

	1	2	3	4	5	6	7	8
f1	-0.36	-0.408	-0.141	-0.176	-0.225	-0.136	-0.254	-0.181
	-0.408	-0.333	-0.162	-0.182	-0.256	-0.225	-0.25	-0.219
f2	0.344	0.262	0.242	0.202	0.213	0.23	0.321	0.236
	0.358	0.221	0.254	0.183	0.157	0.125	0.267	0.216
f3	-0.423	-0.403	-0.303	-0.295	-0.35	-0.245	-0.283	-0.225
	-0.433	-0.293	-0.268	-0.296	-0.305	-0.258	-0.29	-0.228
f4	-0.448	-0.444	-0.333	-0.308	-0.34	-0.36	-0.361	-0.276
	-0.473	-0.334	-0.305	-0.295	-0.278	-0.351	-0.331	-0.254
f5	0.458	0.368	0.193	0.36	0.254	0.208	0.311	0.264
	0.458	0.297	0.122	0.386	0.206	0.168	0.251	0.23
f6	-0.424	-0.425	-0.215	-0.291	-0.284	-0.326	-0.415	-0.326
	-0.464	-0.314	-0.248	-0.318	-0.223	-0.317	-0.393	-0.29
f7	-0.404	-0.386	-0.295	-0.352	-0.28	-0.337	-0.194	-0.2
	-0.385	-0.284	-0.225	-0.31	-0.219	-0.304	-0.158	-0.208

Table 1 shows that with some exceptions for problem 5, the time length (**f1**), the number of sentences (**f3**), the number of turns (**f4**), and the number of hints (**f6**) and prompts shown (**f7**) have negative correlations with the pre-test scores, while the average word-length of a turn (**f2**) and the percentage of turns receiving positive feedback (**f5**) have positive correlations. These outcomes confirm similar findings from previous studies [22]. Interestingly enough, the number of sentences students produce seem to be less and less correlated with the pre-test scores as the students advance through the training session.

Correlations for the ϕA condition were derived using a more complex process given that students were grouped into 4 knowledge levels based on their overall pre-test score and so, 4 different sets of problems were used. As such, we could not conflate the data across knowledge groups and therefore we studied the correlations for each set of problems separately. In this case, because of macro-adaptation, but also because the number of dialogues for each knowledge level was much smaller (80 students were grouped in 4 knowledge level groups: low (14), medium low (15), medium high (21), and high (30)), the best correlated features were somehow different. Given the space constraints, we will present these results in a future paper.

6. CONCLUSIONS AND FUTURE WORK

This paper presented our work towards predicting students' prior knowledge based on the characteristics of their dialogue while engaging in problem solving with a conversational ITS. The proposed dialogue features can be classified into three major categories: time-on-task, generation, and pedagogy. The features were analyzed throughout an entire training session using instructional tasks as the unit of analysis. Our next step would be to analyze these features across increasing subsets of instructional tasks, e.g. the first Physics problem in a session vs. first two problems vs. first three problems, in order to investigate after how many instructional tasks the features best correlate with prior knowledge. It should be noted that the more tasks into a session we consider the more likely the student model may have significantly change, due to training, compared. Furthermore, we will investigate these features for students at different prior knowledge level, e.g. low knowledge vs. high knowledge students. Finally, we plan to investigate prediction models based on the analyzed features and also to add affect-related features.

7. ACKNOWLEDGMENTS

This research was supported in part by Institute for Education Sciences under awards R305A100875. Any opinions, findings or recommendations expressed in this material are solely the authors'.

8. REFERENCES

[1] Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In LREC (Vol. 10, pp. 2200-2204).

[2] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., and Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, 25, 471–533.

[3] Corcoran, T., Mosher, F.A., and Rogat, A. (2009). *Learning progressions in science: An evidence-based approach to reform.* Consortium for Policy Research in Education Report #RR-63. Philadelphia, PA.

[4] Esuli, A., and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC* (Vol. 6, pp. 417-422).

[5] Fellbaum, C. (1998). WordNet: An electronic lexical database.

[6] Forbes-Riley, K., and Litman, D. J. (2006). Modelling user satisfaction and student learning in a spoken dialogue tutoring system with generic, tutoring, and user affect parameters. In *Proceedings of HLT Conference*.

[7] Forbes-Riley, K., Litman, D., Purandare, A., Rotaru, M., and Tetreault, J. (2007). Comparing Linguistic Features for Modeling Learning in Computer Dialogue Tutoring. In Proc. of the 13th International Conference on AIED, LA, CA.

[8] Hestenes, D., Wells, M., and Swackhamer, G. (1992). Force concept inventory. The physics teacher, 30(3), 141-158.

[9] Lintean, M., Rus, V., and Azevedo, R. (2011). Automatic Detection of Student Mental Models during Prior Knowledge Activation in MetaTutor, *International Journal of Artificial Intelligence in Education*, 21(3), 169-190.

[10] Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71.

[11] Ravi, S., and Kim, J. (2007). Profiling student interactions in threaded discussions with speech act classifiers. *Frontiers in Artificial Intelligence and Applications*, 158, 357.

[12] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007.

[13] Romero, C., López, M. I., Luna, J. M., and Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, 68.

[14] Rus, V., D'Mello, S., Hu, X., and Graesser, A. C. (2013). Recent Advances in Conversational Intelligent Tutoring Systems. AI Magazine, 34(3). *Lang. Syst.* 15, 5, 795-825.

[15] Rus, V., Stefanescu, D., Baggett, W., Niraula, N., Franceschetti, D., & Graesser, A.C. (2014). Macro-adaptation in Conversational Intelligent Tutoring Matters, *The 12th International Conference on Intelligent Tutoring Systems*, June 5-9, Honolulu, Hawaii.

[16] Shermis, M. D., and Burstein, J. (2003). Automated Essay Scoring: A Cross Disciplinary Perspective. Mahwah, NJ: Lawrence Erlbaum Associates.

[17] Shute, V. J. (2008). Focus on Formative Feedback. Review of Educational Research, 78(1), 153 -189.

[18] Socher, R., Bauer, J., Manning, C.D., and Ng, A.Y. (2013). Parsing With Compositional Vector Grammars. Proceedings of ACL 2013.

[19] Stefanescu, D., Banjade, R., and Rus, V. (2014). Latent Semantic Analysis Models on Wikipedia and TASA, LREC

[20] Taraban, R., and Rynearson, K. (1998). Computer-based comprehension research in a content area. Journal of Developmental Education, 21, 10-18.

[21] VanLehn, K. (2006). The behavior of tutoring systems. International Journal of AI in Education. 16 (3), 227-265.

[22] VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., and Rose, C.P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science, 31*, 3-62.

[23] VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems, *Educational Psychologist*, 46:4, 197-221.

[24] Walker, M. A., Litman, D. J., Kamm, C. A., and Abella, A. (1997). PARADISE: A framework for evaluating spoken dialogue agents. In *Proceedings of the 8th conference on EACL* (pp. 271-280).

[25] Williams, C., and D'Mello, S. (2010). Predicting student knowledge level from domain-independent function and content words. In *Intelligent Tutoring Systems* (pp. 62-71).

[26] Woolf, B. (2008). Building Intelligent Interactive Tutors, Student-Centered Strategies for Revolutionizing E-Learning, Elsevier & Morgan Kaufmann Publishers, 2008.

[27] Yoo, J.-B., and Kim, J. (2012). Predicting Learner's Project Performance with Dialogue Features in Online Q&A Discussions. In *Intelligent Tutoring Systems* (pp. 570-575).

Assigning Educational Videos at Appropriate Locations in Textbooks

Marios Kokkodis
NYU Stern
mkokkodi@stern.nyu.edu

Anitha Kannan
Microsoft Research
ankannan@microsoft.com

Krishnaram Kenthapadi
Microsoft Research
krisken@microsoft.com

ABSTRACT

The emergence of tablet devices, cloud computing, and abundant online multimedia content presents new opportunities to transform traditional paper-based textbooks into tablet-based electronic textbooks. Towards this goal, techniques have been proposed to automatically augment textbook sections with relevant web content such as online educational videos. However, a highly relevant video can be created at a granularity that may not mimic the organization of the textbook. We focus on the video assignment problem: *Given a candidate set of relevant educational videos for augmenting an electronic textbook, how do we assign the videos at appropriate locations in the textbook?* We propose a rigorous formulation of the video assignment problem and present an algorithm for assigning each video to the optimum subset of logical units. Our experimental evaluation using a diverse collection of educational videos relevant to multiple chapters in a textbook demonstrates the efficacy of the proposed techniques for inferring the granularity at which a relevant video should be assigned.

1. INTRODUCTION

Education literature has extensively highlighted the central role that textbooks play in delivering content knowledge to the students, improving student learning, and in helping teachers prepare lesson plans [19]. The rapid proliferation of cloud-connected electronic devices has enabled the availability of textbooks in electronic format. However, many of these e-textbooks are merely digital versions of the printed books, and hence do not make use of the rich functionalities provided by the electronic medium (and/or the cloud-connectedness). Thus, we have the opportunity to enrich the reading experience by augmenting e-textbooks with supplementary materials appropriate to the learning style of the student, be it auditory, visual or kinesthetic style [5, 6, 8, 15, 18]. In fact, studies show better content retention [17] and improved concept understanding [14] when educational multimedia content is shown along with textual material.

With the availability of abundant online video content [13], we can use retrieval algorithms [2] to narrow the video collection to a relevant subset for the textbook. Since the videos on the web are not created specifically for the textbook of interest, there are significant differences in the authoring style of a video creator versus that of a textbook author. The textbook author creates a logical hierarchy (chapter → sections → subsections, *etc.*) that is suitable for presentation of all the material that needs to be covered in the book. In contrast, the author of a video focuses only on the content to be presented in the video. This central difference makes it challenging to match videos to textbook units. While some videos may provide a high-level overview of the subject and hence may be appropriate at the granularity of the entire book, other videos may illustrate a specific concept or demonstrate an activity and hence may be appropriate at the level of a subsection or even a paragraph. Similarly, there may be videos that summarize a chapter or a section, and hence may be best placed at an intermediate granularity. For example, a video that contains material about different sections in a chapter can either be placed at the chapter beginning (if it provides an overview), or at the chapter end (if it helps to review the material in the chapter).

The focus of this paper is to recognize this mismatch and automatically determine the appropriate textbook locations for assigning the videos. More precisely [11]: *Given a textbook (or a chapter in a textbook) and a video relevant to the textbook (or the chapter), how do we identify the best subset of logical units (such as sections) that covers the material present in the video?*

We propose a rigorous formulation of the video assignment problem and present an algorithm for assigning each video to the optimum subset of logical units. As part of computing the objective function, we provide a novel representation for videos in terms of concept phrases present in the textbook, and their significance to the video. Our empirical study over a diverse collection of educational videos corresponding to multiple chapters in a textbook demonstrates the efficacy of the proposed techniques.

2. RELATED WORK

There has been considerable work on augmenting textbook sections with relevant supplementary materials mined from the web [1, 2, 3]. In [3], the focus has been on finding textual content from the web that is relevant for a section. Somewhat related is the work proposed in [20] that augments textual documents such as news stories with other textual documents such as blogs. In [1], a method was proposed to identify the focus of the section, which was then used to obtain relevant web videos. However, it is not always possible to assign a video to a single section. A video may contain content that extends across sections, as the author of the video may have chosen a logical ordering different from that of the author of the textbook. In this paper, we present a technique that, given the videos relevant to the entire chapter, identifies the minimal combination of sections that best encapsulates the material covered in the video. Towards this goal, we infer a representation for a video as a byproduct of the COMITY algorithm [2] which we adapt to obtain relevant videos.

3. CANDIDATE VIDEO SELECTION

We obtain the candidate set of videos relevant to a textbook chapter using an adaptation of COMITY algorithm [2] that was proposed in the context of augmenting textbook sections with images. We observed that when we applied this technique at the section level (§5), there was a huge redundancy in the retrieved videos across multiple sections[1]. We highlight two key observations: First, the content of the same video can be shared across multiple sections, calling for an approach such as the one proposed in this paper to identify the combination of sections that best describes the video. Second, by applying the algorithm at the chapter level, we identify a richer set of videos, by exploiting dependencies across sections.

Our adaptation of Comity is presented in Algorithm 1. A chapter in a textbook is represented as a set of concept phrases (*cphrs*), obtained as the set of phrases that map to Wikipedia article titles [7, 16], and further refined using the techniques proposed in [3]. Comity forms $\binom{n}{2}$ video search queries by combining two *cphrs* each, in order to provide more context about the chapter. Note that a *cphr* in isolation may not be representative of the text as the same text can discuss multiple concepts. At the same time, a single long query consisting of all concept phrases can lead to poor retrieval [9]. Figure 1 shows an example of how the queries are constructed from *cphrs* extracted from a textbook chapter on Biology. A relevant video for the chapter is likely to occur among the top results for many such queries. Thus, by aggregating the video result lists over all combinations of queries, we obtain the most relevant videos for the chapter.

Algorithm 1 Comity

```
Input: A textbook chapter; Number of desired video results $k$.
Output: Top $k$ video results from the web.
Obtain (up to) top $n$ concept phrases from the chapter.
Form $\binom{n}{2}$ queries consisting of two concept phrases each.
Obtain (up to) top $t$ video search results for each query.
Aggregate over $\binom{n}{2}$ video result lists, and return top $k$ videos.
```

4. APPROACH & ALGORITHMS

4.1 Representation of Textbook

Each section in a textbook represented by a set of *cphrs*, along with their *context-dependent importance* scores based on the importance of *cphrs* to the section. The computation of the score is based on the following observation: If a *cphr* is important for the context of the text, then the videos retrieved using it as *one of* the query terms will be related to each other. On the contrary, if the *cphr* is not, then the videos retrieved using it as *one of* the query terms will be very diverse and diffused. Figure 2 shows top *cphrs* associated with three most frequent videos for two *cphrs*, 'water' and 'gold foil experiment' (we describe the computation of *cphrs* in a video in §4.2). Consider the *cphr* 'water'. The intersection of the three sets of *cphrs* is only the *cphr*, 'water'. On the other hand, for the *cphr* 'gold foil experiment', the top three most frequent videos have a much larger set of common *cphrs*: {electron, Ernest Rutherford, gold foil experiment, foil, gold leaf, atom, structure, discovery, neutron, proton} (note that the intersection is computed over all the *cphrs* associated with the videos whereas only the top *cphrs* are shown). Thus, a specific phrase is likely to lead to videos that are more similar to each other than a generic phrase.

With this intuition, we measure the importance score, *I*(c) as the average pair-wise inner product between top m videos retrieved when c is used in conjunction with all other *cphrs* in the textbook.

$$I(c) = \frac{\sum_{1\leq i<j\leq m} < V_i, V_j >}{\binom{m}{2}},$$

where V_i is the vector representation (in terms of *cphrs* and associated weights) for i^{th} top video for c. We used $m = 3$ in our experiments. To account for variances in the scores due to sparsity, we also clustered the scores, and assigned the cluster means of the closest cluster to each of the *cphrs* [10].

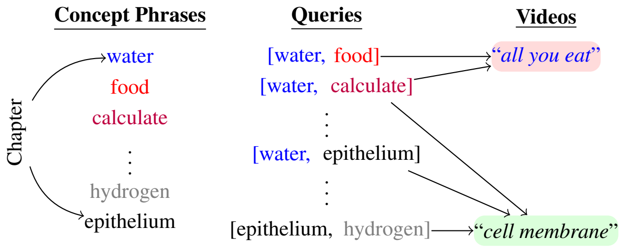


Figure 1: Query based video representation

4.2 Representation of Candidate Videos

We devise a representation for the videos motivated by the following observation: When a video is retrieved in a highly ranked position for a query, the corresponding query represents some aspects of the content of the video. As an example, consider Figure 1. The video "all you eat" describes dietary habits, and is retrieved as a top result for the queries "water, food" and "water, calculate". Thus, the *cphrs*, 'water', 'food', and 'calculate' can be associated with this video. Similarly, for the video "cell membrane", the relevant *cphrs* are 'epithelium', 'hydrogen', 'water', and 'calculate'. However, the relative importance between the *cphrs* that lead to retrieving a video varies. In this example, the video on cell membrane should be related more to epithelium than to water. Therefore, we represent a video with not only the *cphrs* that led to the video, but also their importance to the video. For each *cphr* c and video v, we define the importance $w_{v,c}$ of c to v as the fraction of queries that contain c for which video v was retrieved as a top result:

$$w_{v,c} = \frac{\{q \in Q_c | (v \in TopResults(q)\}}{|Q_c|},$$

where Q_c is the set of queries that contain *cphr* c. The intuition behind this definition is that the higher the fraction of queries that led to a specific video, the more related this phrase is with the video.

In our implementation, we restricted the possible *cphrs* that can lead to a video to be only those that are present in the textbook. However, one can extend this representation in many ways, *e.g.*, by using multiple books of the subject matter or by identifying the *cphrs* in the transcript of the video, especially when the transcript is user-uploaded.

4.3 Section Subset Selection For Videos

For a given candidate video v and a large candidate set $\mathcal{S}$ of sections from the textbook chapter, our goal is to select a *minimal subset* of top sections, $\mathcal{T} \subset \mathcal{S}$ that best covers the content in the video. We model this section subset selection problem as identifying a subset of sections $\mathcal{T}^*$ that maximizes the objective function:

$$\mathcal{T}^* = \arg\max_{\mathcal{T}\in 2^{\mathcal{S}}} \left(\text{cover}(v, \mathcal{T}) - \lambda|\mathcal{T}|\right), \tag{1}$$

where cover$(v, \mathcal{T})$ is a function that measures how well the set of sections $\mathcal{T}$ captures the content of the video v. Our objective function incorporates a penalty for using more sections than required for explaining the video, by discounting for the number of sections $|\mathcal{T}|$. Thus, the objective function provides a trade-off between the extent to which the content of the video is captured and the number of sections used. Different trade-offs can be obtained through different choices of the non-negative parameter λ: A large value of λ corresponds to a greater penalty for having more sections. We estimated the value for the size penalty parameter λ using a cross validation set. This process resulted in $\lambda = 0.48$.

[1] Similar observation was made for image retrieval [2].

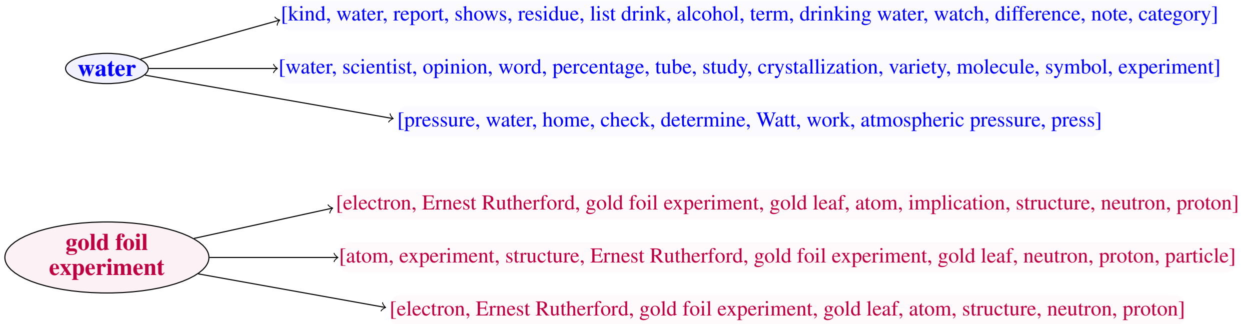

water
[kind, water, report, shows, residue, list drink, alcohol, term, drinking water, watch, difference, note, category]
[water, scientist, opinion, word, percentage, tube, study, crystallization, variety, molecule, symbol, experiment]
[pressure, water, home, check, determine, Watt, work, atmospheric pressure, press]
gold foil experiment
[electron, Ernest Rutherford, gold foil experiment, gold leaf, atom, implication, structure, neutron, proton]
[atom, experiment, structure, Ernest Rutherford, gold foil experiment, gold leaf, neutron, proton, particle]
[electron, Ernest Rutherford, gold foil experiment, gold leaf, atom, structure, neutron, proton]

Figure 2: Illustration of important ('gold foil experiment') vs non-important ('water') concept phrases

Computing cover$(v, \mathcal{T})$**:** Let $\mathcal{C}_{book}$ denote the set of all *cphrs* (concept phrases) in the book. Let $C(v) \subseteq \mathcal{C}_{book}$ denote the set of *cphrs* present in our representation of video v and let $C(\mathcal{T}) \subseteq \mathcal{C}_{book}$ denote the set of *cphrs* present in the subset of sections $\mathcal{T}$. We define cover$(v, \mathcal{T})$ to be the weighted fraction of the *cphrs* in the video that are also covered by the subset of sections:

$$cover(v, \mathcal{T}) = \frac{\sum_{c \in (C(v) \cap C(\mathcal{T}))} w_{vc} I(c)}{\sum_{c \in C(v)} w_{vc} I(c)} .$$

The cover score takes values between 0 and 1, and the higher the value, the more video content is contained in the corresponding subset of sections.

Brute-force optimization: Given the set of sections in a textbook chapter and a candidate video as inputs, our algorithm first checks whether a certain minimum fraction, θ of the video content can be covered by including all sections in the chapter, and if so, returns the optimal subset of sections (by exhaustively searching over all possible subsets). Upon performing sensitivity analysis, we observed that the algorithm is not sensitive to θ in the range $[0.6, 0.9]$, and hence we set $\theta = 0.8$ in our experiments.

Greedy optimization: In [10], we show that our objective function (Eq. 1) exhibits submodularity and hence admits an efficient greedy algorithm with provable quality guarantees, when the number of sections is large. Let k^* denote the number of sections included using this greedy algorithm, and $F_{k^*,greedy}$ denote the corresponding value of the objective function. Let $F_{k^*,opt}$ denote the optimum value of the objective function subject to the cardinality constraint that exactly k sections are present in the solution. We formally state the theorem below (see [10] for the proof).

$$F_{k^*,greedy} \geq \left(1 - \frac{1}{e}\right) \cdot F_{k^*,opt} - \frac{\lambda \cdot k^*}{e}.$$

5. EVALUATION

We next perform empirical validation to demonstrate the efficacy of our approach in identifying the subset of sections that best covers the material presented in a video relevant to the chapter.

Dataset: We first construct a ground truth test set of videos for each textbook chapter. However, given the huge number of videos available online, it is infeasible to create such a set by inspecting all the videos. Therefore, we take a different approach: We consider the first five chapters of a 9^{th} grade science book. We chose this textbook for two reasons. First, these chapters span different sub-branches of science: Physics (Chapter 1: "Matter in our surroundings" and Chapter 2: "Is matter around us pure"), Chemistry (Chapter 3: "Atoms and molecules" and Chapter 4: "Structure of the atom"), and Biology (Chapter 5: "The fundamental unit of life"). There are about 5 sections (median value) in these chapters. Second, these chapters differ in the extent to which there is content overlap and commonality across sections. These differences help us to characterize when our approach is most beneficial. Although our approach uses COMITY algorithm at the *chapter level* to obtain the candidate set of relevant videos, for the purposes of comparative evaluation, we chose to apply COMITY algorithm at the *section level* (further explained in the next subsection). That is, for each chapter, we run the COMITY algorithm, but by restricting to combinations of top n *cphrs* that are present in a section. We set $n = 20, t = 50$, and $k = 20$. This process resulted in 178 unique videos across all chapters. We assigned a human assessor to read all these five book chapters. After reading the chapters, the judge is asked to watch each video and manually identify all the sections that together capture the content of the video[2]. The judge can revisit the book to read multiple times. Note that the judge does not have access to the underlying algorithm that identified the video. The judge is also asked to remove videos that are irrelevant, or cover material beyond the scope of the book. This judgment process resulted in 112 videos (denoted by $\mathcal{V}$) along with their sections assignments. In particular, for each video v, $\mathcal{S}_v^G$ is the set of ground truth sections assigned.

Baseline algorithm: We also used COMITY algorithm's assignments as the baseline for comparison. Specifically, for each video v, we associate all the sections for which it was retrieved as a top ranking video, and we denote this set as $\mathcal{S}_v^C$. In fact, only about 50% of the videos are assigned to a single section, 25% to two sections and the remaining to more than two sections. Thus, COMITY can be used as a baseline since it also identified multiple sections for the same video (in nearly half the cases).

Metrics: For each video v, let $\mathcal{S}_v^P$ be the set of sections identified by our proposed algorithm.

Accuracy: This metric measures how accurately an algorithm can identify the entire set of sections that best captures the content in the video: Accuracy $= \frac{\sum_{v \in \mathcal{V}} I[\mathcal{S}_v^A = \mathcal{S}_v^G]}{|\mathcal{V}|}$, where $A \in \{C, P\}$ and $I[\mathcal{X} = \mathcal{Y}]$ evaluates to 1 if the sets $\mathcal{X}$ and $\mathcal{Y}$ have identical elements and 0 otherwise. $|\mathcal{V}|$ is the number of videos in the ground truth.

Relaxed Accuracy: The above accuracy metric is stringent in that it requires all the sections identified by the algorithm to match with that of the ground truth. We define a relaxed version that takes into account how different the inferred set is from the ground truth set: Relaxed Accuracy $= \frac{\sum_{v \in \mathcal{V}} \left(1 - \frac{|\mathcal{S}_v^A \triangle \mathcal{S}_v^G|}{|\mathcal{S}_{all}|}\right)}{|\mathcal{V}|}$, where $A \in \{C, P\}$, $|\mathcal{S}_{all}|$ denotes the number of sections in the chapter, and $\mathcal{S}_v^A \triangle \mathcal{S}_v^G$ denotes the symmetric set difference between the set of sections

[2] Our initial experiments confirmed that this task was not suited for Amazon Mechanical Turk (due to the volume of work per judge).

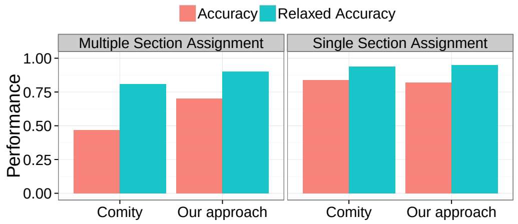


Figure 3: Performance based on COMITY assignment

identified by an algorithm and the set of ground truth sections.

5.1 Results

We evaluated the algorithms based on two different ways of slicing the data: (A) grouping based on the number of sections assigned by COMITY to evaluate overall performance, and (B) chapter–wise results to understand performance based on chapter characteristics.

Performance based on COMITY assignments: Here, we compare the two algorithms based on the number of sections to which a video is assigned to by COMITY. To this effect, we partitioned the videos into two groups: videos that are assigned to only one section by COMITY, and those that are not. Roughly 50% of the videos fall into either of these two groups.

Figure 3 shows the results. We can see that when COMITY assigns a video to multiple sections, in many cases, it does so incorrectly, as shown by the achieved accuracy of 0.47. On the other hand, our approach is able to assign videos to the appropriate subset of sections with much higher accuracy (0.73). Under the relaxed accuracy metric, COMITY's performance is still lower than our approach (0.81 *v.s.* 0.90), indicating that even though the videos considered are relevant (recall our assumption that relevant videos are provided at the chapter level), COMITY either incorrectly assigns additional sections or finds only a subset of the ground truth sections. We further analyzed failure cases and found that our approach often fails to assign the right set of sections due to insufficient representation of the video, arising from the inherent restriction of issuing queries based on the section content.

For the group of videos where COMITY assigned to only one section, there is no significant difference in performance between the two methods. We investigated the reasons for this similar performance: For a video belonging to this group, the corresponding section often tends to be very focused on a particular topic (we discuss this next), and hence there is only a single logical section to which the video could be assigned. Consequently, the two methods result in similar performance for such videos.

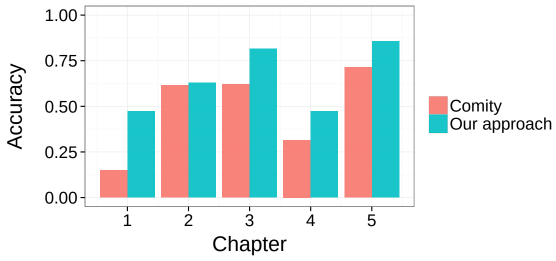


Figure 4: Performance across chapters

Performance across chapters: We also investigated if there is difference in performance across chapters. Figure 4 shows the results. We further analyzed two chapters, one for which the two methods had similar performance and the other with huge difference in performance. For the former, we found that the corresponding sections in the chapter "Is matter around as pure" have unique focus: for instance, section 2 deals with different types of mixtures, while section 3 presents procedures for separating mixtures. These sections do not overlap much in terms of the concept phrases explained. As a result, videos assigned to each section are unique, and thus, the content of each video is not shared across sections in the chapter. In contrast, in chapter 1 titled "Matter in our surroundings", the first section explains the physical nature of matter, while the second one discusses the characteristics of particles of matter, leading to a huge overlap in the content of these sections. This commonality across sections results in videos that have similar content. Since our approach explicitly models these dependencies, it is able to assign the videos more accurately. In contrast, COMITY is myopic and hence is unable to tease out the relationships between sections in the chapter.

6. SUMMARY AND FUTURE WORK

In this paper, we introduced the problem of identifying a set of logical units in a textbook that best captures the content in a relevant educational video. We provided a scalable solution that is effective across various subjects and for educational videos in the wild.

Through this work, we have only touched the tip of the iceberg for effective augmentation of textbooks with videos. There are multiple other considerations such as presenter [12] or presentation styles that need to be taken into account. We also need to design rigorous evaluation methodology factoring in these considerations and perform large scale user study in classroom settings [4]. In a blended learning setting, a teacher may choose to combine course materials including multimedia presentations from multiple courses. Our work is a step towards addressing challenges that arise in such settings.

7. REFERENCES

[1] R. Agrawal, M. Christoforaki, S. Gollapudi, A. Kannan, K. Kenthapadi, and A. Swaminathan. Mining videos from the web for electronic textbooks. In *ICFCA*, 2014.

[2] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Enriching textbooks with images. In *CIKM*, 2011.

[3] R. Agrawal, S. Gollapudi, K. Kenthapadi, N. Srivastava, and R. Velu. Enriching textbooks through data mining. In *ACM DEV*, 2010.

[4] R. Agrawal, M. H. Jhaveri, and K. Kenthapadi. Evaluating educational interventions at scale. In *ACM L@S*, 2014.

[5] W. Barbe, R. Swassing, and M. Milone. *Teaching through modality strengths: Concepts and practices*. Zaner-Bloser, 1981.

[6] R. Dunn, J. S. Beaudry, and A. Klavas. Survey of research on learning styles. *Educational leadership*, 46(6), 1989.

[7] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI*, 2007.

[8] P. Honey and A. Mumford. *The manual of learning styles*. Maidenhead, 1992.

[9] S. Huston and W. B. Croft. Evaluating verbose query processing techniques. In *SIGIR*, 2010.

[10] M. Kokkodis, A. Kannan, and K. Kenthapadi. Assigning educational videos at appropriate locations in textbooks. Technical Report MSR-TR-2014-62, Microsoft Research, 2014.

[11] M. Kokkodis, A. Kannan, and K. Kenthapadi. Assigning videos to textbooks at appropriate granularity. In *ACM L@S*, 2014.

[12] S. Mariooryad, A. Kannan, D. Hakkani-Tur, and E. Shriberg. Automatic characterization of speaking styles in educational videos. In *ICASSP*, 2014.

[13] M. Meeker and L. Wu. Internet trends. Technical report, KPCB, 2013.

[14] M. Miller. Integrating online multimedia into college course and classroom: With application to the social sciences. *MERLOT Journal of Online Learning and Teaching*, 5(2), 2009.

[15] R. Schmeck. *Learning strategies and learning styles*. Plenum Press, 1988.

[16] M. Strube and S. Ponzetto. WikiRelate! Computing semantic relatedness using Wikipedia. In *AAAI*, 2006.

[17] P. Tantrarungroj. *Effect of embedded streaming video strategy in an online learning environment on the learning of neuroscience*. PhD thesis, Indiana State University, 2008.

[18] S. Tarver and M. Dawson. Modality preference and the teaching of reading: A review. *Journal of Learning Disabilities*, 11(1), 1978.

[19] A. Verspoor and K. B. Wu. Textbooks and educational development. Technical report, World Bank, 1990.

[20] Y. Yang, N. Bansal, W. Dakka, P. Ipeirotis, N. Koudas, and D. Papadias. Query by document. In *WSDM*, 2009.

Better Data Beat Big Data

Michael V. Yudelson, Stephen E. Fancsali, Steven Ritter, Susan R. Berman, Tristan Nixon, Ambarish Joshi
Carnegie Learning, Inc.
437 Grant St.
Pittsburgh, PA 15219
(412) 690-2442
{myudelson, sfancsali, sritter, sberman, tnixon, ajoshi}@carnegielearning.com

ABSTRACT

Generalizability of models of student learning is a highly desirable feature. As new students interact with educational systems, highly predictive models, tuned to increasing amounts of data from previous learners, presumably allow such systems to provide a more individualized, optimal learning path, give better feedback, and provide a more effective learning experience. However, any large student/user population will be heterogeneous and likely consist of discernable sub-populations for which specific models of learning may be appropriate. Student sub-populations may differ with respect to cognitive factors, the level and quality of instruction, and many other environmental and non-cognitive factors.

The era of both "big data" and widely deployed educational software, including Carnegie Learning's Cognitive Tutor (CLCT) intelligent tutoring system, presents opportunities to analyze increasingly large volumes of data collected during learners' interactions with educational systems. These data cover a broad spectrum of learners, allowing researchers to investigate the structure of an increasingly representative student population. In this work, we investigate discovering student sub-populations from "big data." Using a year's worth of data from CLCT, we test the hypothesis that commonly used stratifications of student sub-populations (e.g., school location, socio-demographic factors) offer ways to meaningfully partition learners. We discover that, rather than finding distinct subpopulations that should be treated differently, a particular sub-population of learners provides especially "high quality" data and that models learned from this sub-population outperform all other models even when predicting student learning for the sub-population on which other models were trained. In this way, "better data beat big data."

Keywords

Big data, student modeling, learner sub-populations.

1. INTRODUCTION

Generalizability is an important property of any model of student learning developed by researchers and practitioners in educational data mining, learning analytics, and cognitive modeling. As such, investigators generally aim to iteratively refine models of student learning based on data as it is acquired; experimental iteration informs future versions of computer-based educational systems so that such systems can adapt to (and better serve) larger populations of learners.

Discovering the appropriate grain size (e.g., learning models at the group-, school-, or class-level versus individualized, student-level models) to achieve such generalizability is a topic of recent interest in the literature. The student population (i.e., the user base of an educational system) is likely to be heterogeneous, and important aspects of its structure can potentially be identified. Student sub-populations may have particular characteristics and profiles that can be stratified with respect to demographics, learning capabilities, instructional quality, among other factors. Less clear are ways in which such stratifications can be useful for determining sub-populations over which better models of student learning might be learned.

A body of prior work goes beyond building models of undifferentiated populations, modeling individual student differences [4, 8] and also modeling groups of students (e.g., classes and schools) [5, 7]. Other work builds models of student behavior and compares sub-populations defined by school setting (e.g., urban, suburban, or rural) [1]. Most efforts to model individual student differences or to stratify student sub-populations consider relatively small datasets, with an exception of work by Pardos and Heffernan that uses the largest open access dataset on student learning currently available – the KDD Cup 2010 dataset.[1]

On an industrial scale, adapting at the student- and/or group-level provides an opportunity to deliver an optimized learning experience to a large user base, for example, the hundreds of thousands of users of Carnegie Learning's Cognitive Tutor® (CLCT) intelligent tutoring system (ITS) [6]. Using CLCT data, we focus on the discovery of student sub-populations over which parameters used to track student mastery of knowledge components (KCs) or skills can be learned (i.e., "tuned") to better deliver instructional content to different sub-populations. Little (if any) prior research considers what data to include in an *a priori* school profile that might determine appropriate sub-populations (i.e., groups of schools) for such tuning and similarly for *a posteriori* profiles that include student interaction data after CLCT has been used for a substantive period of time.

In this work, we explore the possibility of utilizing information about a particular school (e.g., demographic and socioeconomic indicators) and about its students (e.g., prior performance) to effectively structure a large selection of schools into distinctive groups to determine if and how groups of schools might benefit

[1] KDD Cup 2010 http://pslcdatashop.web.cmu.edu/KDDCup/

from a specific parameter tuning of the CLCT. We set out to discover generalizable sub-populations of schools, but rather we find that a subset of schools provides "high quality" data, models of which effectively generalize to all schools in our sample and outperform (in terms of prediction accuracy on held out data) models learned on other subsets and larger samples of data. In this sense: better data beat big data.

2. CARNEGIE LEARNING COGNITIVE TUTOR

CLCT is an ITS for mathematics that uses cognitive modeling to structure a target domain (e.g., algebra) into knowledge components (KCs). CLCT adapts instruction based on its assessment of which KCs a learner has or has not mastered at any given moment. CLCT provides feedback as to the correctness of their actions on problem-solving steps and also provides context-sensitive hints upon request. Curricula, like algebra, are divided into units of instruction; units are comprised of topical sections, and sections consist of individual problems that are broken up into steps. Problem-solving steps are tagged with one or more KCs.

As students solve problems, CLCT updates its assessment of students' KC mastery using a probabilistic framework called Bayesian Knowledge Tracing (BKT) [3]. BKT is a Hidden Markov Model with two hidden states, representing whether a particular KC is un-mastered or mastered. Observations of student performance on opportunities to practice a KC are binary: a student either solves a problem step correctly or not (due to error or because of a hint request). While students might go through dozens of attempts to get a particular step correct, traditionally, only students' first attempts are considered for updating KC mastery estimates.

BKT uses probabilistic parameters to capture the nature of mastering a skill. These parameters are the probability of knowing the skill *a priori*, the probability of learning the skill at the next practice attempt (i.e., transitioning from the unknown state to the known state), the probability of guessing correctly while in the un-mastered state, and the probability of slipping (i.e., answering incorrectly despite being in the mastered state). In the commercial deployment of CLCT, BKT parameters are set by hand by cognitive scientists and also go through revisions based on data.

3. DATA

We consider a large set of CLCT student usage data, collected in 2010. Although the tutor was used in several thousand schools across the United States, we do not collect detailed interactions for all schools, so our initial data covered 144,080 registered student accounts in 899 schools with close to 473 million records overall, including activity unrelated to problem-solving (e.g., login) as well as solving practice problems. Unfortunately, not all registered students used the tutor or attempted more than one unit of the curriculum. After trimming down the data we arrived at a dataset that included 342 schools, 72,082 active students, and 88.6 million problem-solving actions.

We queried the National Center for Education Statistics (NCES)[2] for school metadata that included: the number of students enrolled (as a proxy of school's relative size), student-teacher ratio, number of students eligible to receive free or reduced price lunch (as a proxy for socioeconomic status), and the school's location (metropolitan area): rural, suburban, or urban. Although some of the school metadata from NCES were from the year 2011, we assume that year-to-year fluctuations are negligible. We matched NCES data and our data and arrived at a set of 232 schools, narrowing our selection to 55,012 students with substantive usage (i.e., attempting more than one unit of instruction) and 67.3 million problem-solving transactions.

In addition to school metadata, we computed school-level student performance statistics from our logs. For each school, we have computed: the average number of distinct units students were attempting, the standard error of the mean number of units attempted, number of distinct units students attempted. We have also retained a binary vector of units attempted by schools' students for grouping schools based on the similarity of attempted units.

To further characterize schools, we ran a mixed effects logistic regression model on the data (see Eq. (1) and Eq. (2)). Here, θ_i represents the ability of student i (a student intercept), and β_j is a problem complexity intercept. For each skill k relevant to problem j, δ_k is general skill easiness (i.e., a skill intercept), and γ_k represents skill k's learning rate; t_{ik} captures student i's number of prior attempts at skill k.

$$m_{ij} = \theta_i + \beta_j + \sum_k \left(\delta_k + t_{ik}\gamma_k\right) \qquad \text{Eq. (1)}$$

$$\Pr(Y_{ij} = 1 \mid \theta, \beta, \delta, \gamma) = \frac{1}{1 + e^{-m_{ij}}} \qquad \text{Eq. (2)}$$

In this regression model, we treat the student- and problem-intercepts as random factors. From the regression coefficients, we calculated the following values to describe, per school: average student intercept (denotes relative prior preparation of students), average skill intercept (to capture each school's general level of skill difficulties on top of student preparation), and average skill slope (to denote the relative speed of learning for students). Thus, overall we have collected, for each school, four *a priori* metadata descriptive factors and seven *a posteriori* student performance descriptive factors.

4. APPROACH

We seek to determine if, based on one or more descriptive factors described above, it is possible to effectively separate schools in our dataset into groups such that schools within groups are more similar to each other in terms of learning than to schools in other groups. We propose to use the accuracy of student modeling as a measure of similarity. That is, if a student model fit to a particular group of schools predicts performance of students in these schools better than models fit to the data of other groups of schools and this is true for all group models, then the school grouping in question effectively separates schools into distinguishable sub-populations.

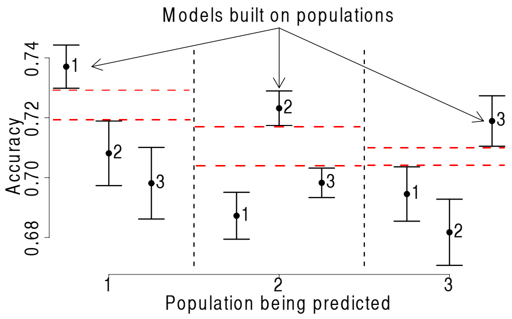


Figure 1. An example criterion of a good split into sub-groups

[2] National Center for Education Statistics http://nces.ed.gov

An illustration of an effective separation of schools into groups is shown in Figure 1. In this graph using idealized data, all schools are split into three groups (or populations). Based on the data from each of the groups we built three models. Each of the three models are used to predict held out data from each of the three groups of schools giving us 3*3=9 predictions. Prediction of held-out data for group of schools #1 is shown in the leftmost column where the accuracy of each of the three models' predictions are shown as dots with serifs denoting standard errors of the mean. Here, we see that model built on group #1 performs better on held out data than models built on the data from groups #2 and #3. Since the range of the serif denoting standard error of the mean for model #1 does not overlap with serif ranges for models #2 and #3, the advantage of model #1 is deemed "significant." Columns 2 and 3 show the same phenomenon: a model built on the data from the respective subgroup outperforms models built on other subgroups.

4.1 Dividing Schools

We have considered all eleven descriptive factors to guide groupings of schools: 1) school locale, 2) percentage of students eligible for free and reduced priced lunch, 3) student-teacher ratio, 4) enrollment, 5) average student units attempted, 6) standard error of student units attempted, 7) number of unique units students attempted, 8) school unit coverage group (based on similarity of binary vectors of distinct units attempted by students in particular school)[3], 9) average student intercept from the logistic regression model (a proxy of average student preparation in the school), 10) average skill intercept for the school from the logistic regression, and 11) average logistic regression skill slope for the school. The factors are grouped into three batches: school metadata factors that are known *a priori*, student usage statistics factors that can be computed from surface logs of student activity, and student model factors that require detailed data to be derived.

Among all factors, school locale and the school unit coverage group are categorical factors. We binned the remaining nine continuous factors into three value ranges – low, medium, and high – so that the number of students in all three is roughly the same. In addition to splitting schools using just one factor, we have computed school splits based on multiple factors. Namely, all factors from all groups[4], only school metadata factors, only student usage factors, only student model factors, and all *a posteriori* student factors (student usage and model factors). The multi-factor groupings were produced with the help of `R` package `cluster` using Goward distances metric and Ward's hierarchical clustering algorithm via function `hclust` with the number of clusters set to 3 for simplicity.

4.2 Cross-Validating School Groups

Since the number of the schools varied across single-factor and multi-factor splits, we sampled 30 schools from each group where 20 schools were used for training a group model and 10 schools were set aside as held out test data. Rather than relying on single-point estimations of model accuracy, we repeated sampling 20 times and obtained the means and the standard errors of prediction accuracies. Thus, for each grouping we selected 20 (samples)*3(groups)*2(fit and test)=120 data sets; within each of the 20 samples fit and test data for a particular group of schools did not overlap, while across samples they could.

For each of the 20 samples we fit three group models. Each of the three models is used thrice to predict three held-out data sets for each of the groups (9 predictions overall). Fitting models and producing prediction accuracies was done with the help of a BKT utility built for use with large datasets [8].

We stipulate that, in order for a grouping of schools to be considered producing distinct groups, for every group, the in-group prediction should be significantly better than out-group prediction (cf. Figure 1).

5. RESULTS

First, we consider several school metadata factors, knowable *a priori* (prior to any student usage of CLCT). Figure 2 is a group split graph for school enrollment. As we can see, models built on groups of low and middle ranges of enrollment are not discernable from each other across all prediction tasks. The model built on high enrollment schools is visibly worse even when predicting held out data of high enrollment schools.

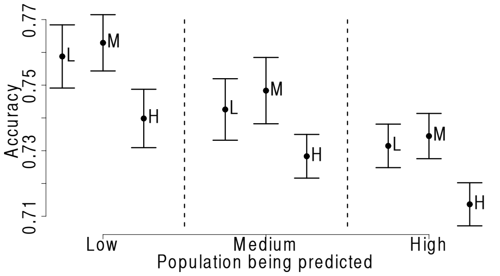


Figure 2. Group separation by school enrollment

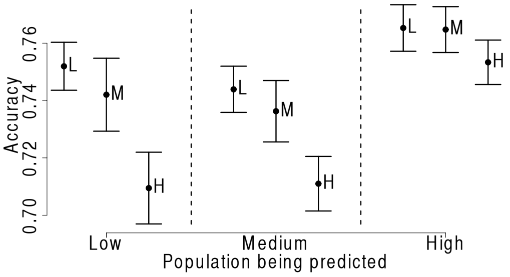


Figure 3. Group separation by the ratio of students eligible for free and reduced price lunch

Figure 3 is a group separation graph for the ratio of students eligible for free and reduced price lunch. Again, we see that this factor is not separating schools into reliably discernable groups. Models built on schools with a high proportion of students eligible for free and reduced price lunch are visibly worse across all populations, while models of low and medium groups are not discernable, again across all populations.

Neither school metadata factors separately nor a grouping based on a clustering solution of these metadata factors produce a desirable split. Instead, we see model accuracies lined up in identical fashion: one particular model is a slightly better predictor universally; a second model is slightly worse, and the remaining model is worse than the second.

However, for 3 out of 7 remaining individual factors and one multi-factor case (all factors but metro area), models built on one group of schools are consistently and significantly better than other models in at least 2 prediction tasks. See, for example,

[3] The grouping was done with the help of `R` package `cluster` using Euclidean distances and Ward's hierarchical clustering algorithm via function `hclust` with k=3.

[4] School locale factor was excluded since using it defaulted the clustering to be identical to the metro area factor itself.

Figure 4. Here, schools where students finish a high number of units on average (more than 9.2 units) produce a model that outperforms another model in two out of three comparisons and ties in third.

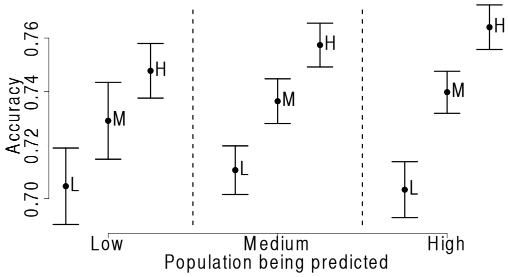


Figure 4. Group separation by average student units attempted

We find a similar pattern for average student intercept (a proxy of average student preparation), where the model built on a group of better-prepared students wins in two comparisons and ties in one. The third factor with one-model-trumps-all is the average skill slope (a proxy of speed of learning), where the winning model actually is built on the group of schools where the average skill slope is in the medium range. When cross-correlated, only the correlation of average units attempted and average student intercept is relatively high and significant (r=0.56, p<0.001).

6. DISCUSSION

We set out to discover subsets of schools for which models of practice could be built for sub-populations to optimize the CLCT learning experience for students in that sub-population. Instead, we find that particular sub-populations of schools can be used to learn parameters that perform best over the *entire population*. In essence, we have identified a set of schools for which particular aspects of their interaction with the CLCT provide high-quality (e.g., less "noisy") data for such model building.

While this substantial subset may still count as "big" data, we disregard a large number of students to arrive at this generalizable model, and the characteristics along which the group of schools from which these students are drawn are not obvious *a priori*. While much focus is placed on the revolutionary potential of big data applications in education, careful consideration and attention must be paid to the quality of such data for particular purposes and application contexts.

We find that the sub-populations that yield a universally better model tend to contain students who are better prepared and students who attempt more CLCT units.. However, with respect to average skill learning rates, the best model contains many students in the "middle" group. At this point we hypothesize that students that should be considered for inclusion in learning a generalizable model are not just better students but those that yield a substantial data footprint in terms of curriculum coverage. Students who should likely be excluded are those who only cover a fragment of units, insufficient to provide for a "good" model.

Several caveats could hinder how strongly the phenomenon of "better data" vs. "big data" manifests itself. One is that CLCT allows instructors to deploy "custom" curricula; different schools sometimes use different content units and, as a result, practice different skills. Consequently, when validating the model on the held out data where a particular unit was not practiced, we used default modeling parameters that could potentially lead to lower accuracy. Together with known issues with fitting BKT models (e.g., local maxima and non-identifiability [2]), this might have led to the inter-group differences being underestimated and the effect of "one group model takes all" – lessened.

Second, we cannot judge, for example, whether our 2010 dataset constitutes a representative sample of all US schools with respect to the school metadata variables we considered. However, we estimated whether our selected subset of 232 schools maintains the same distribution of the school locale (i.e., whether schools are rural, urban, and suburban) as that over 729 school of our original 899 schools for which we have appropriate data to make the comparison. The split between rural, suburban, and urban schools in the larger sample of 729 schools are 29%, 33%, and 38%, respectively. Our smaller sample of 232 schools breakdown as 29%, 25%, and 46%, respectively. While the percentage of the rural schools is the same, the percentage of urban schools significantly grew, and the ratio of suburban schools declined. While this may introduce bias, it is unclear whether such bias, given the relatively large sample overall, would have a substantive impact on the generalizability of our results.

7. REFERENCES

[1] Baker, R. S. J. de & Gowda, S. M. (2010). An Analysis of the Differences in the Frequency of Students' Disengagement in Urban, Rural, and Suburban High Schools. In 3rd International Conference on Educational Data Mining (EDM 2010), Pittsburgh, PA, USA, 2010 (pp. 11-20).

[2] Beck, J. E. & Chang, K.-m. (2007). Identifiability: A Fundamental Problem of Student Modeling. In User Modeling, Corfu, Greece, 2007 (pp. 137-146). Springer.

[3] Corbett, A. T. & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4(4), 253-278.

[4] Lee, J. I. & Brunskill, E. (2012). The Impact on Individualizing Student Models on Necessary Practice Opportunities. In Proceedings of the 5th International Conference on Educational Data Mining, Chania, Greece, June 19-21, 2012, Chania, Greece, 2012 (pp. 118-125).

[5] Pardos, Z. A. & Heffernan, N. T. (2010). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In 18th International Conference on User Modeling, Adaptation, and Personalization (UMAP 2010), Big Island, HI, USA, 2010 (pp. 255-266). Springer.

[6] Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.T. (2007). Cognitive Tutor: applied research in mathematics education. Psychon Bull Rev, 14:249-255.

[7] Wang, Y. & Beck, J. (2013). Class vs. Student in a Bayesian Network Student Model. In 16th International Conference on Artificial Intelligence in Education (AIED 2013), Memphis, TN, USA, 2013 (pp. 151-160). Springer.

[8] Yudelson, M., Koedinger, K. R., & Gordon, G. J. (2013). Individualized Bayesian Knowledge Tracing Models. In 16th International Conference on Artificial Intelligence in Education (AIED 2013), Memphis, TN, USA, 2013 (pp. 171-180). Springer.

Building a Student At-Risk Model: An End-to-End Perspective

Lalitha Agnihotri, Ph.D.
McGraw Hill Education
2 Penn Plaza
New York, NY 10121
1-914-434-2372
lalitha.agnihotri@mheducation.com

Alexander Ott, Ed.D.
New York Institute of Technology
Northern Boulevard
Old Westbury, NY 11568
1-516-686-1037
aott@nyit.edu

ABSTRACT

Poor graduation and retention rates are widespread in higher education, with significant economic and social ramifications for individuals and our society in general. Early intervention with students most at risk of attrition can be effective in improving college student retention. Our research aim was to create a first-year at-risk model using educational data mining and to apply that model at New York Institute of Technology (NYIT). Building the model creates new challenges: (1)the model must be welcomed by counseling staff and the outputs need to be user friendly, and (2)the model needs to work automatically from data collection to processing and prediction in order to eliminate the bottleneck of a human operator which can slow down the process. The result of our effort was an end-to-end solution, including a cost-effective infrastructure, that could be used by student support personnel for early identification and early intervention. The **St**udent **A**t-**R**isk Model (STAR) provides retention risk ratings for each new freshman at NYIT before the start of the fall semester and identifies the key factors that place a student at risk of not returning the following year. The model was built using historical data for the 2011 and 2012 Fall Class and the STAR system went into production at NYIT in Fall 2013.

Keywords

Students At-Risk Model, Ensemble Model, End-to-End system

1. INTRODUCTION

On average less than 60% of full-time students who begin a four-year program of college study graduate in six years [7]. Moreover, the highest rate of attrition occurs during the first year of study — from the student's first fall semester to what would be his or her second fall. Figure 1 shows a box plot of graduation and first year retention rates in the United States for 2006, 2007, 2008, and 2009. (The dataset in the box plots derives from the Delta Cost Project [5], which in turn is based on Integrated Postsecondary Education Data System (IPEDS) data as made available by the National Center for Education Statistics [9].) The graduation rates are around 50% and the first year retention rates are clustered around 70%. Therefore, the logical starting point for improving graduation rates would be to find ways to improve first-year retention.

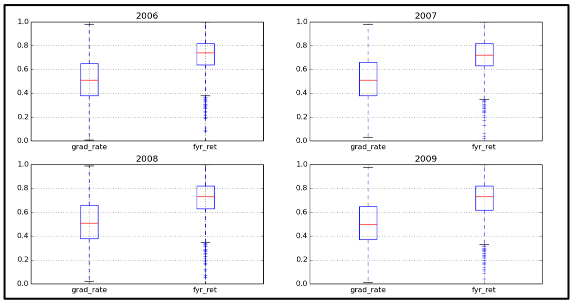


Figure 1. Graduation and First Year Attrition Rates

Research shows that counseling intervention with students at highest risk of attrition can be effective in improving retention [6][10][12]. Essential to this intervention, however, is that it be early in the student's first semester at college [12]. The problem is twofold: (1)how to identify and intervene with these at-risk students before it is too late, and (2)how to identify the key factors putting these students at higher risk of attrition so as to inform the counseling intervention and improve its effectiveness. Given that evidence exists in the literature that it is possible to build a data-mining based model that would address both dimensions of the problem [13], NYIT undertook such an effort, beginning in earnest in the fall of 2012.

However, having the most powerful predictive model would not be useful if either (1)the counseling staff that must ultimately use the model were not willing to do so, either because they did not want the model in the first place or because they were uncomfortable with the model outputs, or (2)the model itself needs intensive manual intervention to produce the output, therefore slowing it down. To overcome these challenges we needed to employ an "end-to-end" approach in user-oriented model creation as well as in building a highly automated model on a technological level.

In terms of building a user-oriented model, the model itself was built in an iterative cycle between the end users and the IT model creator. The problem definition came from the actual users—the retention-focused counseling staff. The data identification was done in collaboration with the solution provider and the users. The IT solution provider did the data gathering and preparation, model building, evaluation, and deployment. Once the knowledge deployment occurred, the users were looped backed in to provide feedback and critique and the process was restarted.

On the technological level, we similarly used an end-to-end solution to build a highly automated model: We used the

Microsoft SQL server as our tool of choice and built the database, prediction models, and the front end used by the counselors all on the same platform. The model was built "in house" at NYIT.

The resulting **St**udent **A**t-**R**isk Model (STAR) provides retention risk ratings for each new freshman at NYIT before the start of the fall semester and identifies the key factors putting a student at risk to not return the following fall. The model was built and used for intervention for the incoming Fall 2013 freshman class.

2. NYIT STAR Model, Version 1.0

NYIT's Student Solutions Center (SSC), which is NYIT's "one-stop-shop" for enrollment services, engages with new students by providing counseling guidance to improve student success and retention. The SSC's counseling intervention is called the 4-3-2-1 Plan, which involves individual counselor-new student meetings occurring early in a new student's first semester at NYIT.

In fall 2011, the SSC attempted to build its own model to identify the most at-risk students, therefore allowing earlier, targeted intervention with these students. This STAR Model, version 1.0, was rather simplistic in its approach. We essentially gathered data on each student from multiple sources and compiled in one Excel sheet. We then used the retention literature and our own inclinations to identify variables and assign each variable with a score of "1" or "0"—with a 1 being a retention risk. The higher the score for each student, the more at risk he or she is.

As one might expect, this approach was highly problematic. On a conceptual level, it was based on student behavior at other institutions (via the retention literature) and not behavior at NYIT. It was also a blunt instrument, in that all variables were weighted equally. On a practical level, compiling the Excel sheet involved significant labor, gathering information from multiple data sources manually.

2.1 NYIT STAR Model, Version 2.0

In order to overcome these "Model 1.0" limitations, we took two key steps: First, we built the dataset in our Data Warehouse where it can be created automatically as soon as a new student registers. Second, we decided to use data mining tools to train machine-learning models to perform the classification task. These models use the variables to predict whether or not a student will return the following year which is then used to flag the risk of new students.

We chose Microsoft SQL Server for the following reasons: All our data exists in the SQL Server; the SQL Server Analysis Services (SSAS) provides capability to use Neural Networks, Naïve Bayes, Logistic Regression, and Decision Tree models for prediction. And finally once a model is trained, SQL Server Reporting Services (SSRS) allows us to query the model on an on-demand basis to populate a report hosted on a SharePoint site that serves as the front end for the counselors to access the data. Figure 2 shows our Microsoft Business Intelligence stack that we used for building the STAR model. The SSAS Modeling part happens only once.

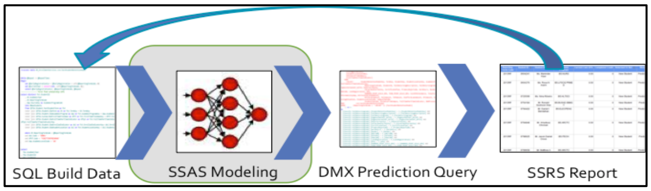


Figure 2. End-to-End Technology Solution

We selected a total of 25 variables after multiple iterations. These variables are from the same three sources as in STAR 1.0—students' admission application data, registration/placement test data at NYIT, or from a survey that the students complete when they take the Compass placement exam at NYIT. However, STAR 2.0 also includes financial data, which research indicates plays a role in retention risk, albeit a complicated one (see, for example, [4][8]).

3. Data Mining Models

Campbell et al. [3] propose the use of analytics to academics, which is what we have attempted in building STAR 2.0. Bayer et al. [1] used student data enriched with data derived from students' social behavior to predict student failure. This works well for longitudinal snapshot data. Romero et al. [11] present data mining methods for classifying students based on their Moodle usage data. They have defined a set of attributes specific to Moodle usage and compared a number of methods and their algorithmic implementations. Taylor and McAleese [15] presented a system that uses data intelligence and analytics for more efficient and effective student success interventions, though they used an analytics company to do the modeling effort. Since we developed our models in house, the data stays in house, so there are no security issues—a significant advantage. Further, having access to the models enables us to drill down into the prediction results to give a detailed picture for each student, as will be demonstrated more fully later. In addition to all the data mining methods, use of ensemble models is growing in popularity as it has the ability to generalize much more than any single method. Yu et al. [16] used Ensemble models in their classifier for 2010 KDD Cup and won the first prize in the challenge.

The process of modeling involves training and testing of multiple models and then selecting the one that works the best for the application. We chose to build four different initial models: Neural Networks, Naïve Bayes, Decision Tree, and Logistic Regression. On top of this we built an Ensemble model that, in addition to taking the variables for each student, takes the output of the initial four models as input and predicts whether a student is at risk or not.

3.1 Model Performance

Key measurements that are used to gauge the power of a predictive classification model are recall and precision. Recall compares the number of students who were predicted as not returned with all those who actually did not return. That is, recall measures the following: Of all those students who actually did not return the following fall, what percentage were correctly predicted by the model as not returning? Precision compares the number of students correctly predicted as not returned with all those who are predicted by the model as not returned. That is, precision measures the following: Of all those students predicted by the model not to return, what percentage of those students actually did not return the following fall?

3.1 Model Selection

Each of the models has a number of parameters that can be changed. We analyzed each of the models and decided to vary the parameters to generate almost 400 variants of the initial models. For example, the reasoning for the number of models that can be built for Naïve Bayes is derived as follows. The Naïve Bayes model has four parameters that can be varied: Maximum Input Attributes, Maximum States, and Minimum Dependency

Probability. We chose not to alter the first parameter that enables feature selection for reasons explained below. We change states from 0 to 250 in steps of 10 for total of 26 different values. Also we vary the minimum dependency probability from 0.1 to 1 in steps of 0.1. In total we get 26*10=260 models of Naïve Bayes.

In order to eliminate feature selection, we ran the 20 possible neural networks with all possible values for parameter Maximum Input Attributes going from 1 to 25 in steps of 1 for a total of 25*20=500 models in all. The same recall and precision was obtained for classifiers with features 21 or more that was the best recall of all the 20 models generated. Further, since all our features are readily available, we decided not to deal with feature selection for our modeling process in order to speed up the development of a model that can be used. In the future, we will revisit the feature selection more rigorously to eliminate variables that may be irrelevant.

Based on analysis for each of the models, we trained a total of 372 models: 20 for NN, 26 for Logistic Regression, 66 for DT, and 260 for Naïve Bayes. SQL Server Analysis Services (SSAS) has a scripting language called DMX that can be used to automatically generate and train models. A DMX script was generated using a SQL query to generate these models automatically. Once the models were generated, their recalls and precisions were computed automatically using another SQL query and stored in a table and the ones with highest recall were selected. If multiple models had the highest recall, we chose the one that provided the highest precision. The model that gave us the highest recall (and precision) for each of the four models was then chosen to generate the ensemble model. The ensemble model takes all of the student data as input as well as the output from the four initial models.

A total of 1453 students who were admitted in fall of 2011 and 2012 were used for the purpose of training and testing the models. Of these 983 students returned to the campus in following fall and 470 did not return. We used 70% of the data for training and 30% for testing. SQL Analysis Services randomly samples the data to help ensure that the testing and training sets are similar.

Following are the models and the selected parameter value chosen for each of the types based on the automatic selection of 372 total models built.

1. Decision Tree: Complexity Penalty = .8, Score Method = Bayesian with K2 Prior, Split Method = Binary
2. Logistic Regression: Maximum States = 10
3. Naïve Bayes: Maximum States = 10, Minimum Dependency Probability = .1
4. Neural Net: Hidden Node Ratio = 19

The Logistic Regression had the best recall and was hence used as the model of choice for the ensemble model. We trained a Logistic Regression model with the same parameters as the chosen initial model to be our final model. This model not only had as input all the student variables, but also the outputs of the four initial models that were chosen automatically as explained above.

3.2 Model Comparison

The model performance using the 2011 and 2012 test data showed a stark contrast between the manual STAR model 1.0 and STAR model 2.0. The recall of the basic four models in version 2.0—Logistic Regression, Neural Network, Naïve Bayes, and Decision Tree—varies from 45% to 62%. This means that the strongest model in version 2.0 in terms of recall was capable of correctly identifying 62% of the not returning population. This represents a major improvement over the 34% recall in version 1.0. In terms of precision, the models in version 2.0 vary from 54% to 70%. This means that in the strongest model in version 2.0 in terms of precision, 70% of those students identified as not returning actually did not return the following fall. This, too, is a major improvement over the 42% precision in version 1.0.

To improve the recall of version 2.0 further, we built an ensemble model that can use the output of the four initial models along with the student data as input and predict whether a student will return or not. This provides the best recall results we have obtained so far, as the model's recall is 74%. This means that the model is able to identify 74% of the students who did not return correctly whereas the other models were able to reach a 62% recall at best. Table 3 summarizes the performance of the models.

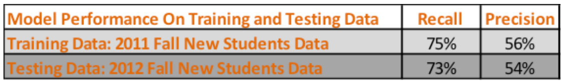
Table 3. Performance Comparison of Models

Model Name	Recall	Precision
Manual (STAR 1.0)	34%	42%
Logistic Regression	62%	57%
Neural Network	56%	54%
Naïve Bayes	51%	69%
Decision Trees	45%	70%
Ensemble	74%	55%

In the end, version 2.0 compares very favorably with not only version 1.0 but also with a similar Data Warehouse-based retention modeling effort described at Western Kentucky University [2]. Bogard et al. report that they were able to achieve a recall of only about 30% based on pre-enrollment data, which is also the data space in which our model operates. As noted, we are able to do much better, in fact up to 74% recall for the ensemble model, due in part to our model selection and also due to the inclusion of financial data and a student survey which is missing in the efforts described by Bogard et al.

We did another validation of our method and trained our models as explained above on Fall 2011 data alone (724 students). Then we used the ensemble model to predict the retention risk on Fall 2012 new students (729 students). As can be seen in Table 4, our model was able to generalize well on 2011 students' data and did a comparable job of predicting the retention risk for Fall 2012 new students.

Table 4. Ensemble Model Validation



Model Performance On Training and Testing Data	Recall	Precision
Training Data: 2011 Fall New Students Data	75%	56%
Testing Data: 2012 Fall New Students Data	73%	54%

The answer to the bigger question of how well our model worked as a guide for actual intervention and as a way to change student enrollment behavior will not be known until Fall 2014.

4. STAR MODEL PRODUCT

As mentioned at the start of this paper, all the smart model building and predictions would not be of any use if they cannot be presented in a manner that is easily digestible and pleasing to use for the counseling staff. This is where our end-to-end iterative model building became essential to the project's success. We built an actual "*product*" that the users could use that had the model and its output under the hood.

As soon as the first version of model 2.0 was complete, we built a report using Microsoft's SQL Server Reporting Services to show the prediction output to the counselors. After several cycles of counselor feedback and report revisions, the counselors had a final, user-friendly product that they were comfortable using and that they participated in creating. The final report tells the counselor which students are at highest risk of not returning the following fall, which allows the counselor to target those students the model is most confident are at risk. Second, the output report lists the reasons for both student risk (e.g., Math placement, affordability) and lack of risk (e.g., high SAT scores, full time enrollment) in the final column. The most recent revision we made to the report was to create a STAR Counselor Log that provides all the output of the model and also allows the counselor to input data about when he or she met with the student, what was discussed, etc.

This STAR Counselor Log is an evolving interface. We have plans to revise this interface further on the basis of feedback from SSC counselors as to how it could be improved. One suggestion is to find a way to categorize a counselor's assessment of the student after the first meeting. For example, we could add a check box that indicates the counselor believes the student is at such high risk that he/she should be followed up with quickly—typically we wait for a second 4-3-2-1 Plan meeting until the following semester.

5. CONCLUSIONS AND CHALLENGES

For colleges considering building an at-risk student model, the key conclusions from the STAR modeling effort are as follows: First, a student at-risk model can be built "in house" if appropriate data is collected and stored in a Data Warehouse. Second, performing data mining is essential to building accurate models in order to weight variables correctly based on student behavior at your institution in particular. Ensemble models can be very useful as the data is rarely clean and each model can only capture so much information on its own. Third, any solution that is provided needs to have an end-to-end perspective in place so that the prediction modeling process is smooth one and the product is user friendly.

As with most attempts to address a complex topic, many challenges remain: First, while the predictive ability of STAR Model 2.0 is quite high, and much higher than STAR Model 1.0, there is still significant room for improvement. For example, the strongest model had a recall of 75%, which means it failed to predict 25% of the students who did not return. Second, assessing whether the STAR-guided intervention is meaningful in a student retention context and, if so, how to demonstrate this. Third, the counseling intervention to at-risk students can affect the model over time. How do we get past the Heisenberg uncertainty principle to build the best model while intervening?

Despite the challenges, the STAR model has been a large step forward at NYIT—it allows NYIT counselors to prioritize intervention with those first-year students most at risk early in the fall semester. This intervention is now based on real student data and is informed by key at-risk variables for each student, allowing the counselor to tailor intervention to the risk factors of each student.

6. ACKNOWLEDGMENTS

Our most sincere thanks to colleagues at NYIT for their contributions to this effort: Mr. Yongxin Ma provided support with Data Warehouse and much wisdom and insight. None of this would have been possible without the support of Mr. Will Wall, SSC Data Guru, who developed version 1.0. And a big shout-out to the counselors in SSC for their invaluable feedback in developing the front end and also their enthusiasm in using the model for intervention. We would like to express our gratitude to Al Essa for his help in editing the paper and also for providing box plots for graduation and first-year retention.

7. REFERENCES

[1] Bayer, J., Bydzovska H., Geryk J., Obsivac T., Popelinsky L. (2012). Predicting drop-out from social behaviour of students. *Educational Data Mining 2012: 5th Intl Conf on Educational Data Mining, Proceedings*. Chania, Greece.

[2] Bogard, M., Helbig, T., Huff, G., & James, C (2011). A comparison of empirical models for predicting student retention. White paper. Office of Institutional Research, Western Kentucky University.

[3] Campbell, J.P., DeBlois, P.B., & Oblinger, D.G. (2007). Academic analytics: A new tool for a new era. *EDUCAUSE Review*, *42*(4), 41–57.

[4] Chen, R., & Desjardins, S. L. (2008). Exploring the effects of financial aid on the gap in student dropout risks by income level. *Research in Higher Education, 49*(1), 1–18. doi: 10.1007/s11162-007-9060-9

[5] Delta Cost Project. Retrieved from http://www.deltacostproject.org/

[6] Fowler, P. R., & Boylan, H. R. (2010). Increasing student success and retention: A multidimensional approach. *Journal of Developmental Education, 34*(2), 2–4, 6, 8–10.

[7] Friedman, B. A., & Mandel, R. G. (2009). The prediction of college student academic performance and retention: Application of expectancy and goal setting theories. *Journal of College Student Retention, 11*(2), 227-246.

[8] Kim, D. (2007). The effect of loans on students' degree attainment: Differences by student and Institutional characteristics. *Harvard Educational Review, 77*(1), 64–100, 127.

[9] National Center for Educational Statistics. (2014). Fast facts. Retrieved from http://nces.ed.gov/fastfacts/display.asp?id=40

[10] Pan, W., Guo, S., Alikonis, C., & Bai, H. (2008). Do intervention programs assist students to succeed in college?: A multilevel longitudinal study. *College Student Journal, 42*(1), 90–98

[11] Romero C., Ventura S., Espejo P.G., & Hervás C. (2008). Data Mining Algorithms to Classify Students, *Educational Data Mining 2008: 1st Intl. Conference on Educational Data Mining, Proceedings*. Montreal, Quebec, Canada.

[12] Seidman, A. (2012). Taking action: A retention formula and model for student success. In A. Seidman (Ed.), *College student retention: Formula for student success* (2nd ed.) (pp. 267–284). Lanham, MD: Rowman & Littlefield.

[13] Singell, L. D., & Waddell, G. R. (2010). Modeling retention at a large public university: Can at-risk students be identified early enough to treat? *Research in Higher Education, 51*(6), 546–572. doi: 10.1007/s11162-010-9170-7

[14] Tampke, D. R. (2009). Developing and implementing an early alert system. In R. Hayes (Ed.), *Proceedings of the 5th National Symposium on Student Retention, 2009, Buffalo*. (pp. 143–151). Norman, OK: The University of Oklahoma.

[15] Taylor, L. & McAleese, V. (2012). Beyond retention: Using targeted analytics to improve student success. *EDUCAUSE Review*.

[16] Yu, H.-F., et al. (2010).Feature Engineering and Classifier Ensemble for KDD Cup 2010. *Conference on Knowledge Discovery and Data Mining*. Washington, DC.

Can Engagement be Compared? Measuring Academic Engagement for Comparison

Ling Tan
Australian Council for Educational Research
19 Prospect Hill Rd,
Camberwell, VIC, Australia 3124
Ling.Tan@acer.edu.au

Xiaoxun Sun
Australian Council for Educational Research
19 Prospect Hill Rd,
Camberwell, VIC, Australia 3124
Xiaoxun.Sun@acer.edu.au

Siek Toon Khoo
Australian Council for Educational Research
19 Prospect Hill Rd,
Camberwell, VIC, Australia 3124
SiekToon.Khoo@acer.edu.au

ABSTRACT

Student engagement is a reflection of active involvement in learning. In digital learning environment, research studies on engagement have been focused on detecting behavioral and psychological engagement indicators from the patterns of activities using feature engineering, but student engagement estimates were rarely compared across sessions or across domains of learning. This paper describes how this could be done by revisiting engagement instrument, diagnosing engagement indicators, estimating engagement parameters, and equating. This study illustrates how engagement reliability can be improved by refining engagement indictors. We demonstrated through DataShop data that student engagement levels can be compared across domains of learning..

Keywords

Behavior, academic engagement, measurement, ITS.

1. INTRODUCTION

In digital learning environment, research study of engagement often focused on detecting behavioral engagement indicators [3,4,5] and psychological engagement indicators [2, 6, 14] using non-intrusive and unobtrusive means. Rather than using surveys to understand engagement, behaviors and affective indicators have been predicted from patterns of activities using feature engineering. The role of machine learning and data mining techniques is to predict behavior or affective status on big data using models developed from training data labeled by human observers. For example, disengagement is inferred by gaming [3, 4] or response time [7]; persistence could be observed by number of revisits to challenging or incomplete tasks [6]; self-regulation could be inferred by the consistency of task completion [1]; and affect status learned from Bayesian Networks [2].

Index of student engagement has been extensively studied to investigate its relationship with learning outcomes. For example, Pardos et.al [14] investigated how well affect states predicted by affect detectors while students worked on exercises throughout a school year in a web-based tutoring platform were correlated with learning outcomes at the end of year. In addition, Rowe, Shores, Mott and Lester [15] found a strong positive relationship between engagement and learning outcomes in narrative-centered learning environments.

This paper is organized as follows. The first section presents the definition of student engagement with a focus on the type of engagement typically found in ITS. The second section presents validity and reliability of academic engagement instrument, diagnostic features of engagement indicators. The last section demonstrates through DataShop data that student engagement levels can be compared across domains of learning.

2. STUDENT ENGAGEMENT CONSTRUT

We argue that there are substantive benefits to study student engagement using methodology found in developing instruments in educational psychology. This approach from instrument point of view offers a number of benefits. Firstly, it sets out to clearly define what kind of student engagement is to be measured at the very beginning. Secondly, it facilitates the comparison of student engagement level across sessions and domains of learning. This means that a student engagement level at the beginning of semester could be compared to the engagement at the middle of semester; and also one’s engagement level on Mathematics can be compared to his/her engagement level on Science. Lastly, engagement estimate would be useful for secondary analysis, e.g. correlation between engagement and learning outcomes, or factors influencing engagement which leads to positive learning gains.

2.1 Academic Engagement Construct

It is necessary to develop a valid and reliable measure of student engagement in order to understand the relationship between student engagement and learning outcomes, and to provide tailored strategies to improve learning outcomes of students. Is it possible to define a blue-print of engagement levels in ITS environment like what we would see in conventional self-report survey instrument? The following section will address this issue.

Table 1 provides a preliminary definition of student engagement by levels and corresponding indicators from observed behavioral activities. The definition of student engagement is based on Skinner and Belmont [17], Bomia et al [8], Schlechty [16], Chapman [9], Markwell [13], Willms [18] and Kember, Biggs and Leung [11], and adapted to the indicators in digital learning environment, drawing on additional works by Baker and colleagues [4,5].

Table 1: Mapping of engagement levels to engagement indicators

Level	Behavior Indicators
Level 5: Enthusiasm in learning	Work on additional tasks. Respond to others’ questions in online forum. Multiple solutions on tasks.

Level 4: Persistency	Revisiting and spent more time to more difficult tasks. Appropriate use of hints. Completion all tasks. Completion on time.
Level 3: Participation	Work on moderately challenging tasks. Completion of minimum number of tasks.
Level 2: Passive participation	Guessing on majority of tasks. Incompletion on all or majority of tasks. Frequent but inappropriate use of hints.
Level 1: Withdrawal	No response on assignments.

2.2 Data Sets

We used 'Assistments Math 2005-2006' and 'Geometry Area (1996-97)' data sets from PSLC DataShop, available at http://pslcdatashop.org [12]. Both data sets were used for analysing student engagement. The first data set (or Algebra data set) contains action logs of 3136 students using ASSISTments Math tutor from middle schools in a city in central Massachusetts in 2005-2006. Students may use the software for two hours, twice a week. This data set contains 834 unique problems, 2,514 unique steps, total 685,615 transactions of attempting to answer questions and/or requesting helps, and total 6,395 student hours. The data set contains a variety of problem classifications (aka knowledge component).

The second data set (or Geometry data set) is a much smaller data set. This data set was used to compare engagement levels found in the first data set. It has action log data of 59 students using Cognitive Tutor for a Geometry course on a single day, 01/Feb/1996. This data set contains 40 unique problems, 139 unique steps, total 6,778 transactions of attempting to answer questions and/or requesting helps, and total 21 student hours. The data set also contains a variety of Geometry knowledge component classifications in Geometry. Cognitive Tutor system determines which skills a student is having difficulty with, and presents each student with tasks of a skill that he or she has difficulty with. In particular, it estimates the probability of a student knowing each skill based on his/her responses recorded in the system, using Bayesian knowledge-tracing [10].

3. RESULTS

Our first research question is whether it is possible to create an academic engagement instrument guided by engagement construct blueprint outlined in Table 1 from action log data typically recorded in ITS.

3.1 Validity and Reliability

We adapted Baker's behavioral classification [5, 6] and extended it into 11 categories in ITS environments: off-task, gaming, guessing, on-task, on-task using appropriate hints, completion minimum work, completion on time, revisit of moderate-difficult tasks, revisit of hard tasks, extra-task, and extra-time. The extended behavioral classification provides a number of indicators to capture moderate to high levels of academic engagement.

The first 5 behavioral indicators are defined at problem level. *Off-task* is defined as no observations on last-*n* temporal-order tasks, or a student is not working on (or skip) some of assigned tasks. *Gaming* is defined as using excessive hints in a short period of time. *Guessing* is defined as going through difficult tasks quickly without using hints, or going through easy tasks without even spending time reading tasks. *On-task* is defined as working on tasks by producing valid responses after spending a minimum amount of time. *On-task using appropriate hints* is defined as on-task while seeking hints on tasks which are moderately hard relative to student's ability.

The remaining 6 behavioral indicators can be defined at any pre-defined session or mini-session level which contains *n* temporal-order problems. *Completion minimum work* is an indicator to show if a student is able to complete minimum assigned tasks in a session. *Completion on time* is an indicator to show if a student is able to complete minimum assigned tasks on time in a session. *Revisit of moderate-difficult tasks* is set to yes if a student took opportunities to revisit the moderately challenging tasks. *Revisit of hard tasks* is set to yes if a student made an additional efforts to attempt challenging tasks. *Extra-task* indicates if a student made additional efforts to practice on tasks beyond minimum requirement. *Extra-time* indicates if a student spent additional time on assignments.

Behavioral indicators including gaming, guessing, on-task, on-task with appropriate hints, revisit of moderate-difficult tasks, and revisit of hard tasks rely on a critical piece of information, i.e. the likelihood of success on a task. For example, guessing occurs when one finds a particular multiple-choice task hard, and it occurs to students of all ability levels. We can reasonably predict if a student is going to guess if we know the likelihood of success of this student on a particular task.

Prior to estimate student engagement levels of behavioral indicators, observations were arranged in temporal order. For Algebra data set, behavioral indicators were created according to problem-level behavior classifications for *n* problems, which were named as B1 to B*n*. In our experiment, *n* was chosen to be a number close to the average number of problems students attempted in a session (i.e. *n*=12). In addition, six indicators, i.e. completion minimum work, completion on time, revisit of moderate-difficult tasks, revisit of hard tasks, extra-task, and extra-time, were created at session level based on action logs from these *n* problems. ACER ConQuest software [19] was used to estimate KC difficulties and person ability, and the probability of success for each person on each KC was then calculated in SPSS.

What engagement levels are typically found in elements of this instrument? Are the rank orders of instrument indicators working as expected? Figure 1 shows variable map for Algebra data set. The engagement indicators represented by B1 to B12 and the names of six other indicators are displayed on the right hand side of map. The latent engagement levels of individuals represented by "X" are shown on the left hand side. The number of cases represented by each "X" is indicated at the bottom of the variable map. Students at the top of the distribution have higher engagement estimates, while engagement indicators at the top end require higher level of efforts.

The variable map shows that it takes an increasing amount of time or efforts for students to complete more tasks, as indicated by increasing rank order of B1 to B12 in the map. It also shows that it takes more efforts to complete minimum tasks on time than just to complete minimum tasks. Students who put additional efforts on revisiting challenging tasks, investing more time, or working harder on extra tasks are shown to be more engaged than those who just completing minimum tasks.

```
----------------------------------------------------
   3            |                                  |
                |                                  |
                |                                  |
               X|                                  |
               X|                                  |
               X|                                  |
   2          XX|                                  |
             XXX|                                  |
           XXXXX|                                  |
          XXXXXX|B12                               |
          XXXXXX|B11                               |
        XXXXXXXX|B10   ExtraTask                   |
   1      XXXXXX|                                  |
         XXXXXXX|      ExtraTime                   |
        XXXXXXXX|                                  |
       XXXXXXXXX|                                  |
      XXXXXXXXXX|B9    RevisitHard                 |
   0  XXXXXXXXXX|B8    CompletionOnTime            |
        XXXXXXXX|B7                                |
        XXXXXXXX|B6    RevisitModerate             |
        XXXXXXXX|      CompletionMinTask           |
       XXXXXXXXX|B5                                |
       XXXXXXXXX|B4                                |
  -1   XXXXXXXXX|B3                                |
         XXXXXXX|                                  |
         XXXXXXX|B2                                |
            XXXX|                                  |
            XXXX|                                  |
           XXXXX|B1                                |
  -2         XXX|                                  |
              XX|                                  |
              XX|                                  |
              XX|                                  |
               X|                                  |
               X|                                  |
  -3           X|                                  |
====================================================
Each 'X' represents 79.8 cases
```

Figure 1: Engagement Variable Map for Algebra Data Set

The reliability coefficients of academic engagement instrument on Algebra data set and on Geometry data set measured by Cronbach's alpha are 0.93 and 0.94 respectively, suggesting correlations among 12 temporal-ordered behavioral indicators and 6 session-level behavioral indicators are high. In conventional survey instruments, reliability coefficients of 0.7 and higher are considered to be reliable. This indicates that reliability of engagement found in these two data sets is as good as those found in conventional survey instruments.

It had been perceived that the rank order of problem-level indicators would be off-task, gaming, guessing, on-task, and on-task using appropriate hints, ordered from the lowest level of engagement to the highest level. We checked this hypothesis by reviewing each indicator. Our detailed analysis on each indicator shows that all problem-level classifications appear to be working as expected, except for on-task using appropriate hints. The rank order of off-task (coded as 0), gaming (coded as 1), guessing (coded as 2), and on-task (coded as 3) can be observed by a clear pattern of increasing average engagement scores. Take the indicator B12 for example (see Table 2). The average engagement scores for off-task cohort, gaming cohort, guessing cohort, and on-task cohort are -0.77, 0.23, 0.57, and 1.32, respectively. However, on-task using appropriate hints did not turn out to have a straight-forward interpretation. In terms of average engagement score, the cohort of on-task using appropriate hints was similar to gaming cohort in observations 1 to 3; similar to guessing cohort in observations 5 to 7; and similar to on-task cohort in observations 10 to 12. For this particular example (i.e. B12), this suggests that it might be better off to combine on-task using appropriate hints with on-task.

Table 2: Engagement Indicator for B12 in Algebra Data Set

Score	Count	% of Total	Pt Bis	Avg	SD
0	9673	68.1	-0.68	-0.77	1.05
1	344	2.4	0.06	0.23	0.53
2	936	6.6	0.18	0.57	0.48
3	2861	20.1	0.61	1.32	0.56
4	392	2.8	0.13	0.88	0.64

3.2 Comparison of Engagement

We have demonstrated validity and reliability of academic engagement instrument through empirical evidence. However, whether the instrument is able to compare engagement levels of a cohort working in Algebra problems with a different cohort working in Geometry problems remains unanswered. In attempting to measure the difference in engagement levels between two different cohorts in different learning contexts of ITS, we will need to create exactly the same behavioral engagement indicators in these two data sets.

Figure 2 shows the scatter plot of behavioral indicator estimates of Algebra data set and indicator estimates of Geometry data set, after adjusting difference in average indicator estimates and ratio of standard deviations. The chart shows that all behavioral indicators had similar rank order in both data sets after taking into account of standard error of estimates. It shows all behavioral indicators were falling into confidence interval lines, except for the indicator, *Revisit of hard tasks* (as circled in red) This indicator appears to be requiring significantly more efforts in Geometry data set than in Algebra data set, with indicator estimates of 0.4 logit in Algebra data set and 1.4 logit in Geometry data set. When the indicator of *Revisit of hard tasks* was excluded, the goodness of fit (R^2) had been significantly improved from 0.78 to 0.99. This indicator was not used in equating due to its large difference in engagement estimates.

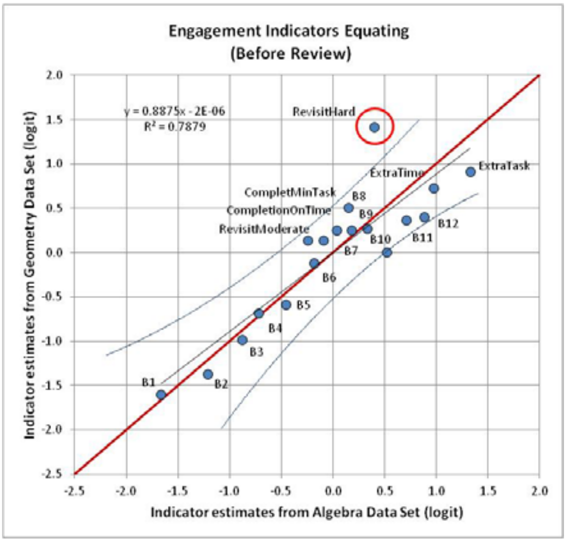


Figure 2: Equating of Engagement Indicators between Algebra Data Set and Geometry Data Set

After applying equating transformation to original engagement scores in Geometry data set, we obtained engagement scores of Geometry data set which can be directly compared to the scores of Algebra data set. Table 3 shows mean and standard deviations of behavioral engagement scores found in Geometry data and Algebra data. The difference in mean engagement between Geometry and Algebra is 0.225 logit, but this difference is not statistically significant (p-value = 0.089), suggesting academic engagement of a cohort working on Geometry tutor was similar to the engagement of the other cohort working on Algebra tutor. The effect size of the difference in average engagement scores is moderate (Cohen's $d = 0.19$).

Table 3: Comparison of Average Engagement between Algebra Data Set and Geometry Data Set

Behavioral Engagement in Geometry			Behavioral Engagement in Algebra		
N	**Mean**	**SD**	**N**	**Mean**	**SD**
59	0.123	0.992	14206	-0.101	1.340

4. CONCLUSION

This paper compared student engagement across domains of learning found in two sets of DataShop data. Our preliminary results did not find any significant difference in behavioral engagement between two different cohorts working on two ITS tutors.

5. ACKNOWLEDGMENTS

This research was supported by Australian Research Council ARC-SRI: Science of Learning Research Centre (project number SR120300015). The views expressed herein are those of the authors and are not necessarily those of the Australian Research Council. We acknowledge the use of 'Assistments Math 2005-2006 (3136 Students)' dataset and 'Geometry Area (1996-97)' dataset accessed via DataShop [12]. We thank Ryan Baker, Zach Pardos, and Neil Heffernan for the permission to use their data in our research and sharing their data through the massive open online course of Big Data in Education.

6. REFERENCES

[1] Appleton, J.J., Christenson, S.L., Kim, D. and Reschly, A.L. (2006) Measuring cognitive and psychological engagement: Validation of the Student Engagement Instrument. *Journal of School Psychology*, vol.44, 427-445.

[2] Arroyo, I. and Woolf, B. (2005), Inferring learning and attitudes from a Bayesian network of log file data, *Proceedings of the Twelfth International Conference on Artificial Intelligence in Education*, IOS Press, pp.33–40.

[3] Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004). Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". *Proceedings of ACM CHI 2004: Computer-Human Interaction*, 383-390.

[4] Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Roll, I. (2006). Generalizing Detection of Gaming the System Across a Tutoring Curriculum. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 402-411.

[5] Baker, R.S.J.d. (2007). Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. *Proceedings of ACM CHI 2007: Computer-Human Interaction*, 1059-1068.

[6] Baker, R.S.J.d., D'Mello, S.K., Rodrigo, M.M.T., and Graesser, A.C. (2010). Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-Affective States during Interactions with Three Different Computer-Based Learning Environments. *International Journal of Human-Computer Studies.* 68, 4, 223-241.

[7] Beck, J.E.(2005). Engagement tracing: Using response times to model student disengagement. In C-K.Looi et al. (Eds). Artificial intelligence in education: supporting learning through intelligent and socially informed technology, 88-95. Amsterdam: IOS Press.

[8] Bomia, L., Beluzo, L., Demeester, D., Elander, K., Johnson, M., & Sheldon, B. (1997). The impact of teaching strategies on intrinsic motivation, Champaign, IL: ERIC Clearinghouse on Elementary and Early Childhood Education. p. 294.

[9] Chapman, E. (2003). Alternative approaches to assessing student engagement rates. *Practical Assessment, Research & Evaluation*, 8(13). Retrieved 7/2/07.

[10] Corbett, A.T. and Anderson, J.R. (1995), Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4, 253-278.

[11] Kember, D., Biggs, J. & Leung, D. Y. P. (2004). Examining the Multidimensionality of Approaches to Learning through the Development of a Revised Version of the Learning Process Questionnaire, *British Journal of Educational Psychology*, 74, 261-279.

[12] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. (2010) A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d.(Eds.) *Handbook of Educational Data Mining*. Boca Raton, FL: CRC Press.

[13] Markwell, D. (2007), A large and liberal education': higher education for the 21st century, Melbourne: Australian Scholarly Publishing & Trinity College, University of Melbourne.

[14] Pardos,Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda,S.M., and Gowda, S.M. (2013), Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes, *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pp. 117-124.

[15] Rowe,J.P., Shores,L.R., Mott,B.W., and Lester, J.C. (2011). Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. *International Journal of Artificial Intelligence in Education*, 21, 115–133.

[16] Schlechty, P. (1994). Increasing Student Engagement. *Missouri Leadership Academy*. p. 5.

[17] Skinner, E. A. and Belmont, M. J. (1993). Motivation in the classroom: Reciprocal effects of teacher behavior and student engagement across the school year. *Journal of Educational Psychology*. 85(4), 571-581.

[18] Willms, J.D. (2003). Student Engagement at School: a sense of belonging and participation: Results from PISA 2000. Organisation for Economic Co-operation and Development. p. i.

[19] Wu, M. L., Adams, R. J., Wilson, M. R., Haldane, S.A. (2007). ACER ConQuest Version 2: Generalised item response modelling software [computer program]. Camberwell: Australian Council *for* Educational Research.

Comparison of Algorithms for Automatically Building Example-Tracing Tutor Models

Rohit Kumar Matthew E. Roy R. Bruce Roberts John I. Makhoul
Raytheon BBN Technologies
Cambridge, MA, USA
{ rkumar, mroy, broberts, makhoul } @ bbn.com

ABSTRACT

In our recent work, we have proposed that multiple behavior demonstrations can be automatically combined to generate an Example-Tracing Tutor model. In this paper, we compare four algorithms for this problem using a number of different metrics for two different datasets, one of which is publicly available. Our experiments show that these four algorithms are complementary to each other in terms of their performance along the different metrics. These findings make a case for incorporating multiple algorithms for building behavior graphs into authoring tools for Intelligent Tutoring Systems (ITS) that use behavior graphs.

Keywords

Tutor Models, Example-Tracing Tutors, Behavior Graphs, Authoring, Automation, Algorithms, Metrics

1. INTRODUCTION

Conventionally, Example-Tracing Tutors [1] are developed in three stages by trained domain experts: (1) User Interface development, (2) Behavior demonstration, (3) Generalization and annotation of the behavior graph. Recently, we proposed [2] that the effort involved in Stage 3 of this process can be significantly reduced by using algorithms that can automatically create a generalized behavior graph from multiple demonstrations.

Automation of tutor model development has been explored in different contexts using completely automated methods as well as augmentation of authoring tools. Barnes and Stamper [3] proposed a method that uses existing student solutions to generate hint messages for the Logic Proof tutor. Recently, Eagle et al. [4] have used clustering of interaction network states as an approach to the same problem. In the context of knowledge-tracing and example-tracing tutors, McLaren et al. [5] proposed the use of activity logs from novice users to bootstrap tutor model development. They developed software tools that integrate access to novice activity logs with tutor authoring tools.

In the next section, we briefly outline four algorithms for automatically generating behaviors graphs. In Section 3, we will present experiments using two datasets, to compare these algorithms along a number of metrics that measure desirable characteristics of tutor models.

2. ALGORITHMS

2.1 Behavior Graphs and Demonstrations

Behavior graphs are directed graphs. The nodes in a graph correspond to valid solution states. Non-terminal nodes represent partial solutions. Edges in the graph represent events, some of which are correct and lead to the next state while others are incorrect and lead back to the same state. Edges are annotated with the conditions that an event must meet to traverse the edge. Behavior graphs may also include unordered groups. As the name suggests, states within an unordered group may be traversed in any order. Constituents of the behavior graph (i.e. nodes, edges, groups) may be associated with a number of annotations based on the educational application.

On the other hand, behavior demonstrations are captured as a sequence of user interface (UI) events. Each event is represented as a 2-tuple $e_i = (u_i, \mathbf{d}_i)$ that includes an identifier u_i of the UI element and data $\mathbf{d}_i$ associated with the event. Note that each behavior demonstration implicitly represents a behavior graph where the nodes in the graph correspond to the state of completion of each event in the demonstration. Such a behavior graph does not generalize to learner behaviors beyond those that are exactly identical to the demonstration. Automatic Behavior Graph Generation (ABGG) algorithms utilize multiple demonstrations of solutions of a problem to generate a behavior graph that can serve as a tutor model for the problem.

2.2 Algorithm 1: Interaction Network

The baseline algorithm used in our work combines the individual behavior graph corresponding to available demonstrations by merging identical nodes and edges in a sequential order. When a non-identical edge is found, a new branch is created in the graph. The resulting behavior graph is an interaction network which has been used in prior work [4] [6]. All paths in the behavior graph generated by this algorithm are assumed to be correct paths i.e. this algorithm is incapable distinguishing between correct and incorrect actions by the learner. While the behavior graph generated by this algorithm is more general than any individual demonstration used to create the graph, no unseen paths are generated. Furthermore, the number of nodes and edges created by this algorithm is fairly large, which makes the annotation of such graphs difficult for problems with many UI elements.

2.3 Algorithm 2: Heuristic Alignment

Our next algorithm, shown in Table 1, utilizes two characteristics of behavior demonstrations. First, if two or more events in a demonstration have the same element identifier u_i, the latter event likely corresponds to a correction of the data value input in the former events. Second, if we assume that there is one and only one correct solution sequence through the UI elements, we can transform the problem of generalizing behavior demonstrations to that of finding the optimal sequence of states through the UI elements.

This research was funded by the US Office of Naval Research (ONR) contract N00014-12-C-0535.

Table 1. Algorithm 2 (Heuristic Alignment)

```
Stage 1. Compute Retracted Demonstrations
• For each demonstration D
    • For each retracted event e = (u, d)
        1. $e_{target}$ = last event in D s.t. $e_{target} \rightarrow u = e \rightarrow u$
        2. Add $e \rightarrow \mathbf{d}$ to $e_{target}.\mathbf{d}_{wrong}$
        3. Remove e from D
Stage 2. Find Sequence of States
• For each unique identifier u
    1. $\mathbf{p}_u$ = set of positional indices of events s.t. identifier = u
    2. $mode_u = mode(\mathbf{p}_u)$
• Sequence states ($s_u$) corresponding to each element
  identifier (u) in increasing order of their $mode_u$
Stage 3. Generate Edges
• For each state $s_{u*}$
    1. Generate correct edge for each unique d for events s.t.
       identifier = u*
    2. Generate wrong edge for each unique entry in $\mathbf{d}_{wrong}$ for
       events s.t. identifier = u*
Stage 4. Identify Unordered Groups
• For each pair of adjacent states ($s_{u1}$,$s_{u2}$)
    1. if $|\cap(\mathbf{p}_{u1},\mathbf{p}_{u2})| > \sqrt{|demonstrations|}$, group $s_{u1}$, $s_{u2}$
```

2.4 Algorithm 3: Center Star Alignment

Note that Stage 2 of the previous algorithm is, in effect, aligning the multiple demonstrations. The Center Star Algorithm can be used to perform this alignment. Algorithm 3 uses the Center-Star Alignment between the retracted demonstrations. Similar to algorithm 2, a new state is generated for each position in the aligned demonstrations. However, since we obtain the alignment using the Center Star algorithm, the second assumption made by algorithm 2 is not necessary, which can lead to multiple states with the same element identifiers. This allows algorithm 3 to generate alternate paths.

2.5 Algorithm 4: Combining Multiple Paths

Algorithm 4 considers ABGG as the process of finding multiple paths in a directed graph. A first order transition matrix obtained from the demonstrations represents a directed graph. Specifically, the longest (non-repeating) path in this directed graph is the most likely path through the UI elements based on the demonstrations. While the problem of finding longest paths in general graphs is known to be NP-hard, in our approach, we employ an exponential time longest path finding algorithm within bounds of the number of UI elements and uses a transformed transition matrix to find multiple shortest paths. The transform changes the weight of each valid edge of the directed graph to row normalized inverse. We merge all the paths found to we construct a behavior graph similar to the process of constructing an interaction network. The algorithm uses Stage 1 and Stage 4 of algorithm 1.

2.6 Discussion

As mentioned earlier, incremental addition of demonstrations to generate interaction networks does not identify incorrect input data values. Using the assumption about retracted events, the other three algorithms are able to identify incorrect inputs. Johnson et al. [6] used a similar assumption in their work on reducing the visual complexity of interaction networks. We notice that the algorithms 2 and 3 are complementary in terms of their ability to find alternate paths and unordered groups. Algorithm 4 on the other hand offers both of these abilities. In the next section, we will discuss the performance of all of these algorithms in terms of quantitative metrics

None of the algorithms discussed in this paper are capable of discovering data values beyond those seen in the training demonstrations. This type of generative ability is particularly useful for learning tasks, such as language learning, where a large number of different inputs may be expected from the learners. In our ongoing work, we want explore the use of grammar induction techniques to learn regular expressions from correct and incorrect data values for each state.

3. EVALUATION

3.1 Datasets

We use two collections of behavior demonstrations/traces to evaluate the performance of the four algorithms described earlier. The first dataset (referred to as the *BBN dataset*) comprises of five physics problems. Nine subjects spent upto one hour each to create demonstrations of the five problems. All nine subjects were able to complete demonstrations of three problems. Six subjects completed the fourth problem and only four completed the fifth problem. Additionally, we used three Assistments datasets accessed via DataShop [7] to form our second collection of behavior demonstrations. This publicly shared large dataset comprises a total of 683197 traces and 1905672 events for 3140 problems. We filtered these datasets to use only problems that had six or more traces and had at least two UI elements.

3.2 Metrics

Metrics used in our evaluation are discussed in detail in our prior publication [2]. These metrics are categorized by the desirable characteristics of automatically generated behavior graphs they measure.

- **Readability/Maintainability:** The conciseness of a graph can be measured using the number of nodes and edges in the graph. Compression ratio measures the rate at which an algorithm is able to reduce demonstration events into behavior states (i.e. nodes) by finding similarities between events.
- **Completeness:** We use the rate of unseen events in held out demonstrations as a metric to measure the completeness of our automatically generated behavior graphs.
- **Accuracy:** Edge accuracy measures the percentage of Correct & Incorrect edges that were accurately classified by the algorithm. Error rate is a frequency weighted combination of edge accuracy that measures the fraction of learner events that will be inaccurately classified by the automatically generated behavior graph.
- **Robustness:** Branching factor is the average number of data values available at each UI element. A large branching factor indicates the capability to process a large variety of learner inputs at each state. Also, the number of unordered groups and the size of unordered groups are indicative of flexibility a graph affords to learners to explore the solution paths of a problem.

3.3 Experimental Design

We use two different experimental designs for the two datasets. Since the BBN dataset is comprised of a small number of demonstrations per problem, we use all available demonstrations for training and report only the metrics that can be derived from the graphs and the training demonstrations. Since a large number of traces are available for the problems in the Assistments dataset,

Table 2. Averaged metrics for the graphs generated for the problems in the BBN & Assistments (Math) dataset

*indicates significant ($p < 0.05$) difference with other algorithms for the same dataset

	Algorithm 1		Algorithm 2		Algorithm 3		Algorithm 4	
Metrics ▼	**BBN**	**Math**	**BBN**	**Math**	**BBN**	**Math**	**BBN**	**Math**
#Nodes	144.8	79.1*	**32.4**	**5.4**	37.0	6.0	**32.4**	6.6
#Correct Edges	160.4	147.9*	**70.0**	**12.8**	97.0	18.3	71.4	17.5
#Incorrect Edges			17.2	24.2*	19.8	33.4*	**5.0**	**19.5***
Compression Ratio	1.8	6.7*	**7.3**	**77.3***	6.3	66.8*	**7.3**	60.2*
% Accurate Correct Edges	76.7	39.1*	77.0	42.2	66.2	42.6	**82.8**	**44.1***
% Accurate Incorrect Edges			99.5	**99.9***	99.5	97.5*	**100.0**	99.5*
Training Error Rate	15.5	51.3*	**7.6**	25.2*	13.0	17.7	**7.6**	**17.4**
Heldout Error Rate		42.7*		23.4*		16.0		**15.6**
% Training Unseen Events	**0.0**	**0.0**	**0.0**	10.5*	4.4	2.2*	8.1	6.7*
% Heldout Unseen Events		**10.1***		19.0*		11.5*		13.8*
Branching Factor	1.2	2.2*	3.1	11.1*	**3.4**	**12.6***	2.7	8.5*
#Groups			1.6	**0.5***			**2.6**	0.02*
Avg. Group Size			**3.3**	**1.8***			2.8	0.04
% Group Coverage			17.7	**30.6***			**31.3**	0.5

we use a three-fold cross validation design to split the available traces into three different training and held out sets. Reported metrics are averaged over each split.

3.4 Results

3.4.1 BBN Dataset

Table 2 shows performance results for the four algorithms on the two datasets. As expected, the interaction networks comprise of a large number of nodes and edges that lead them to have significantly ($p<0.01$) lower compression ratio. Algorithms 2 (Heuristic Alignment) and 4 (Multiple Paths) are able to achieve the highest compression consistently for all five problems.

On the accuracy metrics, Algorithm 4 outperforms the other algorithms on average. However, it is significantly better ($p<0.001$) than Algorithm 1 and 3 on the incorrect edge accuracy metric. Furthermore, the high accuracy for incorrect edges for two of the three algorithms that use the retracted demonstrations partly validates the underlying assumption made by these algorithms.

In contrast to the accuracy metrics, alignment based algorithms (2 and 3) outperform the multiple paths algorithm (4) on achieving a higher branching factor. The frequency based pruning underlying the selection of multiple paths in algorithm 4 leads to the elimination of certain novel edges. Based on the performance of these algorithms on the edge accuracy metrics we see many of these novel edges are likely to be inaccurate due to limited evidence for their classification in the training demonstrations. While the algorithms complement each other, Algorithm 4 seems to be a potential candidate for optimal tradeoff between the different metrics.

In terms of metrics based on unordered groups in a graph, we find that algorithm 4 leads to a larger fraction of nodes (31%) to be included in unordered groups. Finally, we see that pruning significantly degrades the performance of Algorithm 4 on percentage of unseen events i.e. completeness. Since interaction networks losslessly embed all events observed in the training demonstration, their performance on this metric is guaranteed to be flawless. In the next section, we will compare this result to their performance on held out demonstration sequences.

3.4.2 Assistments Dataset

The performance of the algorithms on the Assistments (Math) dataset is also shown in Table 2. Largely, the results on this dataset agree with the results on the BBN dataset. Algorithm 2 (Heuristic Alignment) outperforms all other algorithms on three of the readability metrics. Unlike the BBN dataset, the average compression ratio for Algorithm 2 is significantly better than the other algorithms including Algorithm 4 (Multiple Paths).

Algorithm 4 significantly outperforms the other algorithms on three of the accuracy metrics. Because of their lossless nature, Interaction Networks (Algorithm 1) performs the best on Completeness metrics (% unseen events) as was the case with the BBN dataset. However, we find evidence of over-fitting of the algorithms to training data on this metric as indicated by the approximately 9% higher rate of unseen events for held out demonstrations for all the algorithms.

While the results on the branching factor metrics of the Assistments dataset are consistent with the BBN dataset, Algorithm 2 outperforms Algorithm 4 on the metrics based on the unordered groups. Because Algorithm 2 identifies unordered groups that are larger in size than Algorithm 4, the groups found by the Heuristic Alignment algorithm have a higher coverage of the generated graphs, especially in the Assistments datasets where the number of UI elements is relatively small.

Figure 1 further explores the tradeoff between the key metrics for larger number of traces (i.e., more training data) in Figure 1a and increasingly complex problems (i.e., more UI elements) in Figure 1b. Algorithm 1 does not scale well on readability metrics (Compression Ratio). The algorithms demonstrate stability in accuracy and completeness performance with increasing problem or data complexity. Algorithms 3 and 4 can produce a consistently low error rate despite increasing complexity. The rate of unseen events reduces by over 60% (relative) for a 10-fold increase in training data. This is also

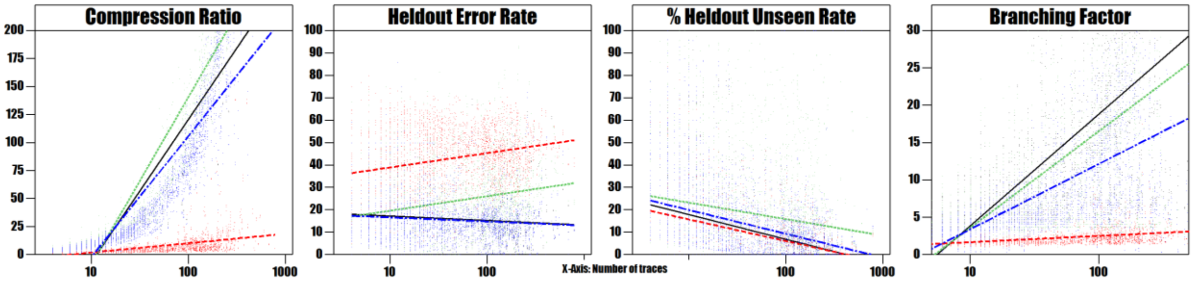


Figure 1a. Algorithm performance for different number of training traces

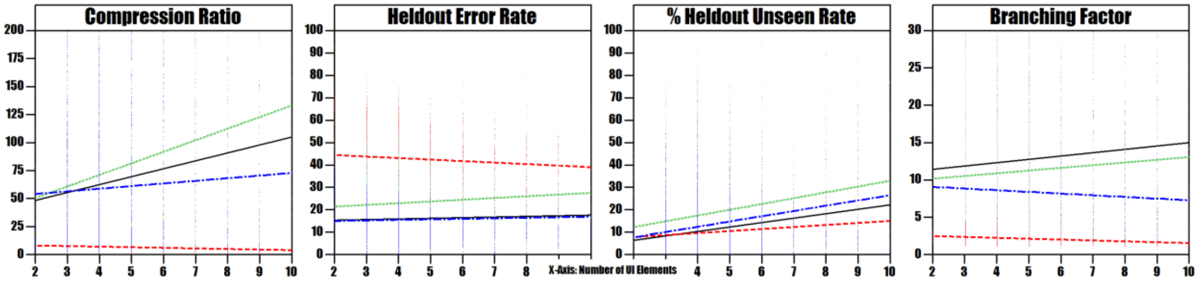


Figure 1b. Algorithm performance for different number of UI elements in a problem

evidenced in the BBN dataset if we compare problems 1, 2 and 6 which have more data than problems 10 and 15. Finally, as is often the case with data-driven approaches, model robustness is dramatically improved with the use of more data.

4. CONCLUSIONS

In this paper, we have presented four algorithms for automatically building example-tracing tutor models using multiple solution demonstrations that may be crowd-sourced or collected from a sample of users in an online ITS. The transfer of this effort from the ITS developers to a low cost (potentially no cost) workforce affords scale to the ITS development process.

Foremost, we must note that due to the inaccuracies in the automatically generated behavior graphs, they need manual inspection and further annotation before they can be operationalized. In our work on creating a general purpose learning platform focused on STEM domains, we are integrating these algorithms into our suite of authoring tools to allow ITS developers to use these algorithms in their workflow. Second, we notice that the algorithms have complementary performance on the different desirable characteristics of the automatically generated behavior graphs. Based on Table 2, we would choose Algorithm 2 for its Readability metrics, Algorithm 4 for Accuracy, Algorithm 1 for Completeness and Algorithm 3 for the key Robustness metric. All of these algorithms should be made available to the ITS developers through the authoring tools. We think that Algorithm 2 may be used as the default choice.

Looking ahead, the pursuit of automation of example tracing tutor modeling has a number of challenges of interest. The complementary nature of these algorithms suggests the potential for combining them to obtain better behavior graphs. Extension of the techniques presented in this paper to automatically update existing behavior graphs, which may have been manually authored, using traces from actual learners can help in maintenance and online improvement of the tutor models.

5. REFERENCES

[1] Aleven, V., Mclaren, B. M., Sewall, J., and Koedinger. K. R. 2009. A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. *Int. J. Artif. Intell. Ed.* 19, 2 (April 2009), 105-154.

[2] Kumar, R., Roy, M.E, Roberts, R.B., and Makhoul, J.I. 2014. Towards Automatically Building Tutor Models Using Multiple Behavior Demonstrations. *12th Intl. Conf. on Intelligent Tutoring Systems* (ITS 2014), Honolulu, HI.

[3] Barnes, T., and Stamper, J. 2008. Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. *9th International Conference on Intelligent Tutoring Systems* (ITS 2008). Montreal, Canada.

[4] Eagle, M., Johnson, J., and Barnes, T., 2012. Interaction Networks: Generating High Level Hints Based on Network Community Clusterings, *5th International Conference on Educational Data Mining* (EDM 2012). Chania, Greece.

[5] McLaren, B.M., Koedinger, K.R., Schneider, M., Harrer, A., and Bollen, L. 2004. Bootstrapping Novice Data: Semi-Automated Tutor Authoring Using Student Log Files. *Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes, 7th International Conference on Intelligent Tutoring Systems* (ITS 2004). Alagoas, Brazil.

[6] Johnson, M., Eagle, M., Stamper, J., and Barnes, T. 2013. An Algorithm for Reducing the Complexity of Interaction Networks, *6thInternational Conference on Educational Data Mining* (EDM 2013). Memphis, TN.

[7] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., and Stamper, J. 2010. A Data Repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining*. Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.). Boca Raton, FL: CRC Press

Computer-based Adaptive Speed Tests

Daniel Bengs[†]

[†]German Inst. for Intern. Educational Research
Information Center for Education
Frankfurt am Main, Germany
bengs@dipf.de

Ulf Brefeld[†‡]

[‡]Technische Universität Darmstadt
Department of Computer Science
Darmstadt, Germany
brefeld@kma.informatik.tu-darmstadt.de

ABSTRACT

The assessment of a person's traits is a fundamental problem in human sciences. Compared to traditional paper & pencil tests, computer based assessments not only facilitate data acquisition and processing but also allow for adaptive and personalized tests so that competency levels are assessed with fewer items. We focus on speeded tests and propose a mathematically sound framework in which latent competency skills are represented by belief distributions on compact intervals. Our algorithm updates belief based on directional feedback; adaptation rate and difficulty of the task at hand can be controlled by user-defined parameters. We provide a rigorous theoretical analysis of our approach and report on empirical results on simulated and real world data, including concentration tests and the assessment of reading skills.

1. INTRODUCTION

The assessment of a person's traits such as ability is a fundamental problem in the human sciences. Perhaps the most prominent examples are the triennial PISA studies launched by the OECD in 1997. Traditionally, assessments have been conducted with printed forms that had to be filled in by the testees, so called paper & pencil tests. Nowadays, computers and handhelds become more and more popular as platforms for conducting studies in social sciences; electronic devices not only facilitate data acquisition and processing, but also allow for real-time adaptivity and personalization.

Psychological testing differentiates between two types of tests, namely *power* and *speeded* tests [2]. The former uses items with a wide range of difficulty levels, so that testees will almost surely be unable to solve all items, even when given unlimited time. By contrast, speeded tests deploy homogeneous items that are easy to solve, and testees are discriminated by the time needed to solve the items. In this paper, we focus on pure speed tests akin to [4] as well as tests where response times are assessed together with item correctness, e.g. to study the efficiency of cognitive processes [5].

We devise an algorithm using a data-driven approach for steering the time limits of individual items actively. Items of constant inherent difficulty are administered in a sequence $t = 1, 2, \ldots$, and a limit on response time $\hat{\tau}_t$ is adapted based on testee performance. After the administration of each item, the algorithm chooses the limit for the upcoming item such that as much information as possible on testee's expected response time is collected. The uncertainty of an estimate $\hat{\tau}$ is represented by a belief distribution over a finite interval of admissible response times. When administering item t, an estimate $\hat{\tau}_t$ is drawn, such that $\hat{\tau}_t$ divides the belief mass in two parts whose areas have a predefined ratio roughly corresponding to the odds that the testee responds within the time limit. After the testee attempts solving the item under the time limit $\hat{\tau}_t$, the algorithm receives feedback ϕ_t encoding three cases: (i) if $\hat{\tau}_t - \tau_t < \epsilon$, the time limit $\hat{\tau}_t$ was insufficient for the testee to answer in time and $\phi_t = 1$, (ii) in case $\hat{\tau}_t - \tau_t > \epsilon$, the setting was more than sufficiently long, and $\phi_t = -1$, and (iii) $\tau_t \in [\hat{\tau}_t - \epsilon, \hat{\tau}_t + \epsilon]$ which corresponds to a just right setting and $\phi_t = 0$.

Our learning algorithm is around the following strategy: Once we observe that $\hat{\tau}$ is too small to allow for solving the item, it is highly probable that all time limits $\tilde{\tau} > \hat{\tau}$ would also be too small, and belief in their correctness can be updated. A similar argument holds vice versa for time limits more than sufficiently long. The feedback is therefore used as a directional signal that triggers the update process. In this paper we develop the mathematical framework for computer-based adaptive speed tests and devise an efficient algorithm. We provide a theoretical analysis and report on empirical results using artificial and real-world data.

2. RELATED WORK

Missura & Gärtner [3] consider the problem of dynamic difficulty adjustment as a game between a master and a player that is played in rounds $t = 1, 2, \cdots$, where the master predicts the difficulty setting for the next round based on the player feedback. The authors introduce an algorithm that represents the set of admissible difficulty settings as a finite discrete set $\mathcal{K}$ endowed with a partial ordering. For each of the difficulty levels $k \in \mathcal{K}$, the algorithm maintains a positive number representing belief in k being *just right.* At each round, the prediction allows to update the maximal amount of belief after feedback has been received. In contrast to [3], we use a continuous framework and do not rely on a predefined discrete set admissible settings, but instead find appropriate settings adaptively on the fly.

Csáji and Weyer [1] investigate the problem of estimating a constant based on noisy measurements of a binary sensor with adjustable threshold. That is, of a constant $\theta^* \in \mathbb{R}$ disturbed by additive, i.i.d. noise N_t, only measurements indicating whether the $\theta^* + N_t$ exceeds an adjustable threshold θ_t are available for $t = 1, 2, \cdots$. Under mild assumptions on the distribution of N_t, a strongly consistent estimator is derived, i.e. a method for choosing the thresholds θ_t such that $\theta_t \to \theta^*$ almost surely for any starting value θ_0. In contrast to [1], we do not make any assumptions on the distribution of the value to be estimated or on its stationarity.

In the field of psychometrics, only a few adaptive speed tests have been designed. For the assessment of concentration ability, Goldhammer & Moosbrugger [4] propose the Frankfurt Adaptive Concentration Test II (FACT-II), which conceptualizes concentration as the ability to respond to stimuli in the presence of distractors. After administration of item t, exposure time of the item $t+1$ is adjusted until a liminal exposure time is reached that just allows the testee to solve the task. Starting with a fixed initial exposure time θ_1, updating is performed multiplicatively depending on whether a correct response is given in time or not. Tests using both accuracy and response times are used to assess efficiency of cognitive processes for instance in the measurement of components of reading abilities [5].

3. CAT-FRAMEWORK FOR SPEED TESTS

We consider a computerised adaptive test where a sequence of items of homogeneous difficulty is presented to the testee and response times are recorded. This scenario encompasses adaptive speeded tests such as FACT2 [4] as well as tests targetting efficiency of congitive processes (e.g. [5]). In the former, response times are limited by an adaptation mechanism and relate directly to the trait being assessed. In the latter case, response times are merely observed and used to analyse testee efficiency. Here, testees might take a long time to think and then score perfectly, leading to undesirable ceiling effects, as observed by [5]. Imposing a time limit may increase testing efficiency and also increase variation in item correctness, leading to a higher data quality. We additionally show that our algorithm can be configured to realize a user-defined probability for a timely response.

3.1 Methodology

We consider admissable response times in an interval $T = [a, b]$. We assume that at each position of the testing sequence, there exists a lower bound on the testee's response time which we consider the just right setting $\tau_t \in T$. This is the minimum sustainable response time enabling the testee to solve the item; we assume that it relates to an underlying trait but is independent of the actual item, as the item bank consists of items of constant difficulty. The goal of the adaptation is to iteratively adjust the time limit until the just right setting is reached. To this end, the algorithm maintains a belief distribution $B_t : [a, b] \to (0, \infty)$ on T that is used for accumulating knowledge about the correctness of previously estimated time limits. Correctness of the predictions is assessed after administering each item by feedback ϕ_t, which is based on the relation of the testee's response time τ_t and the predicted time limit $\hat{\tau}_t$: We have $\phi_t = -1$ if $\hat{\tau}_t < \tau_t$, that is, the item is solved within the time limit, $\phi_t = 1$ if the testee runs out of time ($\hat{\tau}_t > \tau_t$), and $\phi_t = 0$ if the item is solved (exactly) at the time limit ($\hat{\tau}_t = \tau_t \pm \epsilon$).

Here, $\epsilon > 0$ is used to decide whether the τ_t is close enough to $\hat{\tau}_t$ to consider $\hat{\tau}_t$ a correct prediction. This is necessary because response times underly random fluctuations and thus the just right time limit remains hidden to the algorithm. Adaptation and prediction is done using the belief function and two preassigned parameters $\beta \in (0, 1)$ and $\delta \in (0, 1)$ as follows: Belief is initialized to be a strictly positive constant on T.[*] The time limit for administering item t is computed as the value $\hat{\tau}_t$ that splits the area under B_t in two parts $P_t(\hat{\tau}_t) := \int_a^{\hat{\tau}_t} B_t(x)dx$ and $Q_t(\hat{\tau}_t) := \int_{\hat{\tau}_t}^b B_t(x)dx$, such that $P_t : Q_t = \delta : 1 - \delta$. Assuming normalized belief, this can be achieved by determining $\hat{\tau}_t$ that satisfies $P_t = \delta$.

It is easy to see that B_t being non-negative by assumption, the mapping $\hat{\tau}_t \mapsto P_t(\hat{\tau}_t)$ is strictly increasing and thus bijective, so $\hat{\tau}_t$ is uniquely determined if only $\int_a^b B_t(x)dx \neq 0$, which because as $B_1 \neq 0$ and $\beta \neq 0$, all $B_t \neq 0$ due to the updating formula given below. After the testee attempts to solve the item given time limit $\hat{\tau}_t$, the algorithm receives feedback ϕ_t indicating whether the time limit was (i) *too long* and $\phi_t = -1$, or (ii) *too short* and $\phi_t = +1$. Because of transitivity, the algorithm may infer that (i) the time was more than sufficiently long or (ii) any shorter time limit would also have been insufficient for the testee. If the testee responded ϵ-close to the time limit and $\phi_t = 0$, no update is necessary because current belief produced a correct prediction. Otherwise, the belief in all settings (i) longer or (ii) shorter, respectively, is lowered by the updating step, which is carried out by multiplying the respective belief values by the learning rate β:

$$B_{t+1}(x) = \begin{cases} \beta B_t(x), & \text{if (i) and } x \geq \hat{\tau}_T \text{ or (ii) and } x \leq \hat{\tau}_t \\ B_t(x), & \text{else.} \end{cases}$$

The parameter β thus controls how much weight is given to information from the current observation; the closer β is to zero, the faster the adaptation. If β is close to 1, the predictions will show less variation. Thus assumptions on the rate of change of the true time limit and the length of the item sequence can be used to guide the choice of β. We give a theoretical analysis yielding bounds on the difference of successive predictions by our algorithm in Theorem 1.

3.2 Computational Aspects

Each feedback step leads to the updating of either the interval $[a, \hat{\tau}_t]$ or the interval $[\hat{\tau}_t, b]$ by multiplying the values of B_t by β. Consequently, for all t, the function B_t belongs to the space of non-negative step functions on $[a, b]$. This allows for efficient storage, manipulation and prediction based on an interval subdivision scheme. Starting with $T = [a, b]$, we divide the interval containing the current prediction $\hat{\tau}_t$ at $\hat{\tau}_t$ and update the belief values to the left or right of $\hat{\tau}_t$ depending on the feedback ϕ_t by multiplying with $\beta \in (0, 1)$. Formally, we write B_t as a sum

$$B_t = \sum_{i=1}^{N_t} y_i^{(t)} \chi_{I_i^{(t)}}$$

[*] The initial belief function B_1 can also be tailored to incorporate prior knowledge about where to expect τ_1.

for some $N \in \mathbb{N}$, where $y_i^{(t)} \geq 0$ is the value B_t takes on the i^{th} interval given by $I_i^{(t)} = [x_{i-1}^{(t)}, x_i^{(t)})$ for $i = 1, \cdots, N_t - 1$ and $I_{N_t}^{(t)} = [x_{N_t-1}, x_{N_t}]$. The interval endpoints are defined by a partition

$$a = x_0^{(t)} < x_1^{(t)} < x_2^{(t)} < \cdots < x_{N_t}^{(t)} = b$$

of $[a, b]$. By i_t^* we denote the index of the interval containing $\hat{\tau}_t$. If $\phi_t = 1$, we set

$$\begin{aligned} B_{t+1} = & \sum_{i=1}^{i_t^*-1} \beta y_i \chi_{I_i^{(t)}} + \beta y_{i_t^*} \chi_{[x_{i_t^*-1}, \hat{\tau}_t)} \\ & + y_{i_t^*} \chi_{[\hat{\tau}_t, x_{i_t^*})} + \sum_{i=i_t^*+1}^{N_t} y_i \chi_{I_i^{(t)}}, \end{aligned} \tag{1}$$

if $\phi_t = -1$, belief at $t+1$ is defined analogously, that is, nodes $x^{(t+1)}$ are as above, but the weights with indexes greater or equal than i_t^* are multiplied by β. Finally, if $\phi_t = 0$ no update is necessary and $B_{t+1} = B_t$. The belief function can be stored and updated efficiently by storing the endpoints $x_1^{(t)}, \cdots, x_{N_t-1}^{(t)}$ and function values $y_1, \cdots, y_N$. Theorem 1 bounds the minimal and maximal difference between successive estimates of the algorithm.

THEOREM 1. *Let $(\hat{\tau}_t)_{t=1}^{N}$ be a sequence of estimations of the CAST algorithm with parameters β and δ. Then for $t = 1, \cdots, N-1$ it holds that*

$$\frac{\delta(1-\delta)(1-\beta)}{\max_{x\in[a,b]} B_t(x)} B \leq |\hat{\tau}_{t+1} - \hat{\tau}_t| \leq \frac{\delta(1-\delta)(1-\beta)}{\min_{x\in[a,b]} B_t(x)} B,$$

where $B = \int_a^b B_t(x)dx$.

Note that the bounds are invariant under rescaling of the belief function, but depend on the parameter β that controls learning rate: If β is small, then new experience is given more weight and the lower bound on step size is greater than its analogue for $\beta \approx 1$ which gives less weight to new information. The dependance on δ can be interpreted as follows: The more δ deviates from 0.5, the more will inital time limits be biased towards a or b resp. and also adaptation to the observed time limit will be slower. Therefore, δ can be regarded a parameter controlling difficulty bias. Our experiments demonstrate that by varying δ, a wide range of difficulty settings can be realized. We verify this claim in the next Section.

4. EMPIRICAL RESULTS

4.1 Artificial Data

To showcase the adaptivity of our approach, we simulate near-realistic scenarios to create settings that reflect behaviour observed in adaptive psychological speed tests or computer games. We compare the empirical performance of CAST to state-of-the-art baselines POSM [3], Csáji-Weyer-Iteration (CWI) [1], and the algorithm used by FACT-II [4].

Throughout this suite of experiments, we use $T = [0, 1]$. To allow for a fair comparison, the set of difficulty settings for POSM consists of N equidistantly sampled points in T, where n is the number of time steps used. This choice guarantees that the number of subdivisions made by CAST is less than or equal to the number of settings available to POSM.

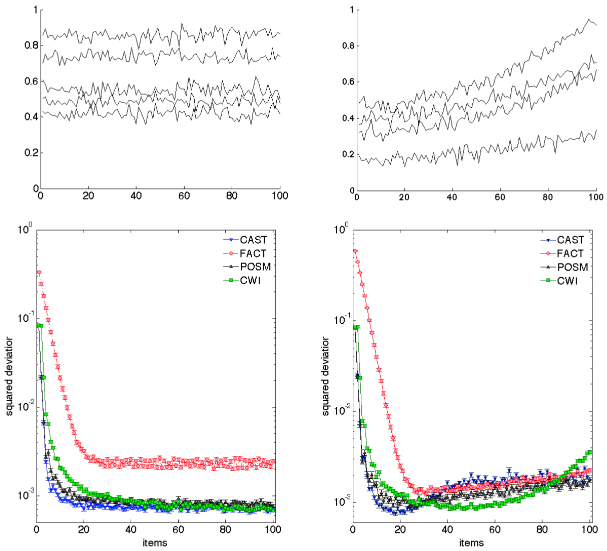


Figure 1: Top: Artificial response times. Bottom: Results for constant (left) and drift (right) scenarios.

Thus, all approaches have access to the same amount of resources. We use optimal parameters for CAST and POSM chosen by model selection.

We study the behavior of the algorithms in constant and dynamic scenarios: In the first setting, the ground-truth τ remains constant. We sample the constants from a uniform distribution on T. In the second setting, we simulate learning and tiredness effects of testees. The true parameter τ thus underlies drifts and the resulting distribution is not stationary. In both settings, simulated response times are additionally disturbed by white noise. Figure 1 (top row) shows sample observations for the two scenarios. We report on average deviations of 1,000 repetitions with randomly generated τ.

Figure 1 (bottom, left) shows the results for the constant setting. All algorithms need some time to adapt to the noisy τ with FACT showing severe problems in the estimation process and finally oscillating between two estimations that are both far away from the simulated ground-truth. The best adaptation is achieved by CAST in terms of speed as well as overall performance. CWI converges to a comparable estimator at the end of the sequence but the adaptation process is not as fast. POSM performs only slightly worse than CAST. The visual differences are reflected in Table 1 that summarizes the results.

Figure 1 (bottom, right) shows the results for the dynamic scenario containing drift. Again FACT is significantly outperformed by the competitors. CWI describes a U-shaped curve and proves not appropriate for dynamic scenarios due to the strict assumptions on the data generating distribution. By contrast, CAST converges quickly to the initial plateau after about 20 responses and looses accuracy when the drift begins to dominate the scenario. POSM takes again more time to adapt to the data but shows a slightly improved performance for intermediate items which also leads to the smallest difference in Table 1. However, note that

Table 1: Sum of squared deviations of Figure 1

	CAST	POSM	FACT	CWI
constant	1.8752	2.0427	14.5896	2.9801
drift	2.6661	2.4396	24.5116	3.4407

we tailored the discrete POSM to the continuous scenario to obtain a fair comparison in terms of computational resources. In real world settings, the optimal discretization of POSM is not obvious and often intractable. CAST can thus be seen as the best off-the-shelf approach although POSM achieved slightly better scores in the dynamic scenario.

4.2 Reading Skills

In this section we evaluate our algorithm in an experiment using real world data from a computerized test of phonological representation by Richter et al. [5]. Testees listen to an auditorial reference stimulus in form of a pseudo word. The presentation is immediately followed by a displayed pseudo word on the screen. The testee's task is to decide whether the displayed word is phonologically identical to the auditory one. No time limit is enforced.

The data consists of response times of 528 children, between five and 11 years old, assessed during a test comprising of $n = 64$ items. We simulate the effects of incorporating a time limit by our algorithm as follows: After preprocessing by removing extreme response times (>2500ms) and compensating the strong linear relationship between number of syllables and mean response time ($R^2 = .83$) to level item difficulty, the linearly transformed response times are between -932.34 and 2267 ms. We use our algorithm to predict expected response time for each participant on the interval $[-1000, 2500]$.

Note that without time limits, ceiling effects in accuracy may be observed [5] while too tight limits on response time can easily lead to frustrated participants. We focus on the proportion of items each participant would not have answered in time for different values of β and δ. The goal is to predict for each participant time limits on each item, such that a non-zero chance of solving the respective item is realized. We compute predictions $\hat{\tau}_i; i = 1, \cdots, 64$ for each participant and analyse the proportion P of items not solved within the predicted time limit and compare the results with the proportion achieved by using percentiles of the testee's response times. We use $\epsilon = 10$ms.

Figure 2 (top, left) shows the distributions of P across participants for $\beta = 0.65$ and $0.05 \leq \delta \leq 0.95$. The figure indicates that proportions P between 20% and 65% can be robustly realized by using different values of difficulty bias δ. By contrast, the proportions realized by a percentile-based approach in Figure 2 (top, right) span a broader range but contain much variance across the participants, showing that our adaptive approach leads to a more homogeneous experience across testees. For our algorithm, dispersion measured by range is at roughly 20 percentage points across all δ while for percentiles, ranges between 20 and 70 percentage points are observed.

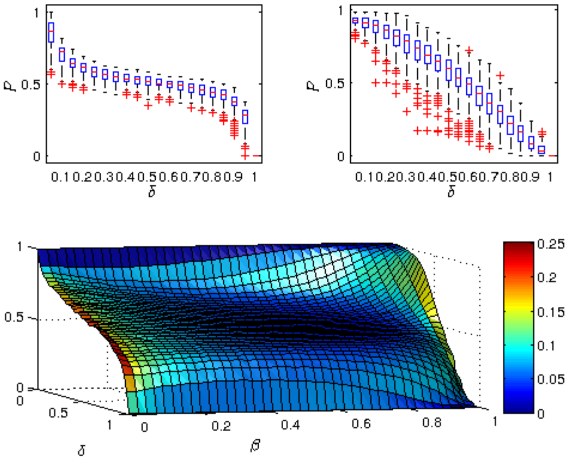


Figure 2: Results for the reading skill experiment.

Figure 2 (bottom) shows mean proportions P of testees not responding in time on the z-axis while the color corresponds to the standard deviation at every point. The figure shows that a low dispersion and a wide range of proportions can be set with our algorithm also when the β parameter is varied; mean proportions are stable for all but extreme values of both parameters. In sum, our algorithm effectively controls the adaptation in both difficulty bias and adaptation rate.

5. CONCLUSION

We introduced a novel technique for computer-based adaptive speed tests. In contrast to existing methods, our approach is devised from a mathematically sound framework and maintains belief distributions on compact intervals to represent estimates of the unknown parameter. In addition, our approach is purely data-driven and does not rely on assumptions on the distribution of the true parameter. Empirically, we showed the effectiveness of our adaptive speed test on artificial and real world scenarios.

Acknowledgements

We are grateful to Johannes Naumann and Yvonne Neeb for sharing the data of the reading skill experiment with us.

6. REFERENCES

[1] Csáji, B.C., Weyer, E.: System identification with binary observations by stochastic approximation and active learning. IEEE CDC-ECE, pp. 3634–3639 (2011)

[2] Furr, R.M., Bacharach, V.R.: Psychometrics: an introduction. SAGE Publications, Incorporated (2007)

[3] Missura, O., Gärtner, T.: Predicting Dynamic Difficulty. Advances in Neural Information Processing Systems 24, pp. 2007–2015 (2011)

[4] Moosbrugger, H., Goldhammer, F.: FAKT-II Frankfurter Adaptiver Konzentrationsleistungs-Test II. Huber, Bern (2007)

[5] Richter, T., Isberner, M.B., Naumann, J., Kutzner, Y.: Prozessbezogene Diagnostik von Lesefähigkeiten bei Grundschulkindern. Zeitschrift für Pädagogische Psychologie 26(4), 313–331 (2012)

Discovering Students' Complex Problem Solving Strategies in Educational Assessment

Krisztina Tóth
German Institute for International Educational Research
Schloßstraße 29
60486 Frankfurt am Main
toth@dipf.de

Heiko Rölke
German Institute for International Educational Research
Schloßstraße 29
60486 Frankfurt am Main
roelke@dipf.de

Samuel Greiff
University of Luxembourg
6, rue Richard Coudenhove Kalergi
1359 Luxembourg-Kirchberg
samuel.greiff@uni.lu

Sascha Wüstenberg
University of Luxembourg
6, rue Richard Coudenhove Kalergi
1359 Luxembourg-Kirchberg
sascha.wuestenberg@uni.lu

ABSTRACT

A huge amount of log data accumulates automatically during computer-based educational assessments that can be analyzed for diagnostic or educational purposes using data mining techniques. In this paper, we describe our work of mining students' complex problem solving interactions when tackling previously unknown and dynamically changing situations.

Based on log data analyses, we discovered several problem solving strategies and examined relationships of these strategies and test outcomes. We applied clustering algorithms to discriminate between students with different levels of proficiency in problem solving. We identified four groups of students: two clusters represent successful problem solvers who differ in their level of efficiency, one group of inefficient students might need further practice to be able to solve these kinds of tasks, and finally we found a mixed-strategy group of students. Students in this last group were dynamically developing their problem solving strategy and in parallel, the ratio of correct responses increased from task to task during assessment. In sum, our findings help to advance research on cognitive processes; we support educational researchers in better understanding complex problem solving behavior and identify levels of problem solving proficiency.

Keywords

Complex problem solving, clustering, MicroDYN, test-taking behavior.

1. INTRODUCTION

Computer-based assessment allows new ways of investigating processes involved in complex problem solving (CPS) ([14]) when students solve real-life problems and the solution cannot be obtained by merely applying preexisting knowledge. To give an example: imagine you bought a new smartphone and would like to install a new application on it, but you have never used this kind of device before and do not want to read the manual. In this case, you cannot rely on previous knowledge and have to find out how the phone works by interacting with the device. After the exploration of the phone, you have a mental representation about how it works and can use your acquired knowledge to reach certain goals (e.g., to install an application). This kind of situation represents a complex problem where participants have to interact with task environments "that are dynamic (i.e., change as a function of user's intervention and/or as a function of time) and in which some, if not all, of the environment's regularities can only be revealed by successful exploration and integration of the information gained in that process" [2].

As the interactions between test-taker and this kind of problem situations (i.e., tasks) are essential for solving the CPS tasks, the CPS competency can only be measured in computerized environments. Due to computer-based test delivery, the test-takers' interactions with the tasks are automatically saved during the assessment. These large amounts of information pertaining to trace data constitute the basis for further analyses of participants' CPS behavior.

Based on the log data accumulated during the CPS assessment, we propose an identification of groups of students showing similar behavior in CPS assessment and identification of various CPS strategies with a clustering algorithm. Furthermore, we aim at examining relationships between the found CPS strategy and test outcomes which help us discriminate between students with different proficiency levels in problem solving.

2. RELATED WORK

CPS is a strong predictor of academic [13] and occupational achievement [3]. CPS has recently received considerable public interest, as CPS competency was tested in the Programme for International Student Assessment (PISA), a large-scale study of educational achievement assessing abilities of approximately half a million students in over 70 countries [10].

In CPS, three main processes can be distinguished: rule identification, rule knowledge acquisition and rule knowledge application [9]. In this paper, we aim at investigating CPS strategies in the rule identification phase.

In the case of data mining, empirical research has focused on investigating problem solving behavior with data mining algorithms (e.g. [1], [4], and [12]). All of these studies involved students facing other types of problem solving situations (like spreadsheets, IMMEX environment). To our knowledge no research based on data mining has yet investigated dynamic CPS strategies, as applied in this paper.

From the perspective of method, the closest study to ours was presented by [1]. It used predefined variables describing participants' problem solving behavior (like time duration, "maximum number of meters that were opened" [1]) and applied K-means to identify groups of students. Other relevant behavior analysis research often applies sequential pattern finding algorithms to search for sequences of actions (e.g. [4]), but the interpretation of clustered action sequences in the field of CPS strategy is challenging. For this reason, we propose to utilize predefined features which characterize individual CPS behavior per task to help discover students' CPS strategies.

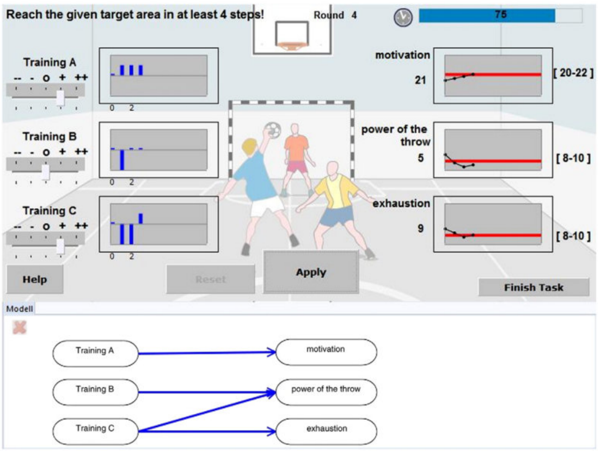
3. METHODS

3.1 Sample

393 German students attending grades 10 to 12 participated in this study (age*:* M=17.07, SD=1.12; 60% female, 1% did not report on gender). Participation was voluntary (see [11]).

3.2 Instrument

Tasks based on the MicroDYN approach [6] were used in this study to measure CPS. MicroDYN tasks are based on linear structural equations, in which up to three input and output variables are related. Participants' tasks while dealing with MicroDYN can be best described by means of the sample task "Handball" depicted in Figure 1.



Reach the given target area in at least 4 steps!

Figure 1 Sample task from the test.

In this task, different kinds of training (Training A, Training B and Training C) serve as input variables and different team characteristics (motivation, power of throw, and exhaustion) serve as output variables. Participants have to find out how the different kinds of training affect the team characteristics in order to reach a given goal state. While working on this task, participants are confronted with three different phases. First, participants freely explore the task by implementing adequate strategies to find out how input and output variables are related (rule identification phase). Therefore, participants manipulate the sliders (cf. Figure 1; below Training A, B and C) in order to change the value of the training (from – – to ++), click on "Apply" and try to retrace changes in the output variables according to their actions. Second, participants are asked to draw a causal diagram between input and output variables to visualize relations between training types and team characteristics (knowledge acquisition phase). Third, participants have to apply the retrieved knowledge in order to reach given target goals in the output variables within four steps (knowledge application phase).

3.3 Data collection

Data collection took place on several days between March and April 2011. The CPS test contained eight interactive tasks. For the present survey, three tasks were selected representing different difficulty levels. The test was administered on the school computers. In groups of maximum 16, students worked on MicroDYN for about 45 minutes.

3.4 Dataset

As outlined, log file data of the rule identification phase were analyzed in this study to identify various CPS strategies. Based on [6], one of the relevant CPS strategies by which relations between variables are identified is "vary-one-thing-at-a-time" (VOTAT), whereby only one input variable is manipulated and the others are kept constant [13]. However, VOTAT was examined as a dichotomous variable at task-level by [5], while we investigated the strategy continuous variable to get detailed information about action level activities. We integrated further action-level features (see Table 1) to offer additional information about individual activities and time information for diagnostic purposes [7]. Consequently we extracted features (see Table 1) for each student to characterize his or her individual CPS behavior. Table 1 contains the pairs of extracted features with respective interpretation.

Table 1. Extracted features and interpretations

Features	Interpretation
Number of executions	The frequency of using the apply button in the rule identification phase
Ratio of rounds with one input variable	No. of executions in which only one input variable was manipulated, divided by the total no. of executions
Ratio of zero rounds	The frequency of executions in which all sliders are set on 0 divided by the total number of executions
Ratio of repeated executions	Number of executions which were previously applied to the task, divided by the total no. of executions
Exploration time	Time given in seconds spent in the rule identification phase

3.5 Method of Analysis

The method of analysis used in this paper is X-means algorithm (with Euclidean distance) implemented in the WEKA tool [8] to detect patterns of (unlabeled) CPS behavior data. The data set contains 393 feature vectors that describe students' activities during test-taking. Each feature vector has 15 attributes, the introduced five process measures (see Table 1) built for all three complex tasks.

4. RESULTS AND DISCUSSION

4.1 Investigating CPS behavior with process measures

The CPS behavior of the students is illustrated in Table 2. It contains the average values of each feature. Data provided in Table 2 indicates that the number of executions ($t_{task1\text{-}task2}$=8.09 and $t_{task2\text{-}task3}$=4.43, p<.01), the exploration time ($t_{task1\text{-}task2}$=9.21

and $t_{task2\text{-}task3}$=7.37, p<.01) and the ratio of repeated executions ($t_{task1\text{-}task2}$=2.97 and $t_{task2\text{-}task3}$=3.24, p<.01) considerably decreased during the test-taking. In line with this finding, the ratio of rounds with one input variable increases significantly ($t_{task1\text{-}task2}$= -7.03 and $t_{task2\text{-}task3}$=-5.25, p<.01). This tendency confirms that students enhanced their CPS competency by practicing while working on the test.

Table 2. Mean value of each feature

Features	Task 1	Task 2	Task 3	*Total*
Number of executions	11.67	8.04	7.03	*8.96*
Ratio of repeated executions	.32	.28	.25	*.28*
Rounds with one input variable	.65	.75	.81	*.74*
Ratio of zero rounds	.07	.06	.06	*.07*
Exploration time	72.55	58.81	51.15	*60.83*

It is apparent that students accomplished tasks in an increasingly effective way and they more and more preferred the VOTAT strategy even though the CPS situations became more challenging as the complexity of tasks increased (the number of involved variables and the connectivity between input and output variables).

4.2 Clustering results

We identified four groups of participants which are characterized by cluster centroids (see Table 3). The members of the first subgroup (27.41 % of the whole sample) prove to be the most active problem solvers in this sample; they have the highest number of executions. Participants in Cluster 2 seem to be the most passive problem solvers as they have the lowest number of executions and they spent the least amount of time on solving the problems. The members of the third cluster are not as active as students in Cluster 1, but they take the longest to decide about the connections of input and output variables. Cluster 4 members required about the average time to solve the tasks, but used a lower number of executions than Cluster 1 and 3 members. However, the most striking difference is that the students in Cluster 4 used the lowest number of repeated executions.

To gain a deeper insight into the four cluster characteristics, we created Table 4 to represent the ratio of correct answers in knowledge acquisition on all tasks among clusters. The ratio of the correct solutions in Cluster 1 and 2 is about 90%, so these clusters represent problem solvers who correctly solved the tasks. But Cluster 3 and Cluster 4 members proved to be significantly less efficient than participants in Cluster 1 and 2.

Based on the ratio of correct responses and process measure values, we identified four groups of students, represented in the four clusters: (1) goal oriented VOTAT-strategy problem solvers, (2) less efficient VOTAT-strategy problem solvers, (3) non-VOTAT strategy users and (4) mixed-strategy users. Goal oriented problem solvers (see Cluster 2) would seem to require only a low number of executions for testing their hypothesis about the influence of factors on more dependent variables, change mostly one aspect of the system (VOTAT strategy), repeat only a few executions and need little time to solve the tasks successfully. Less efficient problem solvers (see Cluster 1) are test-takers who also mostly varied one factor while others were held constant (VOTAT strategy), but used a much higher number of executions (almost three times as many) and required more exploration time for identifying correct rules than goal oriented VOTAT-strategy users.

Table 3. Cluster centroids of the X-means clustering analysis

Features	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Ratio of students	27.41%	19.80%	37.56%	15.23%
Task 1				
No. of executions	22.38	6.72	11.55	7.47
Repeated executions	.63	.18	.34	.12
Rounds with one input variable	.72	.86	.37	.34
Ratio of zero rounds	.08	.04	.13	.05
Exploration time	81.76	60.60	82.45	75.35
Task 2				
No. of executions	13.97	4.84	8.84	5.14
Repeated executions	.56	.15	.31	.09
Rounds with one input variable	.82	.95	.38	.62
Ratio of zero rounds	.07	.02	.16	.01
Exploration time	63.13	49.18	70.63	59.72
Task 3				
No. of executions	11.46	4.43	8.28	4.33
Repeated executions	.53	.13	.27	.06
Rounds with one input variable	.89	.97	.46	.75
Ratio of zero rounds	.08	.01	.16	.01
Exploration time	52.72	42.47	64.82	51.22

Non-VOTAT strategy users (Cluster 3) varied multiple aspects at once (the ratio of correct responses is only .51 in contrast to .90 in Cluster 1 and .92 in Cluster 2) and used a significantly higher number of zero rounds than VOTAT-strategy problem solvers although these tasks did not require zero rounds and they needed the most time to explore the system.

The fourth group of students dynamically developed their CPS strategy, e.g. the ratio of one input variable in a round increases from 0.34 to 0.75. However in contrast with this the exploration time (from 75.35 to 51.22 seconds) and ratio of repeated rounds significantly decreased. The ratio of correct responses correspondingly increased from task to task (as can be seen in Table 4), so the students became more effective during the educational assessment.

In sum, the group of high-achievers using VOTAT strategy can be split into two categories: goal oriented and less efficient VOTAT-strategy problem solvers. Although these students found the connections between input and output variables, they demonstrated different CPS behaviors, which can only be detected by investigating process related test-taking data.

Table 4. Ratio of correct responses

Groups of students	Task 1	Task 2	Task 3	*Total*
Cluster 1	.86	.89	.94	*.90*
Cluster 2	.92	.90	.95	*.92*
Cluster 3	.56	.42	.54	*.51*
Cluster 4	.45	.55	.68	*.56*

Furthermore, based on the introduced process measures we made a distinction between non-VOTAT strategy users and mixed-strategy users. The mixed-strategy problem solvers' group was learning from the tasks and changed/improved their CPS strategy during the test taking. For this reason, they need more time with the same test material to improve their CPS competency. But students using non-VOTAT-strategy were not learning from the test, so they need other type of instructional intervention.

These findings help to advance the research on CPS processes; we thus support educational researchers in better understanding CPS behavior and identifying levels of CPS proficiency. In addition our study helps to detect different types of instructional interventions to improve students' individual CPS competencies.

5. FUTURE WORK

As an important next step, we need to verify our results taking additional data into account. This is a rather straight-forward step that can be split into looking at additional test results for the same set of tasks and afterwards trying to find similar results looking at other CPS tasks. In parallel, we want to examine the literature on CPS strategies. To our knowledge, the groups we have identified are not described elsewhere. Generally, VOTAT and non-VOTAT strategies are distinguished but not analyzed any further, for instance, by integrating process data.

We are currently collaborating with item and test developers working on complex problem solving. Our findings can help them improve their tasks and gain a deeper understanding of how students interact with the tasks. Finally, our results can be used to go beyond dichotomous grading of CPS items. We can try to help students following an inefficient path and use certain interventions to put them back on track.

6. REFERENCES

[1] Angeli, C. and Valanides, N. 2013. Using educational data mining methods to assess field-dependent and field-independent learners' complex problem solving. *Educational Technology Research and Development*. 61, 3 (June 2013), 521-548. DOI: 10.1007/s11423-013-9298-1

[2] Buchner, A. 1995. Basic topics and approaches to the study of complex problem solving. In Frensch P. A. and Funke, J. (Ed.) *Complex problem solving: The European perspective* (27-63). Hillsdale, NJ: Erlbaum.

[3] Danner, D., Hagemann, D., Schankin, A., Hager, M., and Funke, J. 2011. Beyond IQ: A latent state-trait analysis of general intelligence, dynamic decision making, and implicit learning. *Intelligence*. 39, 5 (Sept.-Oct. 2011), 323–334. http://dx.doi.org/10.1016/j.intell.2011.06.004

[4] Fern, X., Komireddy, C. Grigoreanu, V., and Burnett, M. 2010. Mining Problem-Solving Strategies from HCI Data. *ACM Transactions on Computer-Human Interaction*. 17, 1, Article 3 (March 2010), 1-22. doi:10.1145/1721831.1721834

[5] Greiff, S., Wüstenberg, S., and Funke, J. 2012. Dynamic problem solving: A new measurement perspective. *Applied Psychological Measurement*. 36, 3 (May 2012), 189–213. doi: 10.1177/0146621612439620

[6] Greiff, S., Wüstenberg, S., Holt, D. V., Goldhammer, F. and Funke, J. 2013. Computer-based assessment of Complex Problem Solving: concept, implementation, and application. *Educational Technology Research and Development*. 61, 3 (June 2013), 407-421. doi: 10.1007/s11423-013-9301-x

[7] Gvozdenko, E. and Chambers, D. 2007. Beyond test accuracy: Benefits of measuring response time in computerised testing. *Australasian Journal of Educational Technology*. 23, 4, 542-558.

[8] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, R. and Witten, I.H. 2009. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.* 11, 1 (November 2009), 10-18. DOI: http://doi.acm.org/10.1145/1656274.1656278

[9] Kröner, S., Plass, J. L., and Leutner, D. 2005. Intelligence assessment with computer simulations. *Intelligence*. 33, 4, (July-August 2005) 347–368. DOI:http://dx.doi.org/10.1016/j.intell.2005.03.002

[10] OECD. 2010. *PISA 2009 Results: Learning to Learn – Student Engagement, Strategies and Practices* (Volume III). PISA, OECD Publishing. DOI: http://dx.doi.org/10.1787/9789264083943-en

[11] Schweizer, F., Wüstenberg, S., and Greiff, S. 2013. Validity of the MicroDYN approach: Complex problem solving predicts school grades beyond working memory capacity. *Learning and Individual Differences*. 24 (April 2013), 42–52. DOI: http://dx.doi.org/10.1016/j.lindif.2012.12.011

[12] Vendlinski, T. and Stevens, R. 2002. Assessing student problem-solving skills with complex computer-based tasks. *Journal of Technology, Learning, and Assessment*, 1,3, Available from http://www.jtla.org

[13] Wüstenberg, S., Greiff, S., and Funke, J. 2012. Complex problem solving—More than reasoning? *Intelligence*. 40, 1 (Jan.-Febr. 2012), 1–14. doi: 10.1016/j.intell.2011.11.00

[14] Zoanetti, N. 2010. Interactive computer based assessment tasks: How problem-solving process data can inform instruction. *Australasian Journal of Educational Technology*. 26, 5, 585-606.

Discovering Theoretically Grounded Predictors of Deep vs. Shallow Level Learning

Carol Forsyth,Arthur Graesser,
Philip Pavlik,Jr.
The University of Memphis
The Institute for Intelligent Systems
365 Innovation Dr.
Memphis, TN 38152
1 001 901 678 2000
cmfrsyth,graesser,
ppavlik@memphis.edu

Keith Millis
Northern Illinois University
Psychology Building
Dekaulb, IL 60115
1 001 815 753 7087
kmillis@niu.edu

Borhan Samei
The University of Memphis
The Institute for Intelligent Systems
365 Innovation Dr.
Memphis, TN 38152
1 001 678 2000
bsamei@memphis.edu

ABSTRACT

We investigated predictors of shallow and deep learning for 192 college students with high vs. low prior knowledge in a game-like intelligent tutoring system, OperationARA that has an eText, multiple-choice tests, case-based reasoning, and adaptive tutorial conversations. Students are expected to learn about 11 topics of research methodology across three modules that target factual information, application of reasoning to specific cases, and question generation. Our approach blends evidence-centered design (ECD) and educational data mining (EDM) methods to discover the best predictors of deep and shallow level learning for students of varying aptitudes within this game. Theoretically-grounded constructs (e.g., time-on-task, generation, discrimination) were found to be significant predictors of deep vs. shallow knowledge acquisition.

Keywords

Intelligent Tutoring Systems, evidence-centered design, learning, reasoning

1. INTRODUCTION

One major goal of computer-based learning sciences is to predict learning from behaviors and events in technology-based environments. Accomplishing this goal requires a mix of two schools of thought. First, *evidence-centered design* (ECD; [10]) proposes an accurate linking between theoretically-grounded constructs and observable measures. Second, *educational data mining* (EDM; [2]) suggests appropriate statistical modeling to discover phenomenon occurring in educational settings. The current investigation attempts to link these two important schools of thought while investigating learning within an Intelligent Tutoring System (ITS). Specifically, in line with evidence-centered design, well-researched cognitive constructs are investigated as predictors of learning at a fine-grained level. Educational data mining techniques make it possible to discover unexpected patterns on large scale data that may have nested factors, such as different students, instructors and classrooms.

The current study uses these techniques to investigate theoretically-grounded constructs (e.g. time-on-task, generation, and discrimination) as predictors of deep vs. shallow level learning for students with high vs. low prior-knowledge levels in an ITS known as Operation ARA.

1.1 Operation ARA

Millis and colleagues [9] created Operation ARA (previously known as *OperationARIES!*) with the hopes of increasing students' knowledge of research methodology in an environment that is adaptive to students' prior-knowledge levels and that is dynamic and engaging. Both game features and pedagogical techniques are incorporated in Operation ARA. However, the focus of the current study is on the pedagogical features occurring across three distinct modules: teaching students the basic factual information (Cadet Training), application of knowledge (Proving Ground), and question generation (Active Duty).

Across these three modules, students engage in different learning activities while learning 11 topics of research methodology (e.g. causation vs. correlation, random assignment). In the Cadet Training Module, students learn the basic didactic information about the topics via an E-text, multiple-choice questions and natural language tutorial conversations between the human student and two artificial agents. In the Proving Ground module, the student must apply the information learned in the Cadet Training module by identifying flaws in research cases with the help of agents and a hint list that includes a list of potential flaws. An example flaw is "the dependent variable is not valid", or "correlation was confused with causation". Finally, in the Active Duty module, learners actively generate questions about an abstract of a research case and judge the validity of the answer.

Thousands of measures are collected across the learning activities embedded within the three modules. This investigation incorporates pedagogical principles in the learning sciences to choose the measures that may have the most meaningful relationships with shallow and deep learning for college students that vary in prior-knowledge about research methods.

1.2 Well-Researched Cognitive Constructs

Cognitive psychologists have identified several performance metrics as well as cognitive and discourse constructs that predict learning for students in complex learning environments [11,14,15]. Shallow learning includes comprehension of explicit information whereas deep learning requires a mental model about the topics that can be applied to reasoning about cases. Separate constructs may correlate with deep vs. shallow learning at varying depths of processing. Evidence-centered design assumes that each of these hypothetical constructs is carefully aligned with the measures, events, and behaviors that are collected throughout the learning experience. The approach is to identify a small number of general constructs with theoretical underpinnings that are good candidates for predicting learning at the different depths of conceptual processing. The three constructs explored here are time-on-task, generation, and discrimination. These constructs are expected to have different weightings across topics, items, and students, which can be discovered with data mining techniques.

The time a student spends on any particular academic activity is referred to as *time-on-task*. Multiple empirical investigations substantiate a positive relationship between time-on-task and learning [4,14]. Varying degrees of time may be needed depending on the learner's prior-knowledge corresponding to the novelty of the information and the depth of processing on a shallow to deep-level continuum [5,13]. Taraban and colleagues [14] substantiate the positive relationship between time and task and learning, but also suggest that sensitive measures are required to discover the fine-grained relationships between time-on-task and learning.

Generation can be defined as the amount of words produced by students in their self-explanations, questions, and ideas articulated during learning. Beneficial effects of generation over passively reading have been reported in empirical investigations to increase deep learning [3,15]. The similarity theory [7], as well as the semantic associative memory (SAM) model [12], explains the generation effect by postulating a network of semantic associations that get activated during learning, with concepts activating semantically similar concepts in the network. The active generation of information both facilitates and is facilitated by the conceptual similarity of material, but sometimes at the expense of discriminating important distinctions and contrasts. Moreover, generation of information increases with greater organization of material based on prior knowledge [8] and greater depth of processing [5].

Discrimination can be described as separating the signal from the noise, or identifying a correct answer when provided with multiple alternatives. Students can obtain a deep-level conceptualization of difficult concepts through tasks that require them to discriminate between multiple alternatives [1,15] that require subtle distinctions. The theoretical underpinnings of this construct have been captured in the SAM model [12] as well as other models in traditional verbal learning and memory paradigms that focus on the distinctiveness versus similarity of information [7]. Hunt and McDaniel [7] suggest that distinctive items are more likely to be remembered in tasks that rely on recognition (corresponding to shallow knowledge) rather than recall of information (corresponding to deep-level knowledge), whereas similarity enhances performance in tasks that emphasize recall over recognition. However, the conceptual organization of the content must be specified for accurate predictions of performance in these memory paradigms.

The goal of the current investigation is to discover measures within the rich environment of Operation ARA representing all 3 of these time-honored constructs that predict shallow vs. deep learning considering the student's level of prior-knowledge and the topics studied across the three modules of Operation ARA.

2. METHODOLOGY

Participants included 462 students enrolled across 12 sections of research methods courses with 11 different instructors of an undergraduate Psychology course at Northern Illinois University. Students were expected to complete the game as part of the course curriculum, but they were not required to sign informed consents in compliance with the Institutional Review Board. Unfortunately, 232 participants were dropped because either they did not complete the consent form or had missing pretests or posttests. The data was further screened and revealed 38 participants to have extremely fast response times on either the pretest or the posttests. These participants were also excluded. The final number of participants was N = 192 across 11 classrooms and 9 instructors.

The study used a pretest-intervention-posttest design in which all students interacted with all of the modules of Operation ARA. The college students first completed one version of the assessment as a pretest. Next, the students interacted with the three modules of Operation ARA (i.e. Cadet Training, Proving Ground, and Active Duty). After completing the interaction, students completed the posttest.

2.3 Measures

2.3.1 Theoretically Grounded Constructs

Measures were calculated on a by-topic basis for each of the cognitive constructs (i.e. time-on-task, discrimination, and generation) within each of the three modules (i.e. Cadet Training, Proving Ground, and Active Duty). For all measures of time on-task, the square root of the overall metric was computed in order to achieve a normal distribution and accommodate diminishing returns from a gamma distribution with a long positive tail in the distribution. In the Cadet Training module, the measure for time-on-task was the square root of the time spent reading the E-text within the chapter. The measures for time-on-task in the Proving Ground and Active Duty modules were the square root of the total time spent per case for each module, respectively.

Discrimination was calculated for each module on a by-topic basis based on signal detection theory which compares correct answers from distractor information. In the Cadet Training module, discrimination was measured by performance on the multiple-choice questions within each chapter. In the Proving Ground and Active Duty modules, discrimination was scored by computing the proportional number of hits (correctly identified flaws) minus the proportional number of false alarms (incorrectly identified flaws).

Generation was calculated as the number of words produced by the student. Generation in the Training module was the overall number of words articulated by the student within each tutorial conversation per chapter. In the Proving Ground and Active Duty modules, the construct was represented by the total number of words generated by the student while articulating flaws or generating questions.

2.3.2 Assessment of Learning on a Topic Level

There were two versions of the pre- and post-test assessments (version A and version B), which were counterbalanced across students. Both versions included a total of 22 multiple-choice questions. There were two questions assigned to each topic, including a definition and applied question. The definitional questions were used as a measure of shallow learning, whereas the applied questions were a measure of deep learning. The fact that there were only two test items per topic would not provide a very sensitive measure. Therefore, the topics were clustered to gain a more reliable picture of the relationship between learning gains across the 11 topics. The topics were clustered in a previous study [6] based on learnability [(Posttest- Pretest)/ (1-Pretest)]/2 by using Multi-dimensional Scaling (ASCAL algorithm) that segregated topics into two groups (True Experiment vs. Sampling). The “True Experiment” cluster includes topics such as *control groups* and *random assignment* whereas the “Sampling” cluster includes topics such as *representative samples* and *subject bias*. Two topics were excluded because they did not fit into either cluster. In the current study, by using these two clusters, the 11 observations per participants were reduced to 2 groups.

After establishing the topic clusters, a proportional learning gains formula [(Posttest- Pretest)/ (1-Pretest)] was used to calculate learning for shallow and deep items to account for prior knowledge. For shallow and deep-learning gains, the proportional learning gains were also calculated independently for each topic cluster resulting in 4 PLG scores for each participant. Extreme negative values (PLG < -1) were removed from the data on an item level, which reduced the total number of items from 768 to 743 across the 192 participants.

3. ANALYSES AND RESULTS

Before performing any analyses, the measures were transformed using the Winsorizing method to ensure no outliers would skew the data. This method ensures that all outliers beyond 3 standard deviations above or below the mean of the z-score of the given measure are transformed to reflect endpoint scores. Next, two median splits were performed. The first separated the students into two groups based on prior-knowledge (i.e. high vs. low) for shallow learning gains. The second separated students into high and low-prior-knowledge for deep level learning gains. Therefore, the one participant could potentially be in a high-prior knowledge group for shallow learning and in the low- prior knowledge group for deep-level learning. This means that the four groups did not have an equal number of subjects as one subject could be in multiple groups, but rather the goal was to seek group equivalence. The final groups included: (Group 1) low prior-knowledge and shallow learning (N = 141 with 188 units of analyses), (Group 2) high prior-knowledge and shallow learning (N = 141 and 188 units of analyses), (Group 3) low prior-knowledge and deep learning (N = 141, 187 units of analyses), (Group 4), high prior-knowledge and deep learning (N = 126, 176 units of analyses).

Separate analyses were conducted for each of the 4 groups in the following stages. First, Pearson correlations were computed between the cognitive constructs and the PLG. Although this violates the assumption of independence of observations in correlation, these correlations are simply used as a guide. Next, a series of linear mixed-effect regression models were used to test models that included the highly significant correlates (r >|.2|) and also that accounted for the nested factors of participant, classroom, and instructor. The full models included the significant correlates as fixed factors and the random factors of participant, classroom, instructor as well as test form to account for counter-balancing test forms. The best fit models were then validated using 50 iterations of 4-fold cross-validation on the linear mixed fixed-random effects modeling using the R package “lme4” version 1.1-6 that was just released in 2014. Several of the random factors (i.e. participant, instructor, classroom) were not included in the cross-validation because equal distributions were not maintained across the training and test folds with the current dataset. A generalization proportion is also reported for each model. This is the proportion of the training-fold explained variance that generalizes to the test fold. These analyses were performed for each of the four groups (i.e. low knowledge and shallow learning, high knowledge and shallow learning, low knowledge and deep learning, and high knowledge and deep learning).

3.1 Low Knowledge & Shallow Learning

All of the potential predictors (the cognitive constructs for each module) were correlated with the shallow proportional learning gains (PLG) for students with low prior-knowledge. The analyses revealed a significant correlation between the discrimination metric in the Active Duty Module (referred to as ADdisc) with the PLG (r (189) = .24, p <.001).

The full model of ADdisc (i.e. discrimination in the Active Duty module) as a fixed factor with the 4 random factors of participant, classroom, instructor, and test was significantly different from the null model including only the random factors (X^2 (1) = 10.81, p <.001). The ADdisc accounted for about 5.2% of the variance above the random effects (R^2 =.052). The relationship between discrimination in the Active Duty module and PLG was positive in nature (β =.24, p <.001). This means that greater discrimination identifying flaws in the Active Duty module correlates with higher shallow proportional learning gains. The 4-fold cross validation of the mixed model including ADDisc as a fixed factor and test form as a random factor revealed a training set accounting for 6.2% of the variance and a test set accounting for 5.4% of the variance (R^2=.062 and R^2=.054, respectively). The generalization proportion was .86.

3.2 High Knowledge & Shallow Learning

The correlational analyses revealed strong correlations between Topic group (i.e. True Experiment vs. Sampling) and the number of words generated in the Proving Ground Module (referred to as PGwords) each significantly correlated with the shallow-level proportional learning gains (r(188)=.23, r(188)=.21, respectively).

The linear mixed-effects model with Topic group (i.e. Experimental vs. Sampling) and PGWords (generation in the Proving ground module) as fixed factors with the 4 random factors of participant, topic, and test was significantly different from the null model (X^2(2) = 16.67, p <.001) with the overall model accounting for 9% of the variance. Specifically, Topic Group accounted for about 5% of the variance (R^2 = .047) and PGWords accounted for 4% of the variance (R^2 = .039). Both the Topic Group and the words generated in the Proving Ground module had a positive relationship with proportional learning gains (β = .2, p <.01, β = .19, p <.01, respectively). The 4-fold cross validation of the full model including Topic Group and PGWords as fixed factors and test form as a random factor revealed a training set accounting for 10.5% of the variance and a test set accounting for 7.9% of the variance (R^2= .105 and R^2= .079, respectively) with a generalization proportion of .75.

3.3 Low Knowledge & Deep Learning

The Pearson correlations revealed a strong correlation between the discrimination metric in the Proving Ground module (referred to as PGDisc) as well as the time spent in the Active Duty module (referred to as ADTime) with the PLG (r =-.23, r =.24, respectively).

The mixed-fixed random effects model with PGDisc and ADTime as fixed effects and the 4 random effects was significantly different from the null model($X^2(2) = 16.99$, $p < .001$). The overall model accounted for about 8.5% of the variance ($R^2 = .085$) above the null model that included only the random factors with PGDisc accounting for 5.4% of the variance and ADTime accounting for 3.2% of the variance (R^2=.054, R^2=.032, respectively). Specifically, discrimination in the Proving Ground module was negatively correlated with learning whereas the time spent in the Active Duty module was positively correlated with learning ($\beta = -.18$, $p < .05$; $\beta = .18$, $p < .05$, respectively). The 4-fold cross validation of the full model including PGDisc and ADTime as fixed factors and test form as a random factor revealed a training set accounting for 9.2% of the variance and a test set accounting for 7.4% of the variance (R^2= .092 and R^2= .074, respectively) with a generalization proportion of .81.

3.4 High Knowledge & Deep Learning

Pearson correlations were performed between each of the constructs of interest and the proportional learning gains for the applied or deep questions. Unfortunately, no strong significant correlates were discovered. Therefore, the rest of the analyses were not conducted as the researchers concluded that a predictive model for deep-level learning for high-prior knowledge students could not be discovered from these data.

4. CONCLUSIONS

The investigation revealed significant models for three of the four groups of high versus low prior-knowledge for shallow versus deep learning. Specifically, discrimination in the Active Duty module (i.e. the question generation module) was the most predictive measure of shallow learning for students with low prior-knowledge. Word generation in the Proving Ground Module and sampling-oriented topics were positively correlated with shallow learning gains for high prior-knowledge students. The predictive model for students with low prior-knowledge suggested a negative relationship between discrimination in the Proving Ground module and deep-level learning gains as well as a positive relationship between the time spent in the Active Duty module (where students generate questions) and deep-level learning. Each of the models makes sense within the theoretical frameworks of the cognitive constructs used as predictors although they were not predicted a priori but rather discovered through educational data mining methods. Unfortunately, no predictors were found for high prior-knowledge students and deep-level learning. Perhaps good students with high-prior knowledge will achieve deep learning gains regardless of the tutorial experience.

There are limitations in this study. There could be a greater number of observations per prior-knowledge and deep versus shallow groups. There is also the possibility of other measures being better predictors of deep versus shallow level learning. A current investigation is underway to test multiple measures per construct and thereby determine the best predictors of deep vs. shallow level learning. Although there were limitations to this study, the overall results support the approach of blending evidence-centered design and educational data mining to conduct fine-grained investigations of student interactions within an Intelligent Tutoring System. Both are needed to identify when a particular learning principle will be effective for a particular topic and type of student.

5. ACKNOWLEDGMENTS

This research was supported by the Institute for Education Sciences (R305B070349) and National Science Foundation (HCC 0834847). The opinions expressed are those of the authors and do not represent views of the IES and NSF.

6. REFERENCES

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. 1995. Cognitive tutors: lessons learned. *J. Learn. Sci. 4*, 167-207.

[2] Baker, R.S.J.D., and Yacef, K. 2009. The state of educational data mining in 2009: A review and future visions. *J. Ed. Dat. Min*. 1, 3-17.

[3] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., and Hausmann, R. G. 2001. Learning from human tutoring. *Cog. Sci.* 25, 471–533.

[4] Chickering, A. W., and Gamson, Z. F. 1987. Seven principles for good practice in undergraduate education. *AAHE Bull.* 39, 3-7.

[5] Craik, F.I. M. and Lockhart, Robert S. 1972. Levels of processing: A framework for memory research. *J. Ver. Learn. Ver. Beha*, 11, 671–684.

[6] Forsyth, C.M., Graesser, A.C., Cai, Z., Pavlik, P., Millis, K., and Halpern, D. 2013. Learner profiles emerge from a serious game teaching scientific inquiry. Presented at the *Annual Meeting of the American Educational Research Association*. (San Francisco, CA, April, 2013).AERA '13.

[7] Hunt, R. R., and McDaniel, M. A. 1993. The enigma of organization and distinctiveness. *J. Mem. Lang.* 3 ,421-445.

[8] Mandler, G. 1968. Organized recall. Individual functions. *Psych. Sci.* 13, 235-236.

[9] Millis, K., Forsyth, C., Butler, H., Wallace, P., Graesser, A., Halpern, D. 2011. Operation ARIES!: A serious game for teaching scientific inquiry. In M.Ma, A. Oikonomou & J. Lakhmi (Eds.) Serious Games and Edutainment Applications (pp.169-196). London, UK. Springer-Verlag, 2011.

[10] Mislevy, R.J, Steinberg, L.S., and Lucas, J.F. 2003. On the structure of educational assessments.*Measurement: Interdisc. Res. and Pers*. 1, 3-67.

[11] Pashler, H., Bain, P. M., Bottge, B. A., Graesser, A., Koedinger,K.,and McDaniel, M. 2007.Organizing instruction and study to improve student learning: IES practice guide Washington, DC: National Center for Education Research, 2004-2007.

[12] Raaijmakers, J. G. W., and Shiffrin, R.M. 1981. Search of associative memory. *Psych. Rev.* 88, 93-134.

[13] Simon, H. A. 1990. Invariants of human behavior. *Ann. Rev. Psych*. 41, 1-19

[14] Taraban, R., Rynearson, K., and Stalcup, K. 2001. Time as a variable in learning on the World Wide Web. *Beh. Res. Met., Instr., Comp*. 33, 217-225.

[15] VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., and Rose, C.P. 2007. When are tutorial dialogues more effective than reading? *Cog. Sci*. 31, 3-62.

Domain Independent Assessment of Dialogic Properties of Classroom Discourse

Borhan Samei[1] Andrew M. Olney[1] Sean Kelly[2] Martin Nystrand[3]
Sidney D'Mello[4] Nathan Blanchard[4] Xiaoyi Sun[3] Marcy Glaus[3] Art Graesser[1]

[1] University of Memphis [2] University of Pittsburgh [3] University of Wisconsin [4] University of Notre Dame
bsamei@memphis.edu

ABSTRACT

We present a machine learning model that uses particular attributes of individual questions asked by teachers and students to predict two properties of classroom discourse that have previously been linked to improved student achievement. These properties, uptake and authenticity, have previously been studied by using trained observers to live-code classroom instruction. As a first-step in automating the coding of classroom discourse, we model question properties based on the features of individual questions, without any information about the context or domain. We then compare the machine-coded results to two referents: human-coded individual questions and "gold standard" codes from existing data. The performance achieved by the models is as good as human experts on the comparable task of coding individual questions out of context. Yet ultimately, this study highlights the need to draw on contextualizing information in order to most completely identify question properties associated with individual questions.

Keywords

Classroom Discourse, Machine Learning, Authenticity, Uptake

1. INTRODUCTION

A particular style of classroom discourse, known as dialogic instruction, has been found to improve student achievement [1, 10, 11]. Dialogic instruction involves fewer teacher questions and more conversational turns as teachers and students alike contribute their ideas to a discussion. One way in which dialogic instruction leads to improved learning is by increasing student engagement in classroom instruction [2]. Moreover, when teachers focus on provoking student thought and analysis, and postpone evaluation during question and answer sessions by engaging in dialogic instruction, levels of student effort are more evenly distributed among students [7]. In the first major quantitative study of dialogic instruction, Nystrand and colleagues observed discourse practices in 8th and 9th grade classrooms over two years [9, 11]. Nystrand et al.'s coding approach focused on the nature of question *events*, which include the discourse context preceding and following a given question. Five properties of question events were coded: *authenticity, uptake, level of evaluation, cognitive level, and question source*. Nystrand and Gamoran reported that among these variables, authenticity and uptake are the most important properties affecting student achievment [1, 3, 10].

Within this context of dialogic instruction, authenticity is defined as a question for which the asker does not have a pre-scripted answer, i.e. open-ended questions. Such questions, particularly when asked by the teacher, create a context for students to contribute and develop their understanding to an evolving discussion. For example, "What was your reaction to the end of the story?" is an authentic question which leads to open-ended discussion, whereas questions such as "What was the father's name?" are not authentic.

Uptake in the context of dialogic instruction occurs when one asks a question about something that another person has said previously. Uptake of student ideas by the teacher therefore emphasizes the importance of student contributions. In previous work, these indicators were judged considering the question in context as opposed to just the individual question. Indeed, the very definition of uptake suggests that it is not possible to detect it from an isolated question, though this assumption and the corresponding assumption for authenticity have never been empirically tested.

In previous research, these variables were "live coded" by classroom observers who also recorded the question as an index of the discourse context preceding and following a given question event. Coding of question events, as opposed to isolated questions, are ultimately determined by teacher responses to students. In contrast, we attempt to predict the question event features of uptake and authenticity from the isolated question using machine learning techniques. Our work addresses a previously untested theoretical question of whether it is possible to recover these variables from the question, since the question is only loosely coupled to the event.

Olney et al. proposed a method to classify questions based on part-of-speech tagging, cascaded finite state transducers, and simple disambiguation rules [12]. They used 16 question categories which were defined in previous works on question classification [4, 5]. This classifier was manually designed using expert linguistic knowledge (a rule based system). We believed that this classifier, though designed for a slightly different purpose, used features that might be highly relevant to identifying uptake and authenticity, because we believed that different kinds of questions might lead to different levels of uptake and authenticity. For example, we hypothesize that yes/no questions are less likely to lead to extended discussion containing uptake and authenticity than causal questions about why an event occurred or why someone decided to take a certain course of action.

Based on the definition of uptake and authenticity, we expected to achieve a reasonable performance by using the same features as predictors as Olney et al. The study reported here shows that the performance of a machine learning approach based on features previously used in question classification is as accurate as expert humans on the task of classifying authenticity and uptake in isolated questions.

2. METHOD

Our long-term research goal is to develop cutting-edge classifiers in order to identify dialogic questions properties important to effective classroom discourse. In working towards this goal, in the present study we address two research questions: (a) How well do machine classifiers perform relative to trained human raters in coding individual questions *without supporting contextual information*? and (b) Do property codes ascertained from individual questions, either by human or machine, correspond well to *fully contextualized codes* (i.e., the "gold standard")?

To address these questions, we utilize existing data from a study of classroom instruction where fully contextualized question property codes had previously been generated [6, 7]. In addition, in order to answer the first research question, we collected new human ratings, using only the information available to the machine learning algorithm, i.e., the question out of context. This study represents the first empirical investigation of dialogic question properties at the level of individual questions.

2.1 Dataset

2.1.1 Gold Standard Data

The present study relies on the Partnership for Literacy Study data (Partnership), a study of professional development, instruction, and literacy outcomes in middle school. In Partnership study, 120 classrooms in 23 schools were observed twice in the fall and twice in the spring.

Observational data from Partnership classrooms were coded using CLASS 4.24, a computer-based data collection system [8]. Coding reliability studies using CLASS indicate that raters agree on question properties approximately 80% of the time, with observation-level inter-rater correlations averaging approximately .95 [10]. Importantly, the original Partnership codes were based on the full set of contextualizing information, including preceding discourse and classroom events.

In all, the Partnership data consist of 29,673 teacher and student questions coded using CLASS during question and answer sessions. In the present study, after removing partially incomplete observations where one or more of the question codes were missing, we utilized a subset of 25,711 questions as our training data, a subset of which is excluded from training and used as the "gold standard" for evaluation purposes.

2.1.2 Individual Question Coding

As a baseline for evaluation of our models, we asked four human raters who were experts in classroom discourse to code the questions of separate sample instances selected from the gold standard data (one sample for authenticity and another for uptake). The sample sets contained 100 questions exhibiting each category of the question property and a separate 100 not exhibiting that property. For example, the uptake set contained 100 questions originally rated as non-uptake and 100 as uptake.

All the questions in the samples were represented by plain text and randomly ordered so that human judgments were based on individual questions without any information about the context. The questions for both authenticity and uptake were rated using a binary (Yes/No) scale.

This task was designed to investigate the performance of human experts on rating the questions, using the same information that we use to build our classifier model with. We also calculated the agreement among human raters to address the difficulty of the task of rating questions in isolation. The performance of machine coding was compared to both the original live-coded data (coding in context) and the subset of data re-coded by human experts (coding in isolation).

2.2 Machine Learning

As mentioned earlier, we applied machine learning using the features based on previous work on question classification. The feature set consisted of 30 attributes including part of speech tags and sets of keywords. Most of the attributes are binary representing the presence/absence of certain keywords or part of speech in the question, for example 'NEG' is true if there is a negation keyword in the question or false otherwise. However, for some of the attributes we take into account the position of the keyword in the question by defining four values: middle, beginning, end, and none, in which the first three values show the position of the keyword if present in the question. For example, if a question consisted of four words, e.g. "word1 word2 word3 word4" the position of "word1" and "word4" are captured as beginning and end respectively. "word2" and "word3" are both captured as middle. Moreover, if we only had two words in the question, we consider first one as beginning and the other as end.

Binary attributes

In our feature set we defined binary attributes to represent the presence of particular words in the questions, regardless of position. These words are defined in sets in Olney et al.; therefore we define the attributes as true if any member of the set is present in the sentence. Causal consequent words, for example, were defined by a set of words including "outcomes," "results," "effects," etc. Similarly, procedural words included "plan," "scheme," "design," etc. The rest of binary attributes included feature specification, negation, meta-communication, metacognition, comparison, goal orientation, judgmental, definition, enablement, interpretation, example, quantification, causal antecedent, and disjunction which are also defined as sets of keywords related to them. We also defined some attributes representing certain words such as "happen," "no," and "yes." More complete descriptions of these features and their validation for question classification can be found in [12]. We used the source code from a simplified version of the question classifier released as part of the open-source GnuTutor project [13].

Other attributes

As mentioned above, for some of the attributes we defined values to represent the presence and position of certain words and part of speech tags. These attributes included part of speech tags such as determiner, noun, pronoun, adjective, adverb, and verb along with word lists: Do/Have (e.g. "don't," "having," and etc.), be (am, are, is, etc.), modal (would, might, etc.), and certain words such as "What," "How," and "Why." More complete descriptions and justifications of these features for question classification can be found at the references above. By including features for positional information we hoped to approximate the regular expression patterns of the Olney question classifier. However instead of directly using the patterns discovered previously, we decided to allow new approximate patterns to be discovered during the machine learning process. Although there might be a correspondence between previous work on question categorization and the constructs of authenticity and uptake, a 1-to-1 correspondence assumption appeared to be unwarranted.

The training data was selected from the "live-coded" data set (Partnership) to form a set of coded questions with uniform distribution of the authenticity and uptake variables. In the case of authenticity, the original distribution of data was close to uniform. New sampling to make the distribution completely uniform (base rate of 50%) yielded a set of 25,464 questions.

Uptake originally was defined by three values: test, authentic and no uptake; however we reduced the uptake to a binary scale of uptake and no-uptake. The original test uptake values were taken as no-uptake in the new scale. The argument for collapsing test uptake and no uptake is based on the observation that they have indistinguishable impact on student achievement. Collapsing test and no uptake and normalizing to a uniform distribution yielded a total of 9,579 instances with an even distribution of uptake and no-uptake. The magnitude of this reduction relative to the set of authentic questions reflects the large number of instances that were originally coded as no-uptake.

These selected instances from the original "live coded" data were then separately used as gold standards to train the two classifiers for predicting uptake and authenticity on isolated questions. The subset of instances given to the expert judges was excluded from training data and was used to test the models. We used WEKA [14] to train and test J48 decision tree classifiers to predict authenticity and uptake.

3. RESULTS & DISCUSSION

We evaluate our models' performance by comparing the performance to the expert re-coded sample as our baseline (coding in isolation), using the gold standard data as the reference (coding in context). Thus the baseline performance was measured by evaluating the performance of four experts on the task the machine classifiers faced: coding questions in isolation.

Cohen's kappa was used as a metric to assess reliability between two raters and between the computer and the rater. Results showed low agreement among human raters on the task, which suggests that in most cases human raters could not make strong judgments based only on the features of individual questions in isolation. The minimum kappa among human raters for authenticity was 0.18; however for other pairs the kappa ranged from 0.3-0.5 with a maximum of 0.55 and an average of 0.4. Similarly, the average inter-rater reliability for Uptake was 0.42, with a minimum of 0.31 and maximum of 0.51 kappa.

The overall low agreement among human raters illustrated the difficulty of making judgments based only on the individual questions as opposed to having information about the context and other properties of the classroom discourse around each question.

The machine learning model was trained on the gold standard data that were rated by Partnership observers. We built J48 decision tree models and tested the models on the same samples that were given to human raters—which were excluded from our traning data—and compared the performance of the model with experts in terms of kappa and recognition rate (Table 1).

Table 1. Kappa statistics and recognition rate of human raters and machine leaning model compared to Gold Standard ratings for authenticity (A) and uptake (U).

-	Kappa		Recognition Rate	
	A	U	A	U
R1	0.13	0.22	56%	61%
R2	0.17	0.25	58%	62%
R3	0.25	0.30	62%	65%
R4	0.10	0.23	55%	61%
Model	0.34	0.46	67%	73%

As seen in Table 1, the highest performance of human raters on predicting authenticity yielded an accuracy of 62% and 0.25 kappa. The performance of the model on predicting authenticity was better than human experts with 67% accuracy and 0.34 kappa. Authenticity was better judged in context, which is why human raters (coding in isolation) showed lower performance and agreement than the original raters (coding in context) of ~80%. By outperforming human raters on this task, our model's performance on authenticity implies that the features used in training are as predictive as could be considering the lack of contextual information. The performance of our model on uptake was markedly better than human experts. The highest performance for a human rater is an accuracy of 65% and 0.30 kappa. The performance of the model on predicting uptake is 73% accuracy and 0.46 kappa. A question with uptake, by definition, refers to a previous discourse contribution. However it appears that features of individual questions are indirectly marking uptake, because our feature set has suitability for predicting uptake in the absence of context. We also measure the overall performance of the model on the whole gold standard data using 10-fold cross validation. Table 2 shows the overall performance of the models.

Table 2. Overall performance of models on gold standard data using 10-fold cross validation

Models	Kappa	Accuracy
Authenticity	0.28	64%
Uptake	0.24	62%

The overall performance of the authenticity model on the gold standard data was close to performance on the sample data while the uptake model performed with a lower accuracy; however the results are still close to human raters coding questions in isolation which supports the reasonable performance of our models on this task.

To take a closer look at the models, we ran Correlation-based Feature Subset Selection (CFS) on our feature sets. CFS considers the individual predictive ability of each feature along with the degree of redundancy between them to evaluate the worth of a subset of attributes. The results showed that the highest ranked attributes used in predicting authenticity were: Judgmental keywords, WH words, Enablement keywords, and "what." Similar analysis on the decision tree for uptake yielded the following most useful attributes: negation keywords, Judgmental keywords, and "why." The importance of such features for predicting uptake can be inferred from the definition.

Although the CFS analysis identified Judgmental, Negation, and Enablement keywords as the most predictive keyword sets, the CFS analysis was unable to identify the actual keywords used because these keywords had been replaced by the labels corresponding to the keyword sets. To illustrate the actual words that were coded as these features, for each set of keywords we calculated the frequency of these words in the data set and measured the distribution of each word as a proportion of the frequency of all the words in the keyword set.

The distribution of Judgmental keywords showed that "think" (.83), "should" (.06), and "find" (.05) accounted for 94% of the total Judgmental keywords seen in the data set. Other keywords individually contributed less that 1%.

Similar analysis showed that the most frequent Enablement keyword was “need(ed)” (0.81) while the other enablement keywords were less frequent, e.g. “helpful,” (0.05), and “in order to,” (0.05). Furthermore, “not(n’t)” (0.95), was the most frequent negation word and other negation words such as “never” and “neither” contributed less than 1%.

The following questions, for example, were extracted from our dataset, to illustrate the use of mentioned keywords in the actual questions:

Questions with authenticity:

*“Do you **think** enterprising people always **need** to be audacious?”*
*“Did you **find** it **helpful**?”*
*“Do you **think** it **needed** to go on the next ten lines?”*

Questions with uptake:

*“**Why** do you **think** he wants to **help** the little boy?”*
*“You **think** he ca**n't** get **help**, Can you expand on that?”*
*“Like if I make a connection to my life and **not** to all three of them do you **think** that that might **help**?”*

Considering the size of our training data, these results suggest the coverage of our feature set in classifying questions out of context. Moreover, these features, as used in the models, are consistent with the theoretical definitions of authenticity and uptake.

4. CONCLUSION

We examined the performance of machine learning models compared to human experts in predicting authenticity and uptake on a random set of isolated questions sampled from a previous classroom study. The key aspect of our approach is that we did not use any contextual information regarding the discourse moves in the model, yet we showed that the models perform as well as human experts under the same restrictions.

The original coders (coding in context) achieved approximately 80% agreement, but in the current study the expert re-coders (coding in isolation) achieved only 60% with the original coders. This suggests that, on a coding task with equally probable categories, a roughly 20% gap in agreement could be attributed to missing contextual information. A surprising finding is that isolated questions provide sufficient cues to correctly identify many authentic questions and questions with uptake. Based on this finding it may be the case that authenticity and uptake can be redefined in terms of an adequate window size of context before and after the question. In future studies, we anticipate incorporating both additional preceding context and following context in determining authenticity and uptake codes.

.

5. ACKNOWLEDGMENTS

This research was supported by the Institute of Education Sciences (R305A130030) and the National Science Foundation (IIS 1352207). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of these sponsoring agencies.

6. REFERENCES

[1] Gamoran, A., and Nystrand, M. 1991. Background and instructional effects on achievement in eighth-grade English and social studies . *Journal of Research on Adolescence*, 277–300.

[2] Gamoran, A., and Nystrand, M. 1992. Taking students seriously. In F. Newmann, (Ed.), *Student engagement and achievement in American secondary schools*. Teachers College Press, New York.

[3] Gamoran, A., and Kelly, S. 2003. Tracking, instruction, and unequal literacy in secondary school English. In Hallinan, Gamoran, Kubitschek, & Loveless (Eds.). *Stability and change in American education: Structure, process, and outcomes*. Clinton Corners, NY: Eliot Werner.

[4] Graesser, A. C., and Person, N. 1994. Question asking during tutoring. *American Educational Research Journal*, 104-137.

[5] Graesser, A., Person, N., and Huber, J. 1992. Mechanisms that generate questions Erlbaum. *Questions and information systems*.

[6] Kelly, S. 2007. Classroom discourse and the distribution of student engagement within middle school English classrooms. *Social Psychology of Education*, 10, 331–352.

[7] Kelly, S. 2008. Race, social class, and student engagement in middle school English classrooms. *Social Science Research*, *37*, 434-448.

[8] Nystrand, M. 1988. CLASS (Classroom language assessment system) 2.0: A Windows laptop computer system for the in-class analysis of classroom discourse. Wisconsin Center for Education Research, Madison.

[9] Nystrand, M. 1997. Opening dialogue: Understanding the dynamics of language and learning in the English classroom . *Teachers College Press*, New York, NY.

[10] Nystrand, M, and Gamoran, A. 1997. The Big Picture: Language and learning in hundreds of English Lessons. In M. Nystrand. *Opening dialogue Understanding the dynamics of language and learning in the English classroom*. Teachers College Press, New York.

[11] Nystrand, M., Wu, L., Gamoran, A., Zeiser, S., and Long, D. A. 2003. Questions in time: Investigating the structure and dynamics of unfolding classroom discourse. *Discourse processes*, 135-198.

[12] Olney, A. M., Louwerse, M., Mathews, E. C., Marineau, J., Mitchell, H. H., and Graesser, A. C. 2003. Utterance classification in AutoTutor. *Human Language Technology - North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 1-8, Philadelphia.

[13] Olney, A. M. 2009. GnuTutor: An open source intelligent tutoring system based on AutoTutor. *Proceedings of the 2009 AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems*, AAAI Press. 70–75

[14] Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., and Cunningham, S. J. 2011. *Weka: Practical machine learning tools and techniques with Java implementations.* Morgan Kaufmann Publishers.

Empirically Valid Rules for Ill-Defined Domains

Collin F. Lynch
Center for Educational Informatics
North Carolina State University
Raleigh, North Carolina, U.S.A.
collinl@cs.pitt.edu

Kevin D. Ashley
Learning Research & Development Center
University of Pittsburgh
Pittsburgh, Pennsylvania, U.S.A.
ashley@pitt.edu

ABSTRACT

Ill-defined domains such as writing and design pose challenges for automatic assessment and feedback. There is little agreement about the standards for assessing student work nor are there clear domain principles that can be used for automatic feedback and guidance. While researchers have shown some success with automatic guidance through *a-priori* rules and *weak-theory structuring* these methods are not guaranteed widespread acceptance nor is it clear that the lessons will transfer out of the tutoring context into real-world practice. In this paper we report on data mining work designed to *empirically validate a-priori* rules with an exploratory dataset in the domain of argument diagramming and scientific writing. We show that it is possible to identify diagram rules that correlate with student performance but that direct correlations can often run counter to expert assumptions and thus require deeper analysis.

Keywords

Empirical Validity, Argument Diagramming, Ill-Defined Domains, Writing, Assessment, Intelligent Tutoring Systems

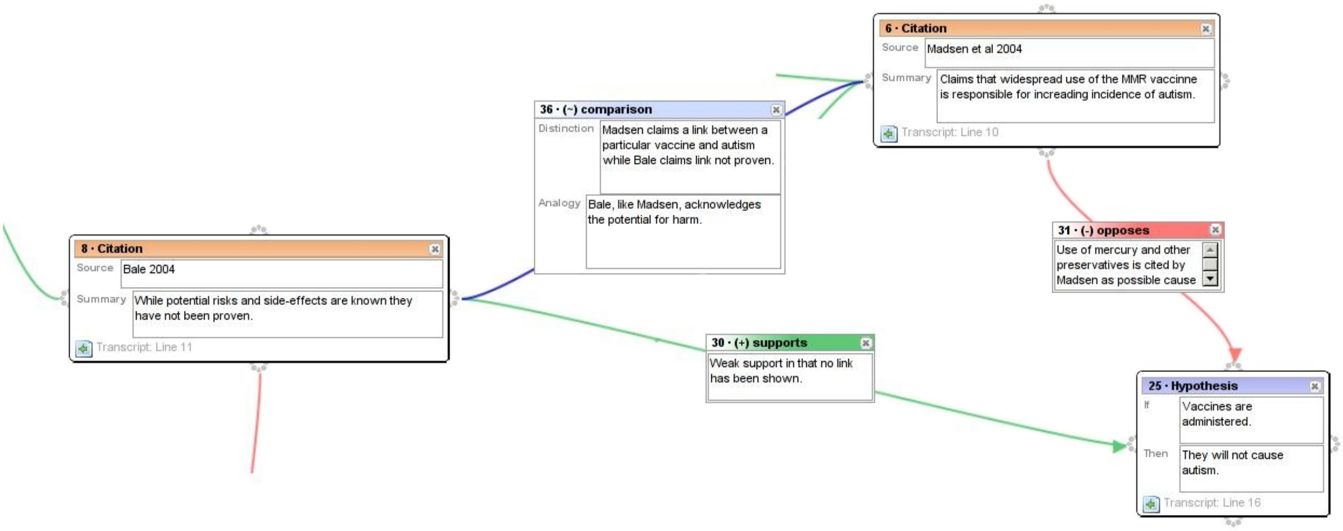
1. INTRODUCTION

Ill-defined domains such as writing and design pose key challenges for automatic assessment and feedback. Solvers of ill-defined problems must reify implicit or open-textured concepts or solution criteria to make problems solvable and then justify those decisions [8, 9]. Consequently, ill-defined problems lack widely-accepted domain theories or principles that can be used to provide automatic assessment and feedback. Moreover it is not always clear that automatic advice can generalize to a wider domain or transfer out of the study context into the real world. Our present goal is to identify *empirically valid* rules that both correlate with subsequent performance on the real-world tasks that are the target of instruction and can be used for guidance and assessment.

Prior researchers have advanced a number of techniques for guidance in ill-defined domains such as peer review and microworlds [9]. Researchers have also developed successful systems which guide students via optional rules or constraints [12, 10], a method known as *weak-theory scaffolding* [9]. This type of scaffolding can include use of constraints to bound otherwise open student solutions [15], or use of structured graphical representations combined with on-demand feedback as in Belvedere [14] and LARGO [12, 11].

LARGO, for example is a graph-based tutoring system for legal argumentation. Students use the system to read and annotate oral argument transcripts from the U.S. Supreme Court. As students read the transcript they identify crucial passages in the text containing legal *tests*, *hypothetical cases*, or *logical relationships* and represent them as elements in a graph with textual summaries. They are guided in this analysis via *a-priori* graph rules that detect violations of the argument model. While systems of this type have shown success, particularly with lower-performing students, no broad systematic attempt has yet been made to demonstrate the empirical validity of these graphical structures or rules. Validity is essential, especially in ill-defined domains, where the utility of the models have been assumed but where we cannot always be sure that a given violation of the model is a student error and not a judicial prerogative. Demonstrating empirical validity of the argument models would support their use both pragmatically, by helping to persuade skeptical domain experts that they are effective, and functionally, by providing us with an empirical confidence measure that can be used to evaluate or weight their implementation.

We have previously evaluated the individual predictiveness of the rules used in LARGO and found that while some could be used to classify students by performance few of rules were strongly predictive [7]. This assessment, however, is qualified by the fact that the rules were used to give advice to the students as they worked. Thus the students flagged by the rules in the analysis either received the advice and ignored it or did not ask. Moreover the performance measures used were comprehension tests and not the production of novel arguments. Some prior researchers (e.g. [2, 1]) have discussed the relationship between student-produced argument diagrams and written essays. Those analyses, however, are purely qualitative. In more recent work we examined the relationship between basic features of student argument diagrams, such as order and size, and found that they could be used to predict students' overall grades [6]. The features



Madsen et al 2004

Claims that widespread use of the MMR vaccinne is responsible for increading incidence of autism.

Madsen claims a link between a particular vaccine and autism while Bale claims link not proven.

Bale, like Madsen, acknowledges the potential for harm.

Bale 2004

While potential risks and side-effects are known they have not been proven.

Use of mercury and other preservatives is cited by Madsen as possible cause

Weak support in that no link has been shown.

Vaccines are administered.

They will not cause autism.

Figure 1: A segment of a student-produced LASAD diagram showing a hypothesis node (lower right) and two conflicting citation nodes with a comparison arc between them.

chosen, however, do not always lend themselves to robust feedback and the grades chosen incorporate a number of criteria beyond performance at argument.

In the present work we focus solely on an *exploratory dataset* where advice was not given and *planning diagrams* in which students plan novel arguments using a domain-specific argument model and the LASAD diagramming toolkit [3]. Students in this study were *not* provided with advice nor were they annotating an existing text. LASAD supports advice through an optional JESS-based system called the AFEngine [13, 3] which we are using in present studies. As part of this work we have shown that it is possible to reliably grade student-produced argument diagrams and essays, and that the expert-assigned diagram grades can be used to predict essay performance [4]. We have also shown that it is possible to make automatic predictions of student essay grades via regression models [5]. In the present paper we focus on individual rule evaluation.

2. METHODS

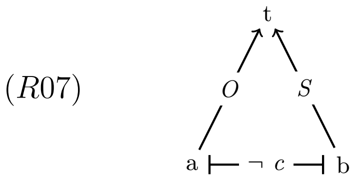
Data for this study was collected in a course on Psychological Research Methods in Fall 2011 at the University of Pittsburgh (see: [4]). Students in this course are taught study design, analysis, and ethics. The course is divided into lab sections. As part of the course students are required to conduct two empirical research projects including hypothesis formation, data collection, analysis, and writeup. Each lab jointly identifies a research topic and collaborates in data collection. The remaining aspects of planning, analysis, and writeup, are completed independently.

For the purposes of the study we augmented the traditional assignment with a graphical planning step. Once the students had completed the study design and data collection they were now required to plan their arguments graphically using the LASAD diagramming toolkit [3]. LASAD is an online tool for argument diagramming that allows for customized ontologies, peer collaboration, and annotation. The students were given a customized ontology with specialized nodes representing *hypothesis statements*, *citations*, and *claims*, and arcs representing *supporting*, *opposing*, and *undefined* relationships as well as *comparisons* between items.

Part of a representative student diagram is shown in Figure 1. The diagram shows a single hypothesis node (#25) at the lower right-hand corner. This node is supported by a citation node located on the left-hand side of the diagram (#8) and opposed by citation node (#6) at the top. These two citations are, in turn, connected to one-another by a comparison arc that states both analogies or similarities between the nodes and distinctions or differences. This structure forms a *paired counterargument with comparison*. Students were instructed to use it to express conflicting citations and to explain the source of the disagreement.

The diagrams and essays collected in the course were graded using a parallel rubric focused on the clarity, quality, persuasiveness, and other aspects of the argument. Grading was carried out by an experienced TA and reliability was tested in a separate inter-grader agreement study (see [4]). In that study we found that 5 of the 14 criteria were reliable and we focus on them below. The criteria chosen focus on: the quality of the research question (*RQ-Quality*); whether or not the hypothesis can be tested (*Hyp-Testable*); whether the author explained why the cited works relate to their argument (*Cite-Reasons*); and whether or not the hypothesis was open or untested (*Hyp-Open*). The final one measured the overall quality of the argument presented (*Arg-Quality*).

As part of this study we identified a set of 77 unique graph features for analysis. 34 of these were *simple features* such as the order and size of the diagram, the number of nodes and arcs of each type, and the amount of text in each node. Some of these were previously evaluated with legal arguments and found to be informative [6]. We also identified 43 com-



$$\left\{\begin{array}{c} t.Type \in \{``Hypothesis", ``Claim"\} \\ a.Type = ``Citation" \\ b.Type = ``Citation" \\ c.Type = ``Comparison" \end{array}\right\}$$

Figure 2: *R07: Uncompared Opposition* A simple augmented graph grammar rule that detects uncompared counterarguments. The rule shows a two citation nodes (a, & b) that have opposing relationships with a shared hypothesis or claim node (t) and do not have a comparison arc (c) drawn between them. The arcs S and O represent recursive supporting and opposing paths.

plex features that were designed to identify pedagogically-relevant subgraphs such as the paired counterarguments discussed above, unfounded hypotheses, and incorrect applications of individual arcs. These complex features were encoded as Augmented Graph Grammars and were evaluated using the AGG Engine [4]. Augmented Graph Grammars are a rule formalism that supports complex field constraints such as text criteria and multiple sub-fields as well as comparisons between nodes. The features were identified by domain experts based upon examination of previously-collected diagram and essay data and *a-priori* assumptions about the structure of good and poor data.

One such rule is shown in Figure 2. This rule detects *R07: Uncompared Opposition*. This occurs when two citation nodes, a & b, disagree about a shared hypothesis or claim node t with one opposing it and the other supporting and the user has *not* drawn a comparison arc between them to explain the disagreement. Students were trained to represent opposing citations with distinct nodes and then to explain that disagreement via a comparison arc. This rule tracks violations of that guidance and would not match the subgraph shown in Figure 1 which has a comparison arc.

For each diagram we collected a frequency count for the individual features and performed a series of pairwise comparisons mapping the observed frequencies to the paired essay grades. The comparisons were made using three candidate distributions: raw count, logarithmic, and binary. For analysis purposes the essay grades were normalized to a range of $(0-1)$. The statistical comparisons were made using Spearman's ρ, a nonparametric measure of correlation in the range $(-1 \leq \rho \leq 1)$ with -1 indicating a strong monotonically decreasing relationship and 1 a strong increasing one. ρ was chosen as it is robust in the face of nonlinear relationships.

3. RESULTS

We collected a total of 132 original diagrams and 125 essay introduction drafts from the course. After dealing with dropouts and incomplete assignments we obtained 105 unique diagram-essay pairs 31 of which were authored by individuals with the remaining 74 were authored by a team of 2-3. Of the features tested we found that eight of the simple features had statistically- or marginally-significant correlations between one or more of the grades and that all of the grades were significantly correlated with at least one simple feature. The weakest such correlation was between the *Order* or the number of nodes in the diagram ($\ln|G_n|$) and the grade *Cite-Reasons* (log: $\rho = 0.162, p < 0.098$). This is consistent with our expectations given the work in [6]. The strongest such correlation was between the presence of a hypothesis node (*Elt Hypothesis*) and the testability of the hypothesis (*Hyp-Testable*) (bin: $\rho = 0.383, p < 0.001$).

We found that 19 of the complex features were correlated with at least one of the grades and again all the questions were related to at least one feature. Here the weakest correlation was between the absence of a hypothesis node and *Cite-Reasons* (bin: $\rho = -0.166, p < 0.09$). The strongest correlation was between the amount of *uncompared opposition* (See Fig 2), that is the number of opposing citations without a comparison arc, and the grade for the openness of the hypothesis (*Hyp-Open*) (log: $\rho = 0.396, p < 0.001$).

Of these correlations some, such as the correlation between the presence of a hypothesis and the testability mentioned above, validate our *a-priori* assumptions. Hypothesis nodes are central to the entire argument and the empirical results validate their importance. We also found that the number of *paired counterarguments*, conflicting supporting and opposing nodes of the type shown in Figure 1, was positively correlated with the openness of the *Hyp-Open* ($(raw)\rho = 0.323, p < 0.001$). This was consistent with the instructions given to the students about how disagreements were to be presented and thus these results are promising. Moreover, we found that the presence of hypothesis nodes with no connection to a citation (*undefined ungrounded hypothesis* is negatively correlated with both *Cite-Reasons* (log: $\rho = -0.226, p < 0.02$), and *Arg-Quality* (log: $\rho = -0.219; p < 0.025$). This too reflects the need to ground the discussion in the appropriate literature.

While these results were positive a number of other significant correlations did not validate our assumptions. One notable example was the positive correlation between the uncompared opposition and *Hyp-Open* discussed above. We also found that the presence of unopposed hypotheses positively correlated with *Hyp-Testable* (log: $\rho = 0.196, p < 0.045$), and that the number of unfounded claims, claim nodes not connected to a citation, was positively correlated with *Hyp-Testable* (log: $\rho = 0.225, p < 0.021$).

4. ANALYSIS & CONCLUSIONS

Our goal in this research was to test the individual empirical validity of our *a-priori* diagram rules and to demonstrate the utility of empirical validation for ill-defined domains. To that end we collected a set of planning diagrams in a Research Methods course paired with graded argumentative essays. Unlike prior studies these diagrams were collected in an exploratory system where no automated advice was given to the students, and the argumentative essays were both novel, and graded independently with a focus on specific features of the arguments and their gestalt quality. In general, we found that some but not all of the features were significantly correlated with subsequent grades. Those correlations, however, were not always consistent with the *a-priori* assumptions that motivated their construction.

These counter-intuitive results are difficult to explain and highlight the central challenge of data-driven rule validation. The *Paired Counterarguments*, for example, are a positive diagram structure. Students were instructed to use them to indicate disagreement and, by extension, the openness of the hypothesis and research question. The rule defining them, however, is less precise than the rule defining uncompared opposition shown in Figure 2. Paired counterarguments omit any test for the comparison arc *c*. Thus all subgraphs detected by the latter rule will also be detected by the former. Given that the students were explicitly instructed to explain any opposing citations we expected that the latter rule would be strongly negative while the former would have a weak correlation at best. The fact that this was not the case suggests that either the students violated the instructions consistently or that the data is otherwise skewed, or that the rules are insufficiently precise to capture our *a-priori* assumptions.

We plan to address these limitations in future work by conducting a more detailed analysis of the existing data and by testing *conditional correlations*. In the case of uncompared opposition, for example, the author must have paired counterarguments in order to have the option of drawing a comparison arc. Thus it may be more informative to evaluate the impact of the uncompared opposition on graphs where paired counterarguments are found. This form of conditioning may address the generality of the rules but may require a larger dataset for us to draw robust conclusions. We also plan to test this approach on other related datasets that are presently being collected and to examine the alignment between the diagrams and essays. While the two elements were produced and graded separately, we anticipate that a more detailed tagging process should identify direct mappings between the diagram components and the essay structures. These mappings, if found, should enable us to perform a more direct evaluation of the role that individual structural elements play in the subsequent essay quality.

Acknowledgments

NSF Award 1122504 “DIP: Teaching Writing and Argumentation with AI-Supported Diagramming and Peer Review” Kevin D. Ashley PI, Chris Schunn & Diane Litman co-PIs.

5. REFERENCES

[1] Chad S. Carr. Using computer supported argument visualization to teach legal argumentation. pages 75–96. Springer-Verlag, London, UK, 2003.

[2] Evi Chryssafidou and Mike Sharples. Computer-supported planning of essay argument structure. In *Proceedings of the 5th International Conference of Argumentation*, June 2002.

[3] Frank Loll and Niels Pinkwart. Lasad: Flexible representations for computer-based collaborative argumentation. *Int. J. Hum.-Comput. Stud.*, 71(1):91–109, 2013.

[4] Collin F. Lynch. The Diagnosticity of Argument Diagrams, 2014. (defended January 30th 2014).

[5] Collin F. Lynch, Kevin D. Ashley, and Min Chi. Can diagrams predict essays? In Stefan Trausen-Matu and Kristy Boyer, editors, *Intelligent Tutoring Systems, 12th International Conference, ITS 2014, Honolulu, Hawai'i, USA*, Lecture Notes in Computer Science. Springer, 2014. (In Press).

[6] Collin F. Lynch, Kevin D. Ashley, and Mohammad H. Falakmassir. Comparing argument diagrams. In Burkhard Schäfer, editor, *Legal Knowledge and Information Systems - JURIX 2012: The Twenty-Fifth Annual Conference, University of Amsterdam, The Netherlands, 17-19 December 2012*, volume 250 of *Frontiers in Artificial Intelligence and Applications*, pages 81–90. IOS Press, 2012.

[7] Collin F. Lynch, Kevin D. Ashley, Niels Pinkwart, and Vincent Aleven. Argument graph classification with genetic programming and c4.5. In Ryan Shaun Joazeiro de Baker, Tiffany Barnes, and Joseph E. Beck, editors, *EDM*, pages 137–146. www.educationaldatamining.org, 2008.

[8] Collin F. Lynch, Kevin D. Ashley, Niels Pinkwart, and Vincent Aleven. Concepts, structures, and goals: Redefining ill-definedness. *International Journal of Artificial Intelligence in Education*, 19(3):253–266, 2009.

[9] Collin F. Lynch, Kevin D. Ashley, Niels Pinkwart, and Vincent Aleven. Ill-defined domains and adaptive tutoring technologies. In Paula J. Durlach and Alan M. Lesgold, editors, *Adaptive Technologies for Training and Education.*, chapter 9, pages 179–203. Cambridge, UK: Cambridge University Press., 2012.

[10] Antonija Mitrovic and Amali Weerasinghe. Revisiting the definition of ill-definedness and the consequences for itss. In *Proceedings of AIED2009*, 2009.

[11] Niels Pinkwart, Kevin D. Ashley, Collin F. Lynch, and Vincent Aleven. Evaluating an intelligent tutoring system for making legal arguments with hypotheticals. *International Journal of Artificial Intelligence in Education*, 19(4):401–424, 2009.

[12] Niels Pinkwart, Collin F. Lynch, Kevin D. Ashley, and Vincent Aleven. Re-evaluating largo in the classroom: Are diagrams better than text for teaching argumentation skills? In Beverly Park Woolf, Esma Aïmeur, Roger Nkambou, and Susanne P. Lajoie, editors, *Intelligent Tutoring Systems*, volume 5091 of *Lecture Notes in Computer Science*, pages 90–100. Springer, 2008.

[13] O. Scheuer, S. Niebuhr, T. Dragon, B. M. McLaren, and N. Pinkwart. Adaptive support for graphical argumentation - the lasad approach. IEEE Learning Technology Newsletter 14(1), p. 8 - 11, 2012.

[14] Daniel D. Suthers. Representational guidance for collaborative inquiry. In *Arguing to learn: Confronting cognitions in computer-supported collaborative learning environments*, page 27âĂŞ46. 2003.

[15] Amali Weerasinghe, Antonija Mitrovic, and Brent Martin. Towards individualized dialogue support for ill-defined domains. *International Journal of Artificial Intelligence in Education*, 19(4):357–379, 2009.

Entropy: A Stealth Measure of Agency in Learning Environments

Erica L. Snow
Arizona State University
Tempe, AZ, USA
Erica.L.Snow@asu.edu

Mathew E. Jacovina
Arizona State University
Tempe, AZ, USA
Mathew.Jacovina@asu.edu

Laura K. Allen
Arizona State University
Tempe, AZ, USA
LauraKAllen@asu.edu

Jianmin Dai
Arizona State University
Tempe, AZ, USA
Jianmin.Dai@asu.edu

Danielle S. McNamara
Arizona State University
Tempe, AZ, USA
Danielle.McNamara@asu.edu

ABSTRACT

This study investigates variations in how users exert agency and control over their choice patterns within the game-based ITS, iSTART-2, and how these individual differences relate to performance. Seventy-six college students interacted freely with iSTART-2 for approximately 2 hours. The current work captures and classifies variations in students' behavior patterns using three novel statistical techniques. Random walk analyses, Euclidean distances, and Entropy measures indicated that students who interacted exhibiting more controlled and systematic patterns demonstrated higher quality strategy performance compared to students who interacted with the system in more disordered fashions. These results highlight the potential for dynamical analyses as stealth assessments indicative of students' degree of agency within adaptive learning environments.

Keywords

Intelligent Tutoring Systems, dynamical analysis, agency, strategy performance, game-based learning, stealth assessment

1. INTRODUCTION

Adaptive environments often incorporate various elements (e.g., customization, games) that promote user control as a means to enhance motivation, performance, and learning outcomes [1-2]. When users take control and exert influence (e.g., through choices) over their situation or environment, they are said to have a strong feeling of agency [3]. Agency has been shown to be a critical component of students' engagement and subsequent learning of academic material. Indeed, it is a widely accepted belief in the classroom that giving students control promotes their motivation and subsequent learning [4].

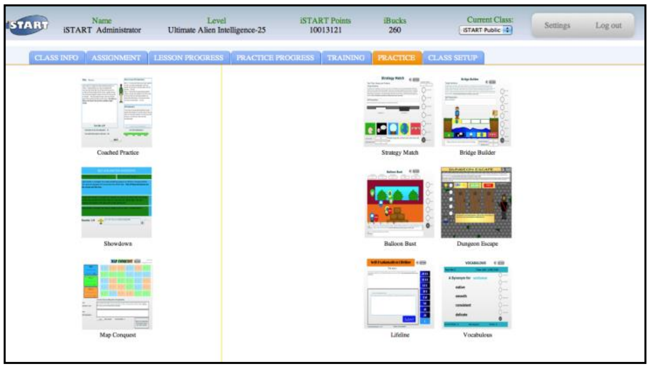
Many game-based learning environments are designed to further enhance users' feeling of agency. Games allow individuals to exert influence over the learning environment by leveraging the mechanics and features found in popular, non-educational video games [11]. For instance, well-designed games frequently present players with interesting choices, leading to increased engagement and persistence [5]. These games also frequently allow players to customize the visual appearance of the features (e.g., player's avatar in World of Warcraft), which has been associated with increased immersion and intention to replay a game [6]. By adding these elements of agency throughout game-based systems, researchers attempt to increase engagement and enjoyment, and indirectly improve learning outcomes [7].

Despite these theoretical and design considerations, research suggests that individuals vary in their ability to exert control over their environment [8]. These behavioral variations, however, are often hard to capture. One proposed way to measure individual differences in controlled behavior is through the use of dynamical analysis techniques. These methodologies focus on the fine-grained and complex behaviors that emerge over time. Dynamical methodologies focus on time as the critical variable, thus offering scientists a unique means of classifying variation in students' behavior patterns when they are given agency within an adaptive system. These methodologies have previously been used to investigate nuanced and fine-grained behavior patterns within various adaptive systems [10].

The work presented here builds upon previous research by employing three novel dynamical methodologies to act as a stealth measure of how variations in behavioral patterns emerge when students are presented with high levels of control (i.e., many choices) within a game-based environment. Although this level of control should lead to high levels of perceived agency for students, some may struggle to exert control over such an open environment. In this study, we examine how students interact with the game-based system iSTART-2, in concert with subsequent learning outcomes associated with those behavior patterns.

The Interactive Strategy Training for Active Reading and Thinking-2 (iSTART-2) system was designed to improve students' reading comprehension by providing them with strategy instruction [11]. More specifically, the iSTART-2 system trains students to use self-explanation strategies while reading challenging science texts. This system has been shown to improve students' reading comprehension ability [11].

iSTART-2 utilizes a game-based environment that was specifically designed to increase students' engagement and persistence, factors that have been shown to positively affect learning [11].



Name: iSTART Administrator
Level: Ultimate Alien Intelligence-25
iSTART Points: 10013121
iBucks: 260
Current Class: iSTART Public

Figure 1. Screen shot of iSTART-2 Selection Menu

The iSTART-2 system consists of two phases: training and practice. Within the training phase, students are introduced to and provided examples of self-explanation strategies. After training, students are transitioned to the practice phase when they are free to interact with the game-based interface embedded within the system (see Figure 1).

There are four types of game-based features within iSTART-2. *Generative practice games* require students to write self-explanations in response to target sentences in science texts. *Identification mini-games* provide example self-explanations and ask students to indicate which previously learned strategy was used to generate the self-explanation. *Personalizable features* allow students to customize the color and appearance of the system interface. *Achievement screens* offer students a summary of their performance levels within the system. In the current study, students were free to interact with these features in any way they saw fit.

The current study uses three novel statistical techniques—random walks, Euclidean distances, and Entropy scores—to categorize nuances in students' choice patterns that emerge while they engage within the iSTART-2 interface. Using these methodologies, we investigated students' choice patterns and the impact of variations in those patterns on learning outcomes (i.e., self-explanation quality) within the context of iSTART-2.

2. METHOD

Participants in the current study included 76 college students who were from a large university campus in the Southwest United States. The students were, on average, 18 years of age, with a mean reported grade level of college freshman. Of the 76 students, 58% were male, 55% were Caucasian, 22% were Asian, 7% were African-American, 10% were Hispanic, and 6% reported other nationalities.

Students in this study completed one 3-hour session that consisted of a pretest, strategy training, game-based practice within iSTART-2, and a posttest. During the pretest, students answered a battery of questions to assess their prior attitudes and motivation. During training, students watched a series of videos that instructed them on various self-explanation strategies and their applications. After training, students were transitioned into the game-based practice portion of the experiment. In this section, students were exposed to the game-based menu within iSTART-2 (see Figure 1) where they were allowed to interact freely within the system interface. Finally, at posttest, students completed a battery of questionnaires similar to those in the pretest.

2.1 Measures

2.1.1 Strategy performance

During game-based practice, students' generated self-explanation quality was measured using the iSTART-2 algorithm, which assigns scores that range from 0 (poor) to 3 (good). This algorithm incorporates both latent semantic analysis (LSA) [12] and word-based measures and has been shown to be reliable and comparable to expert human raters across a variety of texts [for more information see 13].

2.1.2 System Interaction Choices

Within the iSTART-2 system, students can chose to interact with a variety of features that fall into one of the four types of game-based feature categories: *generative practice games, identification mini-games, personalizable features,* and *achievement screens.* All interactions within iSTART-2 were logged by the system and then categorized as one of these four types. Tracking students' choices with these four distinct categories of features allows us to investigate patterns in students' choices across and within each type of interaction.

3. QUANTITATIVE METHODS

To examine variations in students' behavior patterns within iSTART-2, random walks, Euclidean distances and Entropy analyses were conducted to investigate how variations in students' choice patterns impact learning outcomes (i.e., self-explanation quality) within the context of iSTART-2. The following section provides a description and explanation of random walks, Euclidean distance, and Entropy analyses.

Random walks are mathematical tools that provide visualization of patterns that manifest within categorical data across time [14]. In the current study, we used random walks to visualize and capture the fluctuations within students' interaction patterns with iSTART-2 by examining the sequential order of students' interactions with the four types of game-based features (i.e., generative practice games, identification mini-games, personalizable features, and achievement screens). Each of these game-based features was given an assignment along an orthogonal vector in an X, Y scatter plot. These assignments are as follows: generative practice games (-1,0), identification mini-games (0,1), personalizable features (1,0), and achievement screens (0,-1). It is important to note that these vector locations are random and not associated with any qualitative value. This methodology has previously been used to trace students' interaction patterns within the game-based ITS, iSTART-ME [10].

To generate a unique walk for each student, we placed an imaginary particle at the origin (0,0). Then, every time the student chose to interact with a game-based feature, the particle moved in a manner consistent with the vector assignment. The use of these vectors allows us to assign a movement to students' choices within the system. The combination of these movements yields a continual pattern or "walk" for each student's interactions within iSTART-2.

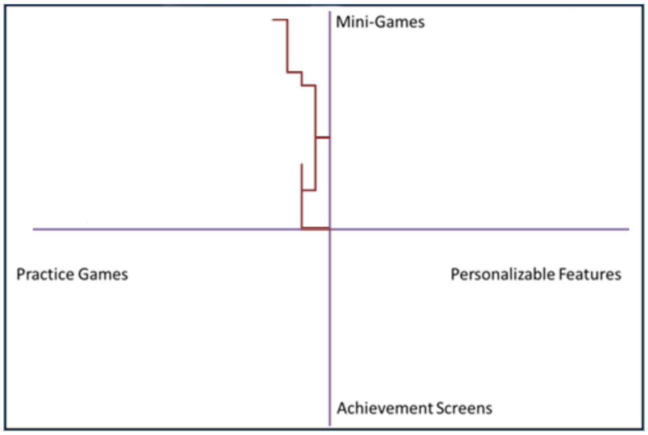


Figure 2. Actual random walk for one participant

Figure 2 is an actual *walk* from the current study. This walk is a visualization of one student's interactions within the system. The trajectory of the walk suggests that this student interacted more frequently with the identification mini-games. Within these random walks, there are fluctuations and nuances that may inform how controlled students' choice patterns were. To understand how students' patterns changed, distance time series were constructed for each student by calculating a measure of Euclidean distance. This calculation was measured from the origin (coordinates 0,0) to each step (see equation 1). Within this equation, y represents the particle's place on the y-axis, x represents the particle's place on the x-axis, and *i* represents the *i*th step within each student's walk:

$$\text{Distance} = \sqrt{(y_i - y_0)^2 + (x_i - x_0)^2} \qquad (1)$$

Euclidean distance was calculated for each step in a student's walk, which produced a distance time series. These distance time series can then be used to reflect the degree to which students controlled their pattern of choices. That is, if students showed a systematic pattern in their walk, the distance time series would have reflected this controlled pattern through coordinated *steps*.

Students' propensity to engage with the system in an ordered fashion was calculated using Entropy [15]. Entropy is a statistical analysis that has previously been used across a variety of domains as a way to measure random, controlled, and ordered processes [16]. Hence, within the current study, Entropy provides a measure of how students' choice patterns reflected controlled versus ordered processes.

Entropy was calculated using the distance time series produced from students' random walks and Euclidean distances (see equation 2). Within Equation 2, $P(x_i)$ represents the probability of a given state. This means that the Entropy for student X is the inverse of the sum of products calculated by multiplying the probability of each achieved state by the natural log of the probability of that state.

$$\text{H(x)=-}\sum_{i=0}^{N} P(x_i)(\log_e P(x_i)) \qquad (2)$$

The Entropy formula in Equation 2, captures the amount of order (or disorder) present in a specific time series. Within the context of the current study, a low Entropy score suggests highly organized choice patterns, whereas high Entropy suggests disorganized choice patterns.

4. RESULTS

4.1 Entropy

This study examined how students' patterns of interactions with game-based features influenced the quality of their generated self-explanation quality. To characterize how students interacted with the system, Entropy was calculated using Euclidean distances generated from each student's unique random walk. These Entropy scores suggested that students varied considerably from controlled to disordered (range =1.32 to 2.32, *M*=1.83, *SD*=0.24; skewness = -.22; kurtosis = -1).

4.2 Interaction Choices

To examine the relation between Entropy scores (i.e., measure of order or disorder) and students' frequency of interaction choices, we calculated Pearson correlations. Students' Entropy scores were not significantly related to their frequency of interactions with generative practice games (*r*=-.14, *p*=.23), identification mini-games (*r*=.05, *p*=.65), personalizable features (*r*=.03, *p*=.77), or achievement screen views (*r*=.09, *p*=.43). Thus, patterns in students' choices were not related to any specific feature within iSTART-2.

4.3 Learning and System Performance Outcomes

To examine the effects of agency on performance during practice, a median split was performed on students' Entropy scores to profile students according to their patterns of interactions. This median split resulted in the creation of two groups: ordered students (*M*=1.6, *SD*=.15) and disordered students (*M*=2.0, *SD*=.11). Differences between the ordered and disordered students' self-explanation quality during practice were examined using a one-way ANOVA. This analysis revealed that ordered students generated higher quality self-explanations (*M*=1.8, *SD*=.55), compare to disordered students (*M*=1.6, *SD*=.44), *F*(1,74)=4.78, *p*=.03, η^2=.10[1].

A similar one-way ANOVA was used to examine differences between ordered and disordered students' game performance within iSART-2. These results revealed that ordered students also earned significantly more trophies (*M*=1.4, *SD*=.22) while playing the practice games than disordered students (*M*=.6, *SD*=.09), *F*(1,74)= 9.17, *p*=.003, η^2=.11. Together, results from both ANOVA analyses indicate that students who engaged in a more ordered behavior pattern showed significantly better game performance relative to students who engaged in a disordered behavior pattern.

5. DISCUSSION

Researchers have argued that enhancing feelings of agency by introducing choice influences students' engagement and ultimately impacts learning outcomes [4]. However, students vary in their ability to effectively control and regulate their behaviors when presented with this freedom [9]. This variability is often missed when researchers use static measures (e.g., self-reports)

[1] Similar trends are found using Entropy as a continuous variable to predict self-explanation quality during practice.

alone. Dynamical analyses offer one way to capture variances in students' ability to control their behaviors when they are presented with additional opportunities to exert agency.

The current study made use of three novel dynamical methodologies by employing random walks, Euclidean distances, and Entropy analyses in an attempt to capture each student's unique interaction pattern within iSTART-2. The current analysis is one of the first to use Entropy as a means to provide a stealth assessment of students' patterns of interactions within a tutoring environment. These scores reveal trends across time that are suggestive of the degree to which students exerted agency within the iSTART-2 system.

Findings from the current analyses fall in line with previous work that has shown that students' ability to regulate and control their learning behaviors has a positive impact on learning outcomes [9]. Students who acted in a more controlled fashion generated higher quality self-explanations during practice. Interestingly, students' controlled patterns of interactions were not related to any specific game-based feature. This indicates that it is not about what students choose to do, but how they choose to do it. Thus, students' ability to effectively control their behaviors when presented with a considerable amount of choice seems to be important for immediate learning outcomes. This is especially important within game-based environments where students are often given considerable control over their learning trajectories [10]. Students who exhibit controlled behaviors are likely to be experiencing and benefitting from strong feelings of agency, as intended by the design of these learning environments.

6. ACKNOWLEDGMENTS

This research was supported by the Institute of Education Sciences, US Department of Education, through Grant R305A130124. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the IES.

7. REFERENCES

[1] Cordova, D. I., and Lepper, M. R. 1996. Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology*, 88, (1996), 715– 730.

[2] Snow, E. L., Jackson, G. T., Varner, L. K., and McNamara, D. S. 2013. Investigating the effects of off-task personalization on system performance and attitudes within a game-based environment. In S. K. D'Mello, R. A. Calvo, and A. Olney (Eds.), *Proceedings of the 6th International Conference on Educational Data Mining*, (Memphis, Tennessee, July 6 -9, 2013), Springer Berlin Heidelberg, 272-275.

[3] Metcalfe, J., Eich, T. S., and Miele, D. B. 2013. Metacognition of agency: Proximal action and distal outcome. *Experimental Brain Research*, 229, (2013), 485–96.

[4] Flowerday, T., and Schraw, G. 2000. Teachers' beliefs about instructional choice: A phenomenological study. *Journal of Educational Psychology*, 92, (2000), 634–645.

[5] Schønau-Fog, H., and Bjørner, T. 2012. "Sure, I would like to continue": A method for mapping the experience of engagement in video games. *Bulletin of Science, Technology* & Society, 32, (2012), 405–412.

[6] Teng, C. I. 2010. Customization, immersion satisfaction, and online gamer loyalty. *Computers in Human Behavior*, 26, (2010), 1547–1554.

[7] Wouters, P., van Nimwegen, C., van Oostendorp, H., and van der Spek, E. D. 2013. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, 105, (2013), 249–265. doi:10.1037/a0031311

[8] Katz, I., Assor, A., Kanat-Maymon, Y., and Bereby-Meyer, Y. 2006. Interest as a motivational resource: Feedback and gender matter, but interest makes the difference. *Social Psychology of Education*, 9, (2006), 27– 42.

[9] Zimmerman, B. J. 1990. Self-regulated learning and academic achievement: An overview. *Educational* psychologist, 25, (1990), 3-17.

[10] Snow, E. L., Likens, A., Jackson, G. T., and McNamara, D. S. 2013. Students' walk through tutoring: Using a random walk analysis to profile students. In S. K. D'Mello, R. A. Calvo, and A. Olney (Eds.), *Proceedings of the 6th International Conference on Educational Data Mining*, (Memphis, Tennessee, July 6 -9, 2013), Springer Berlin Heidelberg, 276-279.

[11] Jackson, G. T., and McNamara, D. S. 2013. Motivation and performance in a game-based intelligent tutoring system. Journal of Educational Psychology, 105, (2013), 1036-1049.

[12] Landauer, T., McNamara, D. S., Dennis, S., and Kintsch, W. (Eds.). 2007. *Handbook of Latent Semantic Analysis.* Erlbaum, Mahwah, NJ.

[13] McNamara, D. S., Boonthum, C., Levinstein, I. B., and Millis, K. 2007. Evaluating self-explanations in iSTART: Comparing word-based and LSA algorithms. In T. Landauer, D .S. McNamara, S. Dennis, and W. Kintsch (Eds.), *Handbook of Latent Semantic Analysis*. Erlbaum, Mahwah, NJ, 227-241.

[14] Benhamou, S., and Bovet, P. 1989. How animals use their environment: a new look at kinesis. *Animal Behavior*, 38, (1989), 375-383.

[15] Shannon, C. 1951. Prediction and Entropy of printed English. *Bell Systems Technical Journal*, 27, (1951), 50-64.

[16] Grossman, E. R. F. W. 1953. Entropy and choice time: The effect of frequency unbalance on choice-response. *Quarterly Journal of Experimental Psychology*, 5, (1953), 41-51.

Error Analysis as a Validation of Learning Progressions

Brent Morgan
University of Memphis
202 Psychology Bldg
Memphis, TN 38152
+1 (901) 338-5041
brent.morgan@memphis.edu

William Baggett
University of Memphis
Department of Computer Science
Memphis, TN 38152
+1 (901) 678-5465
wbaggett@memphis.edu

Vasile Rus
University of Memphis
375 Dunn Hall
Memphis, TN 38152
+1 (901) 678-5259
vrus@memphis.edu

ABSTRACT

Learning progressions (LPs) are a recent educational theory pertaining to student modeling. LPs argue that students with equal test scores may nonetheless have different conceptualizations of the material, with varying degrees of maturity. However, there is little empirical validation for LPs. To this end, we mapped two physics LPs (one predefined, one described in the paper) onto the answer choices of a popular conceptual physics test (the Force Concept Inventory; FCI). We then assessed 444 high school physics students using a pretest-posttest design. Students with more mature incorrect answers on the pretest performed better on the posttest than their less mature counterparts. We discuss implications for theorists and practitioners in learner modeling.

Keywords

Learning Progressions, Learner Models, FCI, Intelligent Tutoring Systems

1. INTRODUCTION

Students arrive at a learning session with a wide variety of backgrounds, skills, knowledge, and abilities. These individual differences imply that each person cannot be expected to learn the same way and at the same speed as another [3]. A set of learner characteristics which impact learning is referred to as a learner model. A learner model allows instruction to be tailored to each individual student with the goal of maximizing learning for that student. As a student progresses through a learning session, her learner model is updated based on the quality of her contributions.

One novel approach to learner modeling relies on a research framework called Learning Progressions (LPs), developed recently by the science education research community as a way to increase adaptivity in traditional instruction. LPs have been defined as "descriptions of the successively more sophisticated ways of thinking about an idea that follow one another as students learn" [12, 13]. LPs provide a promising means to organize and align content, instruction, and assessment strategies to give students the opportunity to develop deep and integrated understanding of concepts. LPs can be viewed as incrementally more sophisticated ways to think about an idea that emerge naturally while students move toward expert-level understanding of the idea [4]. LPs define qualitatively different levels of understanding of big ideas. The levels can be sequentially related or the relation could be more complex. For instance, topic A may develop the ideas from a less sophisticated topic B but also connect to other topics. LPs are organized in levels of understandings which reflect major milestones in learners' journey towards mastery. The lower level, called the Lower Anchor, represents naïve thinking typically associated with novices. The top level, called the Upper Anchor, represents the mastery/expert level of understanding.

Each student's LP level can presumably be determined from the quality of her contributions, just as with a generic learner model. Traditional assessment typically treats all wrong answers as equally wrong. An LP framework, in contrast, would argue that not all wrong answers are equivalent, and that two answers, while both incorrect, may reflect vastly different milestones on the path to mastery. Thus, students with similar assessment scores may still have vastly different understandings of the topic. In this case, the LP theory would expect a student at a higher LP level would reach mastery faster than the student at a lower LP level. Despite an increase in the popularity of learning progressions, there is surprisingly scant empirical evidence to support these claims [16].

In this paper, we describe a new LP (i.e., relative LP levels of various responses) for two physics topics (Force & Motion, Newton's 3rd Law) on a popular physics assessment tool, the Force Concept Inventory (FCI). Each incorrect answer choice was classified as a higher LP level answer or lower LP level answer. Using pretest and posttest FCI scores collected from 444 high school physics students, we also report a preliminary investigation into the efficacy of these FCI LP maps. A learning progression framework would predict that students with generally 'better' incorrect answers on a given pretest topic would be closer to mastery than those with 'worse' incorrect answers, and this would be reflected by higher posttest scores for those closer to mastery. If this is the case, it would provide data-driven (or bottom-up) support of our conceptually-driven (top-down) LP map.

2. MATERIALS

2.1 Force Concept Inventory

The Force Concept Inventory (FCI) is a 30-item multiple-choice "test" designed to assess students' understanding of the *most basic* concepts in Newtonian mechanics [7]. The FCI presents students with various situations and asks them to choose between Newtonian explanations for the phenomena, versus common-sense alternatives [8]. The FCI has been widely used to measure learning in introductory physics courses. For example, Hake [6] in combination with Coletta and Phillips [2] used the FCI to measure learning in 73 university and college introductory physics classes. Its popularity among researchers was a major motivation for developing an LP map for the FCI as part of our DeepTutor project whose aim is to develop the first intelligent tutoring system based on learning progressions [15]. Another attraction was that the FCI was designed to identify known misconceptions that students often possess [8]. The vast majority of these "lures" can easily be classified and incorporated into an LP framework.

2.2 Developing LPs

The FCI covers multiple concepts in Newtonian mechanics, including Free Fall Near Earth, Circular Motion, and Newton's 3rd Law. The predominant concept, however, is Force & Motion. The Force & Motion LP used for this paper was identical to the one developed by Alonzo and Steedle [1]. The higher levels were defined so as to meet national standards for 8th grade students. The 8th grade Force & Motion standards are applicable to the high school standards and match the top level of understanding expressed in the FCI. The lower levels were then populated by compiling student's ideas about Force & Motion reported in the literature. These ideas were then ordered by relative difficulty. The LP was then iteratively revised based on data collected from physics novices.

Although the FCI is predominantly focused on Force & Motion, it does address other Newtonian concepts, such as Newton's 3rd Law, which are not mapped by Alonzo and Steedle [1]. Hence, a Newton's 3rd Law Learning Progression was developed with the direction of two physics professors. The method used to develop this LP was based on Alonzo and Steedle's [1] process described above.

First, we defined the top level of the hypothetical Newton's Third Law LP as the knowledge needed to articulate and apply Newton's Third Law. The knowledge of Newton's Third Law specified by the top level of our LP matches that specified for grade levels 9-12 in the National Science Education Standards [11]. The lower levels were student's ideas about Newton's Third Law reported in the literature [e.g., 9, 10, 17] and ordering these ideas based on suggestions from the literature and/or the intuitions of two physics professors regarding the relative difficulty.

Next, we collected responses from 30 paid workers from Amazon Mechanical Turk, each of whom answered 22 open-ended Newton's 3rd Law questions. The responses were coded according to the LP, and refinements to the LP were made as necessary to accommodate student's responses. Table 1 presents the revised LP for Newton's Third Law.

A team of two physics professors and two authors of this paper collaborated to map each of five answer choices from each of nine FCI questions according to the LPs. The LP map is shown in Table 2. Five of the questions corresponded to the Force & Motion topic, and four addressed Newton's 3rd Law. These questions were specifically selected for this analysis for two reasons. First, each question and answer choice was related only to one specific topic. Second, the incorrect answer choices exhibited a distinct and unambiguous separation in quality of comprehension according to the physics professors.

Table 1. Newton's 3rd Law LP

5	The student understands that all forces arise out of an interaction between two objects and that these forces are equal in magnitude and opposite in direction.
4	The student identifies equal force pairs, but indicates that both forces act on the same object. (For the example of a book at rest on a table, the downward gravitational force exerted by the earth on the book and the upward normal force exerted by the table on the book are identified as an action-reaction pair.)
3	The student uses the effects of a force as an indication of the relative magnitudes of the forces in an interaction.
2.5	The student indicates that the forces are equal because of the properties of the objects involved.
2	The student indicates that the forces in a force pair do not have equal magnitude because the objects are dissimilar in some property (e.g., bigger, stronger, faster).
1	The student believes that inanimate/passive objects cannot exert a force.
0	No statement about relevant interaction forces or Newton's 3rd Law

Table 2. LP map for upper and lower levels

Newton's Third Law Question and Answer maps

	N3L1	N3L2	N3L3	N3L4
Upper	a, d	b, c	b, c	d
Lower	b, c	d, e	d, e	a, b, c

Force & Motion Question and Answer maps

	FM1	FM2	FM3	FM4	FM5
Upper	a, d	c, e	d	a, b	a, b
Lower	c, e	b, d	a, b, e	c, d	d, e

note: We have removed the actual question numbers to avoid making the FCI answer key public knowledge. Please contact one of the authors for the actual FCI question numbers.

2.3 Data Collection

We administered the FCI to 444 students at three public and two private high schools in the mid-south region, across six teachers and 26 classrooms. The students completed the FCI twice, once at the beginning of the semester (pretest) and once at the end (posttest). Between those two time periods, each classroom covered topics relevant to the FCI, though individual course content varied. Students completed the FCI via provided scantron sheets, which were then collated and processed. The results of the scantron sheets were then compared to direct markings on the actual FCI test in the case of blank or unidentifiable scantron responses. There were five students with perfect scores on the five Force & Motion questions and 19 students with perfect scores on the four Newton’s Third Law questions. These students were not included in the respective analyses below.

3. RESULTS

Prior student knowledge was assessed using the initial administration of the FCI (pretest). The mean proportion score of the FCI pretest was 0.26, with a standard deviation of 0.15 (see Table 3). Six students recorded the minimum observed score (1/30), whereas one student attained a perfect score (30/30). A one-sample t-test indicated the mean pretest score was higher than chance (0.2), $t(443) = 7.57$, $p< .001$, though only slightly. Low prior knowledge was ideal for our purposes, of course, providing a large sample of incorrect answers.

Table 3. FCI Pre-Post descriptives by school

Course	**N**	**Pre M (SD)**	**Post M (SD)**	***d***
Public 1 (Pu1)	116	**0.20** (0.10)	**0.22** (0.11)	0.13
Private 1 (Pr1)	25	**0.19** (0.09)	**0.29** (0.10)	1.03
Public 2 (Pu2)	94	**0.27** (0.14)	**0.42** (0.16)	0.94
Private 2 (Pr2)	128	**0.33** (0.20)	**0.47** (0.22)	0.68
Public 3 (Pu3)	81	**0.21** (0.10)	**0.29** (0.13)	0.74
Total	444	**0.26** (0.15)	**0.35** (0.19)	0.55

After assigning student answer choices to the corresponding LP levels described above, we compared the posttest performance of students with more incorrect answers corresponding to the upper level of the LP map on the pretest with students who had more incorrect answers corresponding to the lower level. For example, a student with one upper level answer and three lower level answers would be assigned to the lower level group (irrespective of number of answers correct). To be conservative, students with an equal number of upper and lower level answers were assigned to the lower level.

The comparison of posttest scores, including descriptive statistics, independent samples *t*-tests, and Cohen’s *d*, is displayed in Table 4. Across both topics, students in the upper level had higher posttest scores than students in the lower level. Additionally, the findings were associated with small to medium effect sizes. Although this provides initial support for the LP hypothesis, it is possible that these differences in posttest scores are actually being driven by prior knowledge (i.e., pretest scores). Accordingly, an analysis of covariance (ANCOVA) was used, with pretest scores as a covariate to control for prior knowledge. Even when taking pretest scores into account, there were still differences in overall posttest scores between the upper and lower levels for the Force & Motion LP, $F(1, 418) = 12.78$, $p < .001$, $\eta_p^2 = .03$. Differences on overall posttest scores between the upper and lower levels for the Newton’s 3rd Law LP were marginally significant, $F(1, 405) = 2.92$, $p = .088$, $\eta_p^2 = .01$. The marginal significance is likely due to the fact that the five Force & Motion questions are more relevant to the rest of the FCI than the four Newton’s 3rd Law questions.

Table 4. Posttest descriptives for lower vs. upper level

	Lower Level		Upper Level			
Topic	N	M (SD)	N	M (SD)	*t*	*d*
N3L	115	0.30 (0.17)	310	0.35 (0.18)	**2.50**	**0.28***
FM	131	0.30 (0.15)	308	0.37 (0.19)	**3.70**	**0.40***

* *p* =< .01; N3L = Newton’s 3rd Law; FM = Force & Motion

4. DISCUSSION

In this paper, we presented a method used to develop a novel Newton’s 3rd Law Learning Progression. This LP as well as a previously developed Force & Motion LP were then mapped onto a popular physics assessment tool, the Force Concept Inventory. FCI pretest and posttest scores were collected from over 400 high school physics students. Unlike traditional assessment, an LP-based student model assumes that not all wrong answers are equally wrong. Hence, we predicted that students whose incorrect pretest answer choices corresponded to a relatively lower LP level would be further away from mastery, and this would then be reflected in students’ posttest scores. The results provided support for this claim, and are also among the earliest evidence-based support for learning progressions in general.

It should also be noted that the FCI LP map was able to predict overall (30-item) posttest scores based only on *incorrect* pretest answers of, at most, five questions. The results were also agnostic to instruction type and quality. Relatively stronger effects were observed with the Force & Motion topic than with Newton’s 3rd Law. This is likely due to the fact that many other FCI questions draw on Force & Motion comprehension, whereas none of the other twenty-six FCI questions apply to Newton’s 3rd Law outside of the four discussed in this paper.

As this is a preliminary report, there are of course many limitations. To begin, the FCI is perhaps not an ideal tool to investigate learning progressions. It was not specifically developed with learning progressions in mind, and the answer choices for most questions do not represent all possible LP levels. For example, all 20 answer choices for the Newton’s 3rd Law topic only represented three out of the possible six LP levels. Also, many FCI questions contain answer choices which apply to topics not directly related to the topic addressed by the question (though again, none of these questions were included in this paper). Despite these flaws, however, the FCI is popular, and many researchers may be able to apply a learning progressions framework to future or even past FCI datasets [14].

Also, the analyses included in this brief report are relatively simple and not comprehensive. As such, although the findings provide support for the learning progressions hypothesis, much more evidence is needed to fully validate our Force & Motion and Newton’s 3rd Law LPs.

Given the limitations of the FCI mentioned previously, there is also a market for a comprehensive physics assessment which features answers at a variety of LP levels for each question. For example, a question with the correct answer at LP level 4 and all four incorrect answers at LP level 1 offers nothing to the student model for students with incorrect answers.

Another next step is to incorporate learning progressions into the learner models of Intelligent Tutoring Systems (ITS; [15]). One of the multiple advantages of ITSs over traditional classroom instruction is the capacity to adaptively tailor instruction to meet the needs of each and every learner [18, 5]. These systems can then be scaled up to teach many users at once. This would allow for a more thorough investigation into learning progressions, including experimental evidence as to whether adaptive instruction is beneficial for students at different LP levels. Furthermore, advances in natural language processing may allow us to detect LP differences in the text of student contributions. This could perhaps eliminate the need for a pretest for ITSs.

Finally, we encourage physics education researchers to consider incorporating learning progressions into their learner models. To that end, we are currently preparing a manuscript which will report the full LP map for all 30 FCI questions.

5. ACKNOWLEDGMENTS

This research was supported in part by Institute for Education Sciences under award R305A100875. Any opinions, findings, and conclusions or recommendations expressed in this material are solely the authors' and do not necessarily reflect the views of the sponsoring agencies.

6. REFERENCES

[1] Alonzo, A. C., and Steedle, J. T. Developing and assessing a force and motion learning progression. *Science Education*, 2009, 93, 389-421.

[2] Coletta, V. P., and Phillips, J. A. Interpreting FCI scores: Normalized gain, preinstruction scores, and scientific reasoning ability. *American Journal of Physics*, 2005, 73, 1172-1182.

[3] Corcoran, T., Mosher, F. A., and Rogat, A. 2009. Learning progressions in science: An evidence-based approach to reform. *Consortium for Policy Research in Education Report #RR-63*. Philadelphia, PA: Consortium for Policy Research in Education.

[4] Duschl, R., Maeng, S. and Sezen, A. Learning progressions and teaching sequences: a review and analysis, *Studies in Science Education*, 2011, 47, 123-182.

[5] Graesser, A. C., D'Mello, S. K., Hu, X., Cai, Z., Olney, A., and Morgan, B. AutoTutor. In P. M. McCarthy, & C. Boonthum (Eds.), Applied natural language processing and content analysis: Identification, investigation and resolution (pp. 169-187). Hershey, PA: IGI Global, 2012.

[6] Hake, R. R. Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American journal of Physics*, 1998, 66, 64-74.

[7] Halloun, I., Hake, R. R., Mosca, E. P., and Hestenes, D. 1995. Force Concept Inventory (Revised, 1995); online (password protected) at http://modeling.asu.edu/R&E.Research.html

[8] Hestenes, D., Wells, M., and Swackhamer, G. Force concept inventory. *The physics teacher*, 1992, 30, 141-158.

[9] Kolokotronis, D., and Solomonidou, C. A Step-by-Step Design and Development of an Integrated Educational Software to Deal with Students' Empirical Ideas about Mechanical Interaction. *Education and Information Technologies*, 2003, 8, 229-244.

[10] Minstrell, J. (n. d.). Facets of students' thinking. Retrieved May 12, 2014, from http://depts.washington.edu/huntlab/diagnoser/facetcode.html

[11] National Research Council. National Science Education Standards. (NSES) Washington, DC: National Academy Press, 1996.

[12] National Research Council. Systems for state science assessment. (M. R. Wilson & M. W. Bertenthal, Eds.). Washington, DC: National Academy Press, 2005.

[13] National Research Council. Taking science to school: Learning and teaching science in grades K–8. (R. A. Duschl, H. A. Schweingruber, and A. W. Shouse, Eds.). Washington: The National Academies Press, 2007.

[14] Neumann, I., Fulmer, G. W., Liang, L. L., and Neumann, K. Analyzing the FCI based on a force and motion learning progression. *Science Education Review Letters*, 2013, 8-14.

[15] Rus, V., D'Mello, S. K., Hu, X., and Graesser, A. C. Recent Advances in Conversational Intelligent Tutoring Systems, AI Magazine, 2013.

[16] Sikorski, T. R., and Hammer, D. 2010. A critique of how learning progressions research conceptualizes sophistication and progress. In K. Gomez, L. Lyons, & J. Radinsky (Eds.), *Learning in the Disciplines: Proceedings of the 9th International Conference of the Learning Sciences* (ICLS 2010) - Volume 1, Full Papers. Chicago, IL: International Society of the Learning Sciences, Chicago, IL, 2010.

[17] Smith, T. I., and Wittmann, M. C. Comparing three methods for teaching Newton's third law. *Physical Review Special Topics-Physics Education Research*, 2007, 3(2), 020105.

[18] VanLehn, K. The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 2006, 16, 227-265.

Exploration of Student's Use of Rule Application References in a Propositional Logic Tutor

Michael Eagle, Vinaya Polamreddi, Behrooz Mostafavi, and Tiffany Barnes
North Carolina State University
890 Oval Dr, Box 8206
Raleigh, NC 27695-8206
{mjeagle, vspolamr, bzmostaf, tmbarnes}@ncsu.edu

ABSTRACT

Many tutors offer students reference material or tips that they can access as needed. We have logged data about student use of references with Deep Thought logic tutor which to understand why and how references are used. We find evidence that students use these references in systematic ways that change over the course of the tutor, and can be predictive of rule application errors. We can use this information to increase our understanding of which concepts students find similar, what times during the tutor students feel the need to use references. Our goal is to eventually incorporate data-driven feedback based on when and how the references are accessed.

1. INTRODUCTION

Tooltips are messages that show up when a user hovers over a GUI element for a small amount of time. It is usually a small box with text that explains what the GUI element does [4]. In educational systems, these messages can contain hints or reference material that are intended to aid the student. Alonso at el. used tooltips to explain semantic relationships in a UML tutor [1]; the authors noted that students often used these tips for verification. White et al expanded upon the concept of tooltips by making them tangible within a augmented reality environment [6].

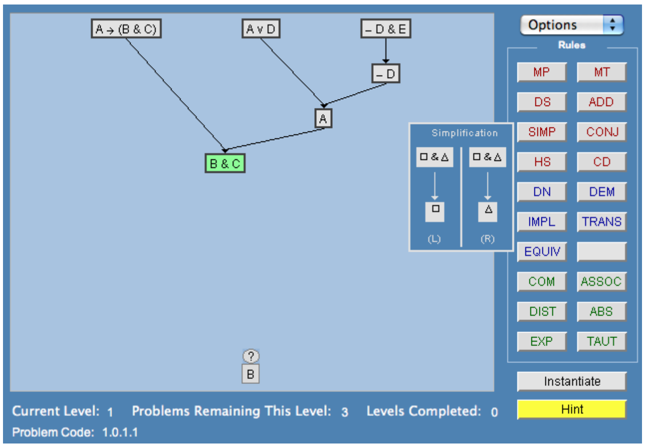
We want to further explore how these tooltips and reference material are used by students and how they affect student performance to create more effective interventions based on previously-collected student reference usage. We are inspired to add feedback and interventions based on this reference-data by the results of adding next-step hints generated from previously collected student solution-data by Stamper et al [5]. The Deep Thought logic tutor [3] provides students with logic axiom references when students hover over axiom icons within the tutor. We added logging to these hover reference actions to understand how student use of these references affect tutor performance.

We hypothesize that we will find differences in student performance metrics, such as error rate, based on their axiom-reference usage. We also expect that the way the references are used will change as students progress through the tutor.

We find that usage of references before a tutor action corresponded with a larger error rate on that action. We also find that as students progress through the tutor and as the problems increase in difficulty, they tend to use the references more often. Finally, we observe axioms that are referenced in succession indicating the rules students associate with each other and the changes in rule association as students progress through the tutor. These observations point towards trends in the collected reference data that provide better understanding of student actions and will allow us to create new, potentially better, interventions.

2. METHODS

We collect our data from the Deep Thought propositional logic tutor [3]. Each problem in Deep Thought provides the student with a set of premises and a conclusion, and asks students to prove the conclusion by applying logical axioms to the premises (see Figure 1). These logical axioms are separated into three groups based on domain concept, delineated with different colors in the tutor: logical inference rules (red), logical replacement rules (blue), and logical equivalence operations (green).



Current Level: 1 Problems Remaining This Level: 3 Levels Completed: 0
Problem Code: 1.0.1.1

Figure 1: Example screenshot of the Deep Thought tutor.

For example, in Figure 1, a student starts with premises

$A \rightarrow (B \wedge C)$; $A \vee D$; $\neg D \wedge E$ at the top of the proof window, and conclusion B at the bottom. The student performs $SIMP(\neg D \wedge E)$, applying simplification (SIMP) to the premise $\neg D \wedge E$ and derives $\neg D$. This leads to the resulting-state of $A \rightarrow (B \wedge C)$; $A \vee D$; $\neg D \wedge E$; $\neg D$.

Errors are actions performed by students that are illegal operations of rule-application, or illegal operations of the tutor. For example: If the student were to apply simplification to premise $A \vee D$ in the above example, the system would log this interaction as a rule-application error, as simplification is not a valid rule that can be used on a disjoint expression.

2.1 Logical Axiom Reference

By default, axiom references show up when a student hovers over the GUI element (button) representing the logical axiom for two seconds with the mouse pointer. References are given as an overlay pop-up window next to the corresponding axiom button, displaying the axiom name, a visual representation of the axiom pre- and post-conditions with operands and geometric shapes as variables, and valid direction of axiom application (one-way implication or two-way equivalence). Examples of these references for the axioms HS, MP, and MT are provided in Figure 2, and Figure 1 shows how these look in the main window for simplification (SIMP).

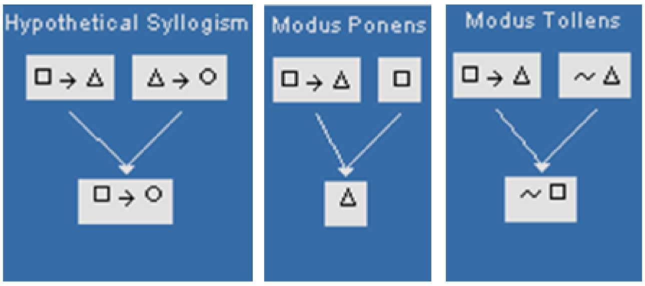


Figure 2: Hover-hints are given as an overlay pop-up window next to the corresponding axiom button, displaying the axiom name, a visual representation of the axiom pre- and post-conditions with operands and geometric shapes as variables

2.2 Data Preparation

Each row of data logged in Deep Thought represents an action performed by a student. Every reference is recorded as a separate action. For the purpose of this study, the data was pre-processed so that each transaction has a reference sequence of rules that were referenced prior to it. If there are no references preceding an action, an empty list and the corresponding 0 is recorded. In the following example (table 1, the references on row 3 and 4 are compressed down onto row 5 and the hover on row 7 is placed with the following action on row 8.

To perform analysis to understand whether hovers are correlated to student performance and errors, the number of hovers in a list and the error of the following action are correlated and tallied. Due to the desire to understand whether or not the number of rules hovered over has an impact, and not the impact of the number of hover instances, the hover lists and the corresponding count are then updated to only include unique rule instances. Due to only including number of unique rules hovered over, the number of hovers value only ranges from 0 to 19 as there are only 19 rules.

Table 1: Reference sequences are constructed from each reference made before performing an action.

User	Ref	Rule
a1	N	HS
a1	N	CD
a1	Y	DS-
a1	Y	MP-
a1	N	MT
b1	N	DS
b1	Y	CD-
b1	N	HS

Ref-Seq	# refs	User	Rule
[]	0	a1	HS
[]	0	a1	CD
[DS, MP]	2	a1	MT
[]	0	b1	DS
[CD]	1	b1	HS

A zero in the error column indicates a valid action while the code 1, 3 and 6 indicate rule-application errors. As shown in the following example, the error values for all the occurrences for each number of hovers is tallied according to whether it indicated an error or not. For example, for the number of hovers of value 1, there are three occurrences; one occurrence has an error code of 0 while two have error codes of 1 and 3. The two with the error codes of 1 and 3 are tallied under the # of errors column and the 0 is tallied in the number of non-errors (see table 2.)

Table 2: Preparation of sequence size vs. errors

Ref lists	#Refs	Err
[]	0	0
[DS, MP]	2	1
[]	0	1
[CD]	1	3
[MT,HS,CD]	3	0
[ASSOC]	1	0
[MT,MT,DS]	3	0
[ABS]	1	1

#Refs	Err	Other
0	1	1
1	2	1
2	1	0
3	0	2

3. RESULTS

Deep Thought was assigned as a mandatory assignment in a philosophy deductive logic course. Six ordered levels of problems were assigned for full completion of the tutor, with each level comprising of 2–3 problems related to specific logical rules or proof problem-solving concepts as dictated by the course curriculum. Completion of an entire level of problems is required for assignment credit. Deep Thought is run as an on-line web applet, with students allowed to work unobserved through the problem sets at their own pace throughout the semester. There are a total of 47 students who were logged as using the system for the class; students who have no log data as a result of an early course drop, or students who did not attempt the assignment are removed from the data set. Three students were removed from the data, as these students altered default system preferences in order to have the hover-hints show up after 0 seconds of hovering, which caused problems with data collection; the normal setting is 2 seconds. This results in a total of 44 students used in this data set.

To explore the connection between the use of references and rule application errors we calculated the error rate for differ-

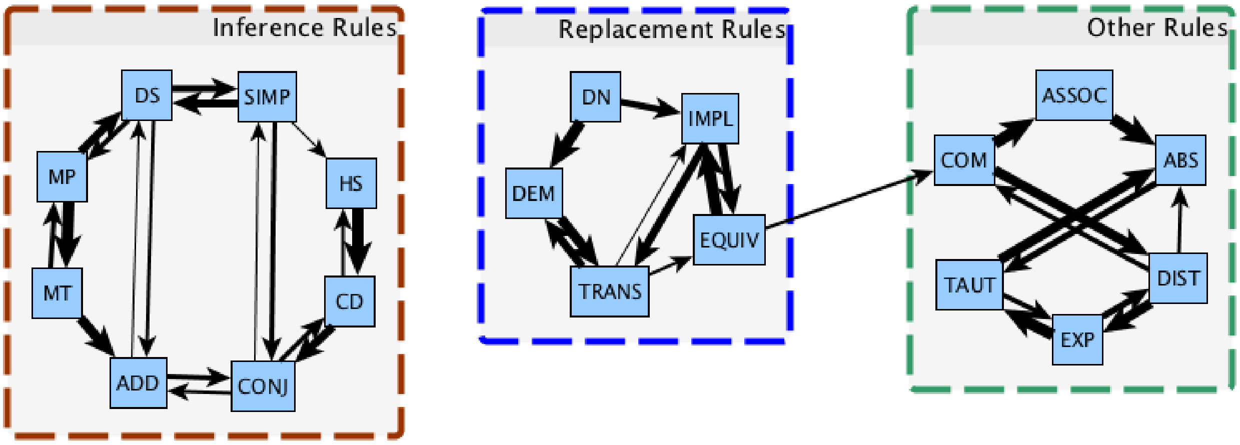


Figure 3: Graph representation of the bigrams of rule reference sequences. The edge thickness is weighted by the strength of the connection, with edges only drawn if the bigram is greater than 20%. Note how the references are generally contained within rule categories.

ing numbers of reference use. We tallied the errors and other interactions, the error rate is calculated for each number of references as follows,

$$ErrorRate = \frac{NumOfRefs}{NumOfRefs + OtherInteractions}. \quad (1)$$

The data is then binned by the number of rules within a string of consecutive hovers (0, 1, 2, 3–4, 5–19). The 0 bin is chosen to capture where references aren't used at all. The 1 bin is chosen to catch the behavior of having an action in mind, using the hover-reference as verification, and the 2 bin is chosen to catch the behavior of confusion between two rules. The 3–4 bin is chosen to catch the student with slightly more confusion, traversing rules based on spacial proximity. The >5 bin is chosen to capture referencing behavior that extended over the previous buckets. The results are presented in Table 3.

Table 3: Error rate after a number of references.

Hovers	Errors	Interactions	Error Rate
0	1201	22907	5.24%
1	104	889	11.70%
2	30	181	16.57%
3–4	14	132	10.61%
5–19	32	173	18.50%

To explore evidence of systematic reference behaviors, we analyze reference sequences as bigrams [2]. The sequences were used to generate counts of pair of rules referenced in secession which were then used as an adjacency matrix where the percentages represent edge weights (Figure 3). Edges are only drawn for bigrams greater than 20%.

In order to explore differences in how the references are used as students progress through the tutor, we generated the bigrams for all rules separated along conceptual shifts in the tutor (every two levels). We present all rules with a bigram greater than 20% during levels 1–2 in the first column of Table 4, the next two columns represent the change in bigram association from one level to the next (+ indicates that the rule is now past the 20% threshold, while - indicates that it is no longer above the threshold.)

Table 4: Rules and their associations over the course of the tutor.

Rule	1–2	3–4	5–6
ADD	CONJ, DS, MT	-DS, -MT	
CD	CONJ, HS		
CONJ	HS, SIMP	+ADD, +CD, -HS, -SIMP	-ADD
DS	ADD, MP, SIMP	-ADD	-MP
HS	CD, SIMP		-SIMP
MP	DS, MT		
MT	ADD, MP	+DS	
SIMP	CONJ, DS	+HS	
DEM	DN, TRANS	-DN	
DN	DEM, HS,	-HS, +IMPL	
EQUIV	COM, IMPL,		
IMPL	EQUIV, TRANS		
TRANS	DEM, EQUIV, IMPL		-IMPL

To analyze the use of references through the duration of the tutor, the interactions are broken up by tutor level and split into whether they were actions taken after the use of references or were actions without prior referencing. Table 5 shows the results of this analysis across the six assigned levels.

Table 5: Use of references before actions increases as student progress through the tutor.

	1	2	3	4	5	6
Actions	8256	2771	5110	5098	4424	2493
w/o ref	7862	2655	4779	4756	4082	2260
w/ ref	394	116	331	342	342	233
Percent	4.77%	4.19%	6.48%	6.71%	7.73%	9.35%

4. DISCUSSION

From Table 3 we find that usage of references before a tutor action corresponds with a larger error rate on that action. While this indicates that references alone do not seem to help students that reference perform better than those that don't reference, it does indicate that students that reference are more likely to make an error so the start of a reference sequence could be an effective place for an intervention. The number of errors after a single reference was lower than that after 2 or >5 references which could indicate that those that are using references for verification might be less likely to make an error than those that want to reference multiple rules. Investigation into the types of errors students made after two references revealed that 17 out of 30 errors contained only three of the 19 rules (DS, MP, SIMP,) this could indicate an area of confusion. The 3–4 bin has an unexpectedly lower error rate which could be because the students are thoroughly checking between the rules where their confusion lies instead of hastily taking a decision earlier. The bigram analysis supports this extrapolation because it shows that for any particular rule there are only a few strong connections, so students that reference 3–4 rules might be hovering over all the closely associated rules.

Figure 3 represents the bigrams of reference sequences with significant edge weights where the chance that any student would reference the second rule immediately after referencing the first is over 20%. The resulting graph and its clusters directly correspond to the three rule categories available in the tutor; this indicates that students tend to systematically reference rules within these groups (rather than randomly ask for references.) There are two reasons as to why they are hovering on rules in the shown order. One could be that they realize that the rules are related and are referencing them to learn the differences. Another possible reason could be that they are referencing rules in geographic proximity and the rules happen to be related since the tutor was designed to have related rules in one geographic area. The second explanation would indicate that the placement of references in a tutor is important to their utilization.

Table 4 provides evidence that students change which rules they frequently seek references for together over the course of the tutor. For example in Problems 1–2, after getting a reference on ADD the student is likely to seek reference on CONJ, DS, and MT; however in later levels the student is likely to only seek reference on CONJ. This shows some degree of learning, as the DS and MT rules are not very related to the ADD rule. The additions of rule associations can also be an indication of learning as the students are making additional connections as they progress. The rule associations shown are a promising location for new interventions where if the students are found hovering between unrelated rules, they can be shown a more effective hint that allows them to learn more about the references to clear the confusion.

We also found that as students progress through the tutor, and the problems increase in difficulty, they tend to use the references more often (Table 5). The students in the first two levels only consult the references between 4–5% of the time before choosing an action. This compares to students in the later levels using the references before 8–9% of their actions. The increase in levels 5 and 6 could be because the students must use a larger number of axioms to solve these problems, but the increase in levels 3 and 4 indicates that as the levels get more difficult, students feel the need to reference more before taking an action. The decrease for level 2 can be explained by the fact that the increase in difficulty between 1 and 2 is less steep than the increase between the other levels. From these observations, it can be extrapolated that an increase in the percentage of actions taken after referencing indicates that students feel a sense of higher difficulty which can also be useful information in making effective interventions.

5. CONCLUSIONS AND FUTURE WORK

We have found that usage of references before a tutor action corresponded with a larger error rate on that action, and as students progress through the tutor, they tend to use references more often with increased problem difficulty. Using this information we can aid students working through the tutor by using reference usage as an indicator for possible didactic intervention; offering feedback when the system determines a student having difficulty in tutor by their behavior with references.

We have also observed which axioms students tend to seek references for at the same time, revealing some changes in rule association as students become more familiar with the tutor. Using this information coupled with existing knowledge of problem parameters, we can determine which concepts students demonstrate difficulty understanding in the context of a particular problem, and provide feedback for those students accordingly.

6. REFERENCES

[1] M. Alonso, D. Py, and T. Lemeunier. A learning environment for object-oriented modeling, supporting metacognitive regulations. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, pages 69–73. IEEE, 2008.

[2] M. J. Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 184–191. Association for Computational Linguistics, 1996.

[3] M. J. Croy. Graphic interface design and deductive proof construction. *J. Comput. Math. Sci. Teach.*, 18:371–385, December 1999.

[4] S. B. Shneiderman and C. Plaisant. Designing the user interface 4 th edition. *ed: Pearson Addison Wesley, USA*, 2005.

[5] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education*, 22(1):3–17, 2013.

[6] S. White, L. Lister, and S. Feiner. Visual hints for tangible gestures in augmented reality. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–4. IEEE Computer Society, 2007.

Exploring real-time student models based on natural-language tutoring sessions

A look at the relative importance of predictors

Benjamin D. Nye[*]
The University of Memphis
Memphis, TN 38152
bdnye@memphis.edu

Mustafa Hajeer
The University of Memphis
Memphis, TN 38152
mhhajeer@memphis.edu

Carol M. Forsyth
The University of Memphis
Memphis, TN 38152
cmfrsyth@memphis.edu

Borhan Samei
The University of Memphis
Memphis, TN 38152
bsamei@memphis.edu

Keith Millis
Northern Illinois University
DeKalb, IL 60115
kmillis@niu.edu

Xiangen Hu
The University of Memphis
Memphis, TN 38152
xhu@memphis.edu

ABSTRACT

Natural language tutoring systems generate significant data during their tutoring sessions, which is often not used to inform real-time, persistent student models. The current research explores the feasibility of mapping concept-focused tutoring sessions to knowledge components, by breaking sessions down into features that are integrated into a session score. Three classes of tutoring conversation features were studied: semantic match of student contributions to domain content, tutor support (e.g., hints and prompts), and student verbosity (i.e., word counts). Analysis of the relative importance of these features and the ability of these features to predict later task performance on similar topics was conducted. Reinforcing prior work, semantic match scores were a key predictor for later test performance. Tutor help features (hints, prompts) were also useful secondary predictors. Unlike some related work, verbosity was a key predictor even after accounting for the semantic match.

Keywords

Student Modeling, Natural Language Processing, Educational Data Mining, Trialogs

1. INTRODUCTION

Adaptive, natural-language intelligent tutoring systems (ITS) have been a research goal since the early days of AI. The current research focuses on natural language conversation features from tutoring sessions instead of more explicit methods such as multiple choice questions. In this work, we associate each natural language tutoring conversation with a knowledge component, then score the session. This approach can be used in a pure natural language tutoring system to build a persistent, component-based student model that resembles those more commonly found in problem-based tutors. Effectively, tutoring conversations can provide a quasi "stealth assessment" [12] where the intelligent tutor assesses performance without breaking out from the tutoring session.

This study looks at the relative importance of tutoring session features and models a reasonable set of regression weights that could be used to inform an efficient, useful, and interpretable real-time student model. Focus was placed on preparing this model to transfer across domains. This is because the model is ultimately intended for a new tutoring system focusing on natural language tutoring for Algebra I, but it is being trained on data for a system focusing on research methodology. Additionally, a long-term goal of this model is to use it inside a generic persistent student model for AutoTutor-style tutoring [4], which is one major framework for conversational tutoring.

2. BACKGROUND AND RELATED WORK

Forbes-Riley and Litman [2] compared multivariate regression models to determine the relative usefulness of hundreds of features from sessions of the ITSPOKE system, a natural language ITS for physics. A variety of features were evaluated, which were categorized as either shallow features (e.g., student verbosity), semantic features (e.g., concept keywords/# of words), pragmatic features specific to ITSPOKE (e.g., number of goals to retry), discourse structure specific to ITSPOKE (e.g., depth), or local context for dialog acts (e.g., bigram speech act pairs). On holdout test data sets, models with semantic features consistently outperformed any model without these features, with R^2 values ranging from 0.338 to 0.524 depending on the data sets used. So then, semantic features appear to be the most pivotal. Pragmatic features, discourse structure, and local context all showed some evidence of usefulness for training,

[*] Corresponding Author

and on some test conditions, but none had the consistency of semantic features. Shallow features were not effective and were dropped when fitting regressions using all features. Follow-up research based on corpora from both ITSPOKE and BEETLE II, an ITS for electronic circuit analysis, also found meaningful correlations with posttest scores based on the number of semantically-relevant patterns (e.g., stemmed keywords) expressed by the student [9].

While the work with ITSPOKE indicated a potential for overfitting when basing models on tutoring behavior (e.g., ITSPOKE pragmatics) [2], later research on problem-based ITS indicates that pragmatic features may still be valuable. A study on continuous Bayesian knowledge tracing found that calculating partial credit for each problem based on the number of hints and bottom-out answers improved estimates of student performance, even though penalty weights for hints and other support were ad-hoc [13]. This research noted that greater numbers of hints correlated negatively with later performance. Since high-performing students are less likely to need hints, this is somewhat intuitive.

Overall, this review of the literature found that semantic features are essential predictive features and support (e.g., hints) may also be valuable. Verbosity during a session (e.g., average words per statement) was also considered as possibly useful, based on earlier analysis of AutoTutor data. Prior to the analysis, it was anticipated that higher semantic match scores would correlate positively with later performance, while support would correlate negatively with performance. Higher verbosity was expected to be associated with better performance, but was not necessarily expected to add value beyond the semantic match.

3. DATA SAMPLE AND FEATURES

This study analyzed a corpus of data collected from the Operation ARA (Acquiring Research Acumen), which is Pearson Education's commercial version of the desktop tutor Operation ARIES (Acquiring Research Investigative and Evaluative Skills) [10]. Both ARA and ARIES use natural language conversations based on the AutoTutor conversational framework. Operation ARA tutors research methodology using multiple methods, including three-way conversations between a human student and two or more artificial agents (e.g., trialogs). Operation ARA has three phrases (Training, Proving Ground, and Active Duty) and 11 chapters. These phases were preceded by a pretest (2 per chapter, 22 items total) and followed by a posttest (also 2 per chapter, 22 items total). Each chapter focuses on a particular concept related to research methodology (e.g., correlation vs. causation). Only sessions from the Training phase were used as inputs, since these dialogs are most representative of generative natural language tutoring (i.e., where the student attempts to explain concepts).

A set of 192 students across 11 classrooms and 9 instructors was analyzed in the current study. These subjects were from a pool of 462 students from 12 class sections in an undergraduate Psychology course at Northern Illinois University. Unfortunately, many students were excluded due to lack of required consent forms (190), missing pre/post tests (42), or unreasonably fast response times on pre/post tests (38). The latter group answered test questions in under 5 seconds, corresponding to a reading speed of over 10 words/second, far faster than is reasonable to read and methodically select an answer.

Pretest results were considered as a predictive feature, to determine the relative effectiveness of the tutoring conversation features against a traditional assessment. The final posttest results were treated as performance outcomes. Two chapters were excluded from the reported analysis because post-test results correlated poorly with pre-test results, raising some uncertainty over item equivalence. This may just be due to by-topic differences, as observed in an earlier data set collected with ARIES [3]. A follow-up analysis including these chapters found no overall change to model fit or conclusions: one topic raised fit, the other reduced it slightly.

Four features were extracted from each session: the average semantic match score during a session (i.e., average quality of student responses), the verbosity, and two "help" features (the number of hints and of prompts). For human statements answering a tutor or computer student question, the semantic match score and verbosity (number of words) were extracted. AutoTutor calculates and logs these semantic match scores as the session unfolds, using Latent Semantic Analysis [7, 4]. The average semantic match (S) and average verbosity (V) for student contributions was calculated. Statements were only included in this average if an open-ended response would be expected. As a substitute for average verbosity, a logarithm for verbosity was also calculated ($ln(1+V)$). This transform captures the inherent non-linearity of verbosity: a single word is qualitatively different than a blank response, but a single word onto a 100-word statement is seldom important.

Computer agents' statements were classified by their type, such as a hint or prompt. For each session, multiple hints (H) and multiple prompts (P) may exist. While the semantic match and verbosity of student statements partially influence the ITS to produce these speech acts, the specific rule sets and depth of content for tutoring also influences these values. Effectively, these capture the interaction of student input with the author's expert model for when feedback was needed.

Finally, while not a dialog feature, a student model needs to establish the relative importance of more recent dialog sessions as compared to earlier sessions. Since tutoring increases the student's understanding, knowledge levels are inherently non-stationary and have a certain degree of (hopefully positive) drift. The magnitude of this drift will determine the optimal update rate (λ_u) for weighting more recent sessions. For this study, an exponential moving average was applied (e.g., $\overline{X_t} = \lambda * x_t + (1 - \lambda) * \overline{X_{t-1}}$, where $\overline{X_1} = x_1$) [1]. In a simulated exploration, update rates between 0.4 to 0.8 had higher average model fits, with $\lambda_u = 0.5$ being representative. While this update rate is not claimed to be optimal, it was a reasonable starting point. This exponential moving average smoothed and summed each feature into a single feature score for a given concept (i.e., chapter).

4. DATA MINING AND MODELING

The analysis presented here had main goals: 1) Determine which tutoring discourse features contribute unique predictive value and 2) Determine the relative importance of these factors for predicting later test performance. Across the 192 students and 9 chapters analyzed, 1067 student-chapter combinations had at least one tutoring session during the Training phase and were used as the sample. The majority of pairs had only one ((656) or two (333) sessions, with a handful having three (75) or four (3). To note, since lower-performing students received more tutoring sessions, this data may over-represent lower-performing students. With that said, the number of tutoring sessions was not correlated with posttest scores for each chapter.

First, correlations were calculated between the posttest and time-averaged predictors. Next, three regression models were fitted to predict posttest outcomes: pretest only, the full set of tutoring session features, and all features plus the pretest. These were performed using 10-fold cross validation and on the full data set using Weka [6]. Then, the LMG (Lindeman, Merenda, & Gold) method for relative importance of linear regressors [8] was applied, as implemented in the R *relimpo* package [5]. LMG calculates the average R^2 contributed by each factor (e.g., the variance explained) across all orderings and combinations of regressors. Relative importance regressions often produce regression weights that generalize to new data. A second set of relative importance Pratt regression coefficients was also calculated [11]. Pratt weights are standardized coefficients (i.e., Beta weights) multiplied by the correlation between the predictor and outcome (i.e., $\beta * r_{i,out}$).

5. RESULTS

The correlations between factors followed the expected patterns. Table 5 shows Pearson correlations between the posttest results and predictors. The pretest (Pre), postest (Post), semantic match (S), and verbosity ($ln(1+V)$) all have highly-significant, positive correlations with each other ranging from weak to moderate. Hints (H) and prompts (P) correlate positively with each other, but negatively with the other variables. The logarithm of verbosity had much higher correlations with other variables than raw verbosity. For example, raw verbosity correlated 0.05 (p=0.08) with the posttest, compared to 0.17 (p<0.001) for the logarithmic transform.

Table 1: Posttest and Predictor Correlations

	Post	Pre	S	H	P	ln(1+V)
Post		0.14^c	0.14^c	-0.04	-0.08^a	0.17^c
Pre	0.14^c		0.06^a	-0.04	-0.06^a	0.08^b
S	0.14^c	0.06^a		-0.62^c	-0.69^c	0.55^c
H	-0.04	-0.04	-0.62^c		0.58^c	-0.31^c
P	-0.08^a	-0.06^a	-0.69^c	0.58^c		-0.43^c
ln(1+V)	0.17^c	0.08^b	0.55^c	-0.31^c	-0.43^c	-

^{a}p<0.05; ^{b}p<0.01; ^{c}p<0.001

5.1 Linear Weights

All features (pretest, semantic score, hints, prompts, and verbosity) improved the linear model during 10-fold cross validation and on the full data set. Three key models are shown in Table 2: pretest only, all tutoring features, and the combined set of predictors (pretest and tutoring). The low variance explained by the pretest demonstrates the noise in the data, since pretest values often account for the majority of the explained variance [2]. In this data set, dialog features are significantly more predictive than the pretest alone.

Table 2: Variance Explained by Linear Regressions

Predictor Set	R^2 (Training)	R^2 (Cross Val.)
Pretest Only	0.020	0.017
Tutoring Only	0.037	0.028
Combined	0.054	0.043

While the overall variance explained is modest, very limited data was available for each combination of student and chapter. Most pairs contain only a single session and the average session had 2.4 student contributions. To look at the added value for additional sessions, the data was split into two subsets: single-session ($N_S = 1$) and multiple-sessions on a chapter ($N_S > 1$). Table 3 shows the model fits for this split. Discourse features were more predictive when multiple sessions were available. The combined model with two or more tutoring sessions outperforms any other model, and accounts for 8% of training variance and 5.7% for cross-validation. 81% of the $N_S > 1$ subset have two sessions, so even adding one additional session captures 1.4% to 2.6% of the remaining component variance.

Table 3: Impact of the Number of Sessions on Variance Explained

Predictor Set	R^2 (Training)	R^2 (Cross Val.)
Pretest Only ($N_S = 1$)	0.027	0.022
Pretest Only ($N_S > 1$)	0.012	0.005
Tutoring Only($N_S = 1$)	0.028	0.018
Tutoring Only($N_S > 1$)	0.072	0.053
Combined ($N_S = 1$)	0.052	0.038
Combined ($N_S > 1$)	0.080	0.057

The remainder of the analysis focuses on the tutoring features only. While pretests have predictive value, they are content-specific and are unlikely to be transferable to a new domain. Table 4 shows three sets of relative importance weights for each feature: LMG (contribution to R^2), Pratt (standardized, meaningfully-signed coefficients), and a set of interpretable unstandardized weights generated by transforming the Pratt weights. LMG and Pratt weights were similar in relative magnitude, with the exception that the Pratt weights are signed. Verbosity and semantic match scores dominate in both cases, with the influence of hints and prompts almost an order of magnitude lower. With that said, hints and prompts are still significant predictors and improve the model fit.

Table 4: Relative Importance Weights

Predictor	LMG	Pratt	Interpretable
Semantic	0.0124	0.0183	1.00
Hints	0.0019	-0.0024	-0.078
Prompts	0.0022	-0.0026	-0.013
ln(1+V)	0.0208	0.0242	0.28
R^2	0.037	0.037	0.031

Interpretable weights were generated in a two-step process. First, each Pratt weight was divided by the sample standard deviation for that variable. Second, all of these weights were divided by the semantic match score weight. These resulting weights retain the majority of the predictive value on the training data, so long as predictions are clipped to fit in [0,1]. Rescaling the weights until the intercept is zero tended to offer a higher fit than other scalings. This occurred when the semantic match coefficient was close to 1, as displayed in the above weights (1.045 for the 9-chapter set and almost exactly 1 when the an additional chapter was considered). As such, it appears that the semantic match score acts like a de-facto intercept value. This model was used to predict the sum of student posttest scores across their included chapters (i.e., any chapter with a tutoring session). The Pearson correlation between the sum of each student's posttest scores and the sum of predicted knowledge levels was fairly strong (R^2=0.388, p<0.001, N=192).

6. CONCLUSIONS AND FUTURE WORK

The logarithm of verbosity and the semantic match score were the primary predictors, performing better than even the pretest items. The high importance of verbosity was somewhat surprising, given prior work which found little value for surface features [2]. The difference may have been caused by this study focusing on the average verbosity on a particular concept, rather than overall word counts. Additionally, the logarithmic transform improved verbosity from a fairly weak correlate to a powerful one. Support such as hints and prompts was also predictive, and negatively related to later performance on the posttest. This weaker importance is probably caused by causation from poor semantic match (poor answers make hints more likely) and learning due to hints (learning from hints offsets worse knowledge).

The model explained significant variance in the overall posttest score (R^2=0.39), but modest variance for each component. Given that most components had only one short tutoring session (about 2.4 student contributions) to predict a pair of posttest multiple choice questions, this is fairly promising. Using only a handful of student utterances, this model outperformed balanced pretest items for predicting posttest component performance. Since even a single additional session significantly increased the variance explained, more sessions per concept should improve predictions. Additionally, the Proving Ground and Active Duty phases add noise between the Training phase and posttest.

With that said, these results are drawn from tutoring dialogs on a single domain with a fairly small number of topics. Follow-up research will test the model on a new domain (Algebra I), with larger numbers of tutoring sessions per concept. This evaluation will occur during the next year, and should provide useful information about the transferability of this tutoring session scoring model to a new domain. Future studies will focus on the effectiveness and limitations of a student model for classifying student performance, once pilot and evaluation data have been collected.

7. ACKNOWLEDGMENTS

This research was supported by the Office of Naval Research STEM ITS Grand Challenge, the Institute for Education Sciences (R305B070349), and the National Science Foundation (HCC 0834847). The authors alone are responsible for statements in this paper.

8. REFERENCES

[1] R. G. Brown. *Smoothing, forecasting and prediction of discrete time series.* Dover Publications., Mineola, New York, 2004.

[2] K. Forbes-Riley, D. Litman, A. Purandare, M. Rotaru, and J. Tetreault. Comparing linguistic features for modeling learning in computer tutoring. pages 270–277, June 2007.

[3] C. Forsyth, P. J. Pavlik, A. C. Graesser, Z. Cai, M.-l. Germany, K. Millis, R. P. Dolan, H. Butler, and D. Halpern. Learning gains for core concepts in a serious game on scientific reasoning. In *Educational Data Mining (EDM) 2012*, pages 192–195. International Educational Data Mining Society., June 2012.

[4] A. Graesser, P. Chipman, B. Haynes, and A. Olney. AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618, Nov. 2005.

[5] U. Grömping. Relative importance for linear regression in r: the package relaimpo. *Journal of Statistical Software*, 17(1):1–27, 2006.

[6] M. Hall, E. Frank, and G. Holmes. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[7] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284, 1998.

[8] R. Lindeman, P. Merenda, and R. Gold. *Introduction to bivariate and multivariate analysis.* Scott Foresman, Glenview, IL, 1980.

[9] D. Litman, J. Moore, M. O. Dzikovska, and E. Farrow. Using natural language processing to analyze tutorial dialogue corpora across domains modalities. In *AIED 2009*, pages 149–156, Amsterdam, The Netherlands, 2009. IOS Press.

[10] K. Millis, C. Forsyth, H. A. Butler, P. Wallace, A. C. Graesser, and D. F. Halpern. Operation ARIES! a serious game for teaching scientific inquiry. In M. Ma, A. Oikonomou, and J. Lakhmi, editors, *Serious Games and Edutainment Applications*, pages 169–195. Springer, London, UK, 2011.

[11] J. Pratt. Dividing the indivisible: Using simple symmetry to partition variance explained. In *Proceedings of the second international conference in statistics.*, pages 245–260, Tampere, Finland, 1987. University of Tampere.

[12] V. Shute. Stealth assessment in computer-based games to support learning. In S. Tobias and J. D. Fletcher, editors, *Computer games and instruction*, pages 503–524. 2011.

[13] Y. Wang and N. Heffernan. Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education (AIED) 2013*, pages 181–188, Berlin, Germany, 2013. Springer.

Forum Thread Recommendation for Massive Open Online Courses

Diyi Yang, Mario Piergallini, Iris Howley, Carolyn Rose
Carnegie Mellon University
Language Technologies Institute
5000 Forbes, Pittsburgh, PA
{diyiy, mpiergal,ihowley, cprose}@cs.cmu.edu

ABSTRACT

Recently, Massive Open Online Courses (MOOCs) have garnered a high level of interest in the media. With larger and larger numbers of students participating in each course, finding useful and informative threads in increasingly crowded course discussion forums becomes a challenging issue for students. In this work, we address this thread overload problem by taking advantage of an adaptive feature-based matrix factorization framework to make thread recommendations. A key component of our approach is a feature space design that effectively characterizes student behaviors in the forum in order to match threads and users. This effort includes content level modeling, social peer connections, and other forum activities. The results from our experiment conducted on one MOOC course show promise that our thread recommendation method has potential to direct students to threads they might be interested in.

Keywords

Thread Recommendation, Massive Open Online Courses, Matrix Factorization

1. INTRODUCTION

Massive Open Online Courses (MOOCs) have rapidly moved into a place of prominence in the media in recent years. MOOC platforms, such as Coursera[1] and EdX[2], are faced with course registration and participation in the hundreds of thousands, and potentially have even larger student populations. A novel component of online learning courses is the use of interactive discussion forums where instructors and students can ask questions, discuss ideas, provide help to or even socialize with other students. As class sizes grow, the number of threads per course forum increases rapidly. Consequently, it becomes more difficult for students to find what they are looking for or truly interested in.

One solution to this problem would be to give each student a short list of threads that we believe would interest them. If a forum system could automatically detect user interests to generate personalized thread recommendations, it would make it much faster and more convenient for students to find the threads they want to participate in. Additionally, this would decrease the amount of time that new questions go unanswered by directing appropriate users there. A student's potential interest in a thread is largely determined by the match between the student's preferences and the content focus of the thread.

In this work, we propose a model for thread recommendation in MOOC discussion forums that addresses issues caused by massive thread volumes and help students to answer their fellow students' questions more quickly. To do this, first we use content level indicators of threads students have participated in to capture their preferences over threads, which helps to do latent matching between threads and students' interests; then we design an adaptive time window matrix factorization model to take students' behavior in the current time window and predict their behavior in the following time window; finally, we conduct experiments on one Coursera[3] course, and demonstrate that our model gives significantly improved performance over several baselines. Quantitative analysis, including exploration of differing window sizes, is provided to validate our approach.

2. RELATED WORK

Considerable prior research on MOOCs has focused on concerns related to activities in MOOCs apart from discussion, such as watching videos, peer grading [11], and dropout [6]. Previous work in a variety of other contexts [2] explores student activities in discussion forums. In that work, the underlying hypothesis is that participation in learning-related activities such as discussing and sharing, could have a positive influence on knowledge gain [3]. Neo-Piagetian theory on collaborative learning suggests that discussion provides opportunities to experience cognitive conflict, which potentially produces learning [10]. In a classroom setting, Wood et al. [14] explored how learner and tutor interaction influence learning outcomes, which further argues for a relationship between participation in discussion and students' learning. Thus, discussion participation is an important activity to support as it is another source of knowledge and learning within a MOOC.

[1] https://www.coursera.org/
[2] https://www.edx.org/
[3] https://www.coursera.org/

In terms of the context of MOOCs, where the interaction with or guidance from instructors are limited, and dropout in these massively enrolled environments is very high, it becomes more necessary to improve the participation and engagement of students in the course [9]. One direct indication of students' commitment in MOOCs is their activities in the discussion forum. Those discussion threads focus on questions and confusion about lectures, including clarification requests about assignments or exams. Other times they are off-topic or just socializing [7]. Finding an earlier thread that answers one's questions or applies to one's interests among such a large set of threads is challenging. This becomes even worse for students who created threads seeking help since their potential helpers may simply not find them [13]. Thus, thread recommendation (i.e., the production of a short list of potentially interesting threads) has great potential for increasing the value and approachability of MOOC discussion forums.

Existing work in question recommendation mainly focuses on online discussion forums such as Yahoo! Answers [12]. For instance, the work by Hu et al. [8] introduced a user modeling method that estimates the interests and professional areas of each user in order to generate a suitable user set to answer a given question. However, MOOC discussion forums frequently lack the rich information that generic online forums have, such as user reputation (which would be important here because threads of a highly reputed person is more likely to attract others' attentions) [1]. Besides, students in a MOOC forum differ from common users of more typical types of web forums, since their length of participation is typically only around eight weeks in the forum and most students choose to drop out as time proceeds [15]. Thus, research is needed to determine how best to take advantage of expertise in MOOC forums so that the thread recommendation problem is solvable. To the best knowledge of the authors, this is the first work on thread recommendation in MOOC forums.

3. THREAD RECOMMENDATION

We introduce the adaptive feature-based matrix factorization (MF) framework that we use to recommend threads to students in this section. To begin with, we describe the adaptive MF framework, then we explain how we incorporate content level modeling, social peer connection and other contextual information into our framework.

3.1 Adaptive Matrix Factorization

Classical matrix factorization (MF) [5] could address the thread recommendation problem efficiently. MF constructs a reduced representation that mediates some feature based representation of users and threads. That representation can then be used to match users with appropriate threads. However, different from traditional product recommendation, for MOOC thread recommendation one important property is that each time a student logs into the forum, they are more likely to participate in threads that were posted more recently. New threads are more likely to be relevant to the current subject in the course while old threads may be irrelevant to them. Taking advantage of both the MOOC property and state-of-art matrix factorization framework, we propose an adaptive matrix factorization model. We illustrate how we design this adaptive model in two steps as follows.

In the first step, we give a detailed formulation of the feature based matrix factorization. Formally, suppose we have three feature sets G, M, and N called global features, student features and thread features, respectively. α, β, and γ are the extracted feature values. α is for global features, β is for student features, and γ stands for thread features. Then, for each record user, thread, and participation or not indicator, $< u, t, r_{u,t} >$, the predicted score $\hat{r}_{u,t}$ is defined as follows (p_u and q_t are latent vectors associated with users and threads):

$$\begin{aligned}\hat{r}_{u,t} = \; & \mu + (\textstyle\sum_{g\in G}\gamma_g b_g + \sum_{m\in M}\alpha_m b_m^u + \sum_{n\in N}\beta_n b_n^t) \\ & + \textstyle\sum_{m\in M}\alpha_m {p_m}^T \sum_{n\in N}\beta_n q_n\end{aligned} \tag{1}$$

The global features are used to incorporate information which is related to all students and threads, i.e. tendencies that hold for the entire forum. Meanwhile, student features and thread features can capture the information related only to specific students or threads. When the indicators of student and thread are the only student and thread features without any other global features, this feature-based matrix factorization model naturally degenerates to classical matrix factorization. This matrix factorization framework gives us the ability to incorporate as many features as desired.

In the second step, we elaborate how we adapt the basic framework into the adaptive model. Firstly, we define a time window of size Δ that moves along the course weeks. In order to recommend threads to students in week w, our feature-based matrix factorization will be trained only on the data between time $w - \Delta$ and $w - 1$. If $w \leq \Delta$, only the data between week 1 and week $w-1$ is utilized. Additionally, the candidates for recommendation are only active threads, which are threads that were posted or received at least one reply during the time window. Since the time window slides across the course period, the performance of our model can be evaluated by averaging the performances of each week.

3.2 Contextual Modeling

In this section, we present several contextual aspects that we incorporate into the adaptive feature-based matrix factorization framework.

Content Level Modeling: We assume that students' preferences over threads are approximately equivalent to their preferences for the contents of those threads. To exploit the content of the thread to do the latent matching between threads and the interests of students, we represent the content of the thread as a bag-of-words [16]. Thus, we can transform the problem into whether the student is interested in the words or topics in the thread, rather than the thread itself. Intuitively, a thread question t consists of a set of words $W(t)$ out of the entire word set Z. This content level modeling is formulated as follows:

$$\hat{r}_{u,t} = bias + {p_u}^T(q_t + \frac{\sum_{w\in W(t)}\phi_w}{|W(t)|}) \tag{2}$$

bias is some constant representing generalized possible biases and $|W(t)|$ is the number of word contained in thread t. ϕ_w captures the influence of word w on students.

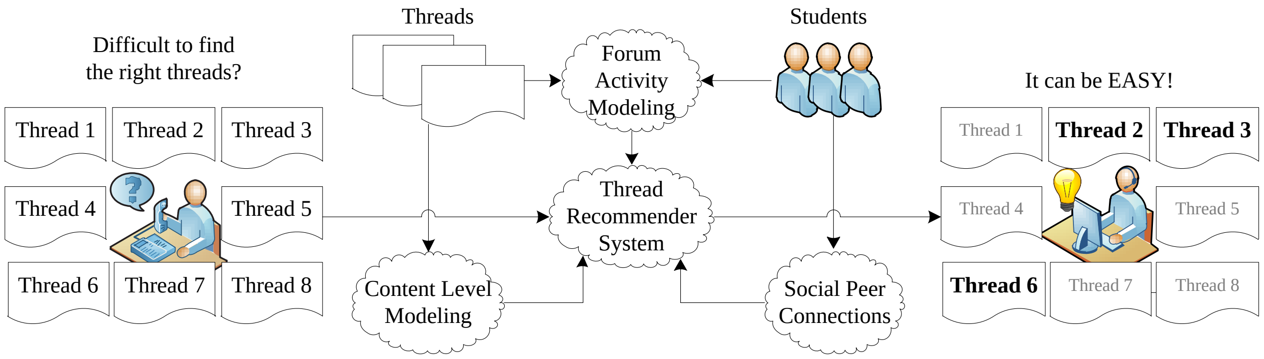


Figure 1: Why do we need a thread recommendation system in MOOCs? For students, it is hard to find which threads among thousands they want to participate. Our designed Thread Recommender System, can fully leverage these features and provide accurate recommendations.

Social Peer Connections: Students who interact frequently in the forum share similar engagement conditions and similar learning interests toward the course. Here, we use the connections between a student and their close peers $S(u)$ to capture peers' influence on students. We define the close peers as the top students who have the most interaction occurrences with student u based on replies. This peer influence could be characterized as follows. (φ_v models the influence ability of the student v as a peer on other students.)

$$\tilde{r}_{u,t} = bias + (p_u + \frac{\sum_{v \in S(u)} \varphi_v}{\sqrt{|S(u)|}})^T q_t \qquad (3)$$

Forum Activity Modeling: For student related features, the number of different threads students participated in before the current week (**Thread Count**) shows their historical participation level, while the number of posts made in the previous week (**Previous Count**) is recorded to reflect their recent activity level. **Cohort** representing when this student registered for the course can be treated as a proxy for the level of motivation(i.e. early course registration indicates initiative motivation). The number of replies and comments **Reply Number** is counted as a thread feature. **Thread Length** representing the total number of words appearing in the thread is also computed.

4. EXPERIMENTS

In this section, we introduce the dataset, experiment setup and baselines. Then we discuss our experimental results along with the adaptive window size exploration.

4.1 Experiment Setup

We conducted our experiments on one Coursera course, 'Learn to Program: The Fundamentals', shortened to 'Python Course'. It has 3590 active students who have at least one post in the forum and 3079 threads across around eight weeks, based on which we performed the time window evaluation. For each thread, we have its replies and comments; threads' contents and students' registration time are also available. Mean Average Precision (MAP) [4] is our evaluation metric. Specifically, we use MAP@1, MAP@3, and MAP@5 to evaluate the performance. Our analysis is limited to only behaviors within the discussion forums.

To make our analysis clear and concise, we define some notation here. Content level modeling is denoted as C; we denote social peer connections as P; the student related features are S, and thread related features as T. Specially, we use *All* to denote the integration of all aspects of the features. We empirically set the size of the time window as 2 weeks. That is, when we predict the preferences of students over threads in week 2, only the data in week 1 is used to train the model; likewise, to make the prediction in week 5, the forum history in week 4 is used.

Baselines used in this work include **Popularity (PPL)** which conducts thread recommendations based on thread popularity, **Directly Content Match (DCM)** which recommends threads based on how similar the content of the thread is to the post history of the student, and **Classical Matrix Factorization (MF)** that maps students and threads into the same latent space without contextual information. Our proposed **Adaptive Matrix Factorization (AMF)** utilizes the rich contextual information via encoding different information into its feature space. We use the notation AMF-{All} to describe a model using all types of features. AMF-{C} means that only content features are used in the AMF model.

4.2 Recommendation Performance

In this section, we present the recommendation results from one MOOC. Based on the results of different models shown in Table 1, we could observe that the DCM is the worst among all models. The performance of the MF model is at least 0.02 higher than the PPL model regarding to MAP@1, MAP@3 and MAP@5. The series of AMF models, which contain each aspect of our designed four aspect features, has better performance over PPL and MF. This demonstrates that each type of feature makes an important contribution in capturing the latent matching between interest of students and the topics involved in threads. One notable point is that the Student Forum Activity Modeling feature set is better than any of the other single feature dimensions. Fully combining all types of features makes the best model, which indicates that the four types of features capture different aspects of modeling of the latent matching between students and threads.

Method	MAP@1	MAP@3	MAP@5
PPL	0.154	0.254	0.307
DCM	0.092	0.198	0.172
MF	0.171	0.280	0.332
AMF-{C}	0.177	0.282	0.340
AMF-{P}	0.178	0.286	0.340
AMF-{S}	0.183	0.290	0.341
AMF-{T}	0.174	0.280	0.334
AMF-{All}	**0.198**	**0.323**	**0.376**

Table 1: Average Results on Python Course

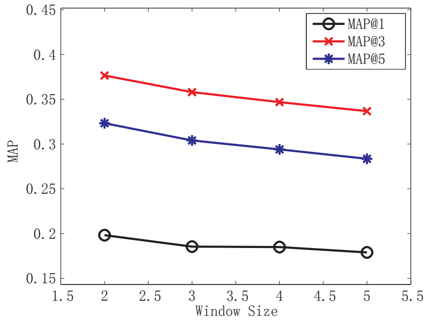


Figure 2: Window Size Exploration

4.3 Window Size Exploration

One important parameter in our adaptive MF model is the window size. We explained that we chose two weeks as the proper time window size. In this section, we describe how we tune this parameter and how the recommendation performance changes as window size increases. We only test how the performance of the best model AMF-{All} changes as window size changes. The results of the MAP changing curve is presented in Figure 2. We can observe that AMF-{All}'s performance is always decreasing as the window size increases. This makes sense because when students log into the forum system, they are more likely to pay attention to recent threads. In conclusion, students' activities in very recent weeks are more predictive of their participation in the later week. The smaller the window size, the better the thread recommendation performance.

5. CONCLUSION AND FUTURE WORK

In this work, we created a thread recommendation system for MOOC discussion forums in order to improve the learning experience of students. For this purpose, we proposed an adaptive matrix factorization framework to capture the affinity of students for threads; then we integrated content-level modeling, social peer connections, as well as measures of students' overall forum activities into that framework. Experiments conducted on the MOOC dataset show that our proposed model significantly outperforms several baselines. In the future, we plan to conduct some deployed studies in active MOOCs to validate our framework.

Acknowledgement

This research was supported in part by NSF grants OMA-0836012 and IIS-1320064.

6. REFERENCES

[1] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 850–858, New York, NY, USA, 2012. ACM.

[2] G. Balakrishnan. Predicting student retention in massive open online courses using hidden markov models. Master's thesis, EECS Department, University of California, Berkeley, May 2013.

[3] R. M. Carini, G. D. Kuh, and S. P. Klein. Student engagement and student learning: Testing the linkages*. *Research in Higher Education*, 47(1):1–32, 2006.

[4] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 661–670, New York, NY, USA, 2012. ACM.

[5] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu. Svdfeature: a toolkit for feature-based collaborative filtering. *The Journal of Machine Learning Research*, 13(1):3619–3622, 2012.

[6] D. Clow. Moocs and the funnel of participation. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 185–189. ACM, 2013.

[7] S. S. Glenda, D. Jennifer, W. Jonathan, and B. Lori. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Workshop on Data Driven Education, Advances in Neural Information Processing Systems 2013*, 2013.

[8] D. Hu, S. GU, S. Wang, L. Wenyin, and E. Chen. Question recommendation for user-interactive question answering systems. In *Proceedings of the 2Nd International Conference on Ubiquitous Information Management and Communication*, ICUIMC '08, pages 39–44, New York, NY, USA, 2008. ACM.

[9] R. Junco. The relationship between frequency of facebook use, participation in facebook activities, and student engagement. *Computers and Education*, 58(1):162–171, 2012.

[10] J. Piaget. The equilibrium of cognitive structures: the central problem of intellectual development. *Human development*, 1985.

[11] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. Tuned models of peer assessment in MOOCs. In *Proceedings of The 6th International Conference on Educational Data Mining (EDM 2013)*, 2013.

[12] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th international conference on World wide web*, pages 1229–1230. ACM, 2009.

[13] I. Roll, V. Aleven, B. M. McLaren, and K. R. Koedinger. Improving students help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, 21(2):267–280, 2011.

[14] H. Wood and D. Wood. Help seeking, learning and contingent tutoring. *Computers and Education*, 33(2):153–169, 1999.

[15] D. Yang, T. Sinha, D. Adamson, and C. P. Rose. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Workshop on Data Driven Education, Advances in Neural Information Processing Systems 2013*, 2013.

[16] Y. Zhang, R. Jin, and Z.-H. Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

Investigating Automated Student Modeling in a Java MOOC

Michael Yudelson
Carnegie Learning, Inc.
437 Grant St.
Pittsburgh, PA 15219, USA
myudelson@carnegielearning.com

Roya Hosseini
Intelligent Systems Program
University of Pittsburgh
210 South Bouquet Street
Pittsburgh, PA,
roh38@pitt.edu

Arto Vihavainen
Dep. of Computer Science
University of Helsinki
P.O. Box 68 FI-00014
avihavai@cs.helsinki.fi

Peter Brusilovsky
University of Pittsburgh,
135 North Bellefield Ave.,
Pittsburgh, PA 15260, USA
peterb@pitt.edu

ABSTRACT

With the advent of ubiquitous web, programming is no longer a sole prerogative of computer science schools. Scripting languages are taught to wider audiences and programming has become a flag post of any technology related program. As more and more students are exposed to coding, it is no longer a trade of the select few. As a result, students who would not opt for a coding class a decade ago are in a position of having to learn a rather difficult subject. The problem of assisting students in learning programming has been explored in several intelligent tutoring systems. The key component of such systems is a student model that keeps track of student progress. In turn, the foundation of a student model is a domain model – a vocabulary of skills (or concepts) that structures the representation of student knowledge. Building domain models for programming is known as a complicated task. In this paper we explore automated approaches for extracting domain models for learning programming languages and modeling student knowledge in the process of solving programming exercises. We evaluate the validity of this approach using large volume of student code submission data from a MOOC on introductory Java programming.

Keywords

Big Data, MOOC, Student Modeling, Automated Domain Model Construction.

1. INTRODUCTION

Today, information and computer technology is all around us. Programming is not an art accessible to the few and taught at select computer science schools anymore. Scripting and programming languages are taught to wider student audiences and programming courses have become a flag post of any technology related program. As more and more students are taking on programming, it becomes a universal skill, a necessity for every student studying increasingly computerized technology. As a result, the distribution of talent in programming classes shifts from the mathematically gifted to the overall population mean.

There have long existed a number of educational systems that have served the purpose of teaching students an abundance of programming languages and since then have greatly advanced the field of online learning. LISPTUTOR – a system teaching students a language of LISP – was the precursor of the modern intelligent tutoring systems [1] and SQL-tutor – a constraint-based system that instructed students who learned SQL [6], to name just a few.

A classical educational system always has a user model – an integral component responsible for keeping track of student progress. The core of a student model is a vocabulary of skills (concepts) that structure the representation of student knowledge. Conceptualizing a set of skills is a hard task in and of itself. However, programming is an inherently structured domain. The basis of a programming language is the grammar that imposes a structure on any code that compiles.

There were several attempts to exploit the inherent structure of the programming language with respect to student modeling tasks. For example, authors of [7] used a parsed concept map of C and Java to perform cross-adaptation of the content while [11] and [4] used the concept structure of parameterized questions for C and Java to provide within-domain adaptive navigation support.

Until now, to the best of our knowledge, there were no attempts to utilize an auto-parsed structure of the code as a substitute for a conceptualization of the knowledge model. The benefits of such automation with respect to programming are many. First of all, it is inherently transferrable to any programming or scripting language: one just has to have a parser for that language. Second, given the parsed concepts, student modeling can be done on the fly. Third, with recent popularity of massive open online courses, there are volumes of data potentially available to experiment.

The challenge of this approach is that, besides relative easiness of extraction, when programs start to get more complex so grows the volume of concepts parsed and the signal becomes noisier. Additionally, identifying programming constructs essential for passing a particular test is not trivial. And finally, high accuracy of such models can ensure help is given to a student while selecting the next problem, while a model's capacity to aid students during problem solving requires a different form of validation.

In this paper, we report on our investigation of automatically generated user models for the assignment-grading system deployed in a set of introductory programming classes. The data intensity of the code submission stream makes the task of knowledge modeling truly a "big data" problem. Results of our retrospective analysis demonstrate that the models created automatically can successfully support students during problem solving activity.

2. DATA

To explain our idea and a set of explored user modeling approaches, it is important to start with a description of data that we had at our disposal and how the data was processed for our studies. Our data came from three introductory programming courses organized by the University of Helsinki; two local courses held during Fall 2012 and Fall 2013, and a MOOC held during Spring 2013. Although the MOOC course lasted 12 weeks in total (see [9] for details of an instance offered in 2012), we included only the first six weeks in the analysis to be able to compare the results directly with the introductory programming courses.

The courses emphasized students' personal effort and constant practice. New topics were always accompanied by a set of programming exercises where the first tasks provided clear guidelines that outlined both the required program structure and required functionality, and latter ones were open-ended, giving students more freedom on the application design. During the six weeks, the students worked on over 100 exercises that were further split into a total of some 170 tasks. All exercises were done using an industry-strength IDE with a plugin that provided textual on-demand feedback that had been encoded into the tasks, records students' progress, and allowed students to submit their solutions for grading directly from within the IDE [10].

Table 1. Overall student population statistics.

Course	N. students (M/F)	Age: Avg./ Med./Max.	N. snapshots: all/median
Fall 2012: Introduction to Programming	185(121/64)	18/22/65	204460 / 1131
Fall 2013: Introduction to Programming	207(147/60)	18/22/57	263574 / 1126
Spring 2013: MOOC on Programming[1]	683(492/60)	13/23/75	842356 / 876

Due to this unique problem-solving approach and careful data archiving, each of the three courses produced a unique picture of student behavior. For each exercise and for each student, the system stored a relatively large sequence of code snapshots that were taken on save, compile and run -events, each representing a complete or incomplete attempt to solve the programming problem. Moreover, since each snapshot was tested on a set of tests designed for the corresponding exercise, information on tests that passed and failed was available. This data provides an excellent start for exploring various approaches to student modeling.

Student population statistics are given in Table 1. Each in-class version of the course had about one half of the MOOC's attendance. All three courses were mostly taken by male students (more so in the case of the MOOC). Age distribution was roughly the same. Around 40% of the students were CS major (in-class courses) and around half of the students had previous programming experience (MOOC). In terms of student activity (the number of snapshots) student medians were quite close for the two in-class course, and for the MOOC this number is lower due to the dropout rates.

3. APPROACH

We investigated an automated approach to creating concept models and student modeling for the domain of introductory Java programming. Our approach was based on two principal ideas: (1) modeling knowledge behind every program submission using the inherent structure of the programming language and (2) automatic testing of program correctness using a set of tests. In brief, we considered every program submitted or saved as a solution of a programming exercise as an application of a range of concepts that were present in the submitted code. Once this program passed one or more tests, we considered it as a *successful* application of these concepts in an absolute sense or relative to the earlier submission. In the specific case explored in the paper, concept extraction from the body of submitted program was done by our concept extraction tool "Java Parser" [3] while the correctness of the submitted student code was determined by the system infrastructure introduced above. Thus, the main body of the paper is focused not on the tools, but on using the large body of collected data to explore the plausibility of the approach - the correctness of the student model itself and its usefulness in assisting students while they work on the code.

3.1 Data Preprocessing

For our analysis we preprocessed the raw student submissions. First the code was compiled and run against the suite of tests recording which tests passed. Each snapshot was also analyzed using JavaParser [3]. The extracted concepts were recorded both as an exhaustive list of all concepts in the snapshot and as a difference from the previous snapshot accounting for additions and removals (initial snapshot copied in full). An additional data-thinning procedure removed all snapshots that had an empty list of concept changes to filter out insignificant changes to the code.

3.2 Hypotheses

First, it is possible to model student knowledge acquisition (models can detect learning). Second, only a subset of code constructs is important for solving a particular problem. Third, constructed models are useful beyond modeling student knowledge acquisition and can be used as a basis for creating a recommendation component to help students with the code.

4. MODELS

We chose a set of models that are widely used in the field of student modeling. We first set the modeling lower boundary with the *Null model* (the majority class model). The next model of our choice was the *Rasch model* (1PL IRT) [5]. Although the Rasch model does not capture learning by definition, it is frequently used in psychometrics and would set a baseline for us. The model is given in Eq. (1). Here, *Pr* denotes probability of student *i* to correctly solve problem *j*. *Inverse.logit* is the sigmoid function, θ_i is the student proficiency parameter, and β_j is the item complexity parameter. Since the result of compiling and running a problem is a binary mask of passed and failed tests, we treated the problem-test tuples as unique items. We broke each student transaction from the data into *n*, where *n* is the number of tests submission is checked against. Passed tests would yield a result of 1, failed a result of 0. Student and concept data were copied across the broken transactions accordingly. We fit Rasch model using mixed effect regression, treating both student and item complexity parameters as random factors.

$$\Pr_{ij} = \Pr(Y_{ij} = 1 \mid \theta, \beta) = inverse.\text{logit}(\theta_i + \beta_i) \qquad \text{Eq. (1)}$$

$$\Pr_{ij} = \Pr(Y_{ij} = 1 \mid \theta, \beta, \delta, \gamma) = inverse.\log it(\theta_i + \beta_i + \sum_k (\delta_{kj} + \gamma_{kj} t_{ikj})) \qquad \text{Eq. (2)}$$

To actually model student learning we would use a variant Additive Factors Model (AFM) [2]. In addition to the parameters in Rasch model, AFM (Eq. (2)) has skill complexity – δ_{kj} (intercept), and skill learning rate – γ_{kj} (slope). Although standard AFM does not have item complexity, we will have it in our AFM models to account for item variability. For each student submission we will count the number of prior attempts to use a particular coding construct – t_{ikj}.

[1] We only include data on the students that answered the survey

In AFM it is customary to fit concept intercepts and slopes across all items. We will treat concepts as within-item effects.

When the standard AFM model is used, for each item or problem step a set of relevant concepts is known. Often, a table relating concepts to items is called a Q-matrix. We do not have information on what programming constructs are relevant for the successful passing of the tests. We used three different rules to select concepts. Rule A selects all concepts that were parsed from the student code. Rule B uses the concepts that were different from the previous code snapshot (added or removed alike). Rule C used concept differences just like Rule B, but treating addition and removal as different instances of one concept (appending a suffix to the concept identifier in case of concept removal).

First, the AFM model is to use all parsed concepts or concepts difference lists. It is, however, safe to assume that not all concepts are relevant for solving a problem and different subsets of concepts could be relevant for each particular test the problem is verified against. To set aside the concepts that have a significant influence on the successful passing of the problem's test, we used a PC algorithm for systematic conditional independence search implemented in the Tetrad – a data-mining tool developed at Carnegie Mellon University [8]. For each problem in our three datasets we composed a data-mining problem for the PC algorithm to find a bipartite graph where arcs go from concepts to tests denoting causal links (but not between tests or concepts). We admittedly violate i.i.d. assumptions and, although we are mining for these graphs across multiple students, we are using multiple data points from the same student. However, we are not going to draw causal conclusions on the included arcs and are only using the results of the algorithm to filter out concepts. For the tests of independence we used a *p*-value of 0.05. Our experimentation with different *p*-values did not result in tangible changes of the output.

One important phenomenon we noticed in the data is that students have different submission speeds. One student might submit one code snapshot per 10-20 minutes of work, while the other would submit every change to the code with several submissions per minute. As a result, the number of attempts per code construct per unit of time would vastly differ across students and the estimations of the concept learning rates would be extremely noisy. To compensate for these differences, we applied natural logarithm function to the student opportunity counts (t_{ikj}).

Four different versions of AFM models were constructed by turning on and off of the two features: whether or not to filter concepts, and whether or not to log counts of concept opportunities, together with one Rasch and one null model, give us 14 models in total. In order to go beyond model-fitting accuracy and to check our third hypothesis and to make sure that our models can potentially serve as a basis for a component to recommend changes to the code, we ran a specialized validation procedure. In this procedure we distinguished four changes between passing and failing of a particular problem's test in successive code snapshots. Namely, from fail to fail (NN – not passing to not passing), from pass to pass (YY – passing to passing), from pass to fail (YN), and from fail to pass (NY). In each of the four cases we looked at which concepts students added and which concepts they removed between the snapshots. For additions and removals, we computed support scores – sums of concept slopes in the model giving us model's judgment in favor of all addition and all removals. These two sums were either positive (P), negative (N), or zero (0), giving us 9 different combinations. Thus each successive code snapshot was assigned a 4-letter code. For example, NYP0 would denote that a student went from failing to passing a test and the model has a positive support score for concept addition and a neutral 0-score for concepts removal. Based on these codes, for each of our models we computed four conditional probabilities.

Probability A: the non-negative support of the changes to the concepts in cases of two successful passes of the test. *Rationale*: Since in two consecutive attempts student's code passed the test, model negative support of code changes is undesirable.
Probability B: negative code changes support in the case of pass changes to fail. *Rationale*: Since students apparently made the code worse, we want the model to vote against it.
Probability C: non-positive support for the code changes in the case of two successive fails. *Rationale*: The code did not improve and the model should not support any changes made.
Probability D: positive support for the changes made between a failure and a success. *Rationale*: When a student is on the right path, the model should be supportive of that.

We performed validation with respect to the three rules of the concept selection (A – all concepts, B – changed concepts, and C – changed accounting for removals and additions) as well as filtering of the concepts (only considering slopes for concepts that were selected by the PC algorithm).

5. RESULTS

Table 2 is a summary of the model fitting and validation results for the 14 models we discussed. The dataset was balanced: with the majority class model performing only a little better than chance. The Rasch model that assumes no learning is a tangible improvement with 71% accuracy. AFM models perform better with respect to accuracy. Models considering all concepts in the snapshot (A) are doing better, and models considering changes on concepts distinguishing additions and removals (C) being second. Filtering concept lists using PC algorithm improves model accuracies, while taking logs of opportunity counts does a little bit of the opposite. Out of the top three models with respect to accuracy, two are picking all concepts available and two are using PC algorithm for concept filtering.

An important consideration is the size of the input data. More data complicates training the models as well as online-prediction of potential modifications to the code. Models using concept selection, rule A, are more data hungry. Applying the PC algorithm to only leave influential concepts reduces the data requirement. Logging opportunity counts increases the data requirement mostly due to the text representation of our data. Model accuracy and data requirements together paint a mixed picture.

Reviewing the validation columns of Table 2, We see in the average validation probabilities columns, probabilities A and C described model quality with respect to situations when a student neither improves the code nor makes it worse (in terms of passing the tests). In these cases, we would like our models to not discourage changes when students' code did not improve beyond an already passing rating (probability A) and we would like models to not support changes when students do not improve their code and the tests still fail to pass (probability C). Arguably, A and C are secondary to probabilities B and D, where we want them to positively reinforce changes from pass to fail (probability D) and negatively reinforce changes from fail to pass (probability B). In an attempt to make model selection more rigorous we take an average of all probabilities (A through D), and an average of the columns of the primary interest (B and D).

Looking at validation results alone, models with logged opportunity counts using concept selection rules A and B are in the lead, model AFM B +PC+Log has a slight edge (third and first with respect to the two averages of the conditional probabilities). This model also has a top average rank overall. It is only 5% over the accuracy of the Rasch model, but it is quite low on data requirements and performs well in the validation.

Table 2. Summary of model fitting and validation statistics. Models ranked among top three in each category are bold faced.

Model	Accuracy & rank*		File size, Mb & rank		Avg. validation prob. & rank				Overall rank
					A-D		B, D		
Null	.56	-	-	-	-	-	-	-	-
Rasch	.71	-	49	-	-	-	-	-	-
AFM A	.81	4	1312	11	.61	5	.39	7	6.75
AFM B	.73	11	446	8	.62	4	.39	8	7.75
AFM C	.78	6	445	7	.59	9	.23	12	8.50
AFM A+PC	**.84**	**1**	1528	12	.57	11	.34	10	8.50
AFM B+PC	.77	7	526	9	.60	7	.44	4	6.75
AFM C+PC	**.83**	**2**	530	10	.56	12	.30	11	8.75
AFM A+Ln	.75	10	242	5	**.62**	**2**	**.45**	**3**	5.00
AFM B+Ln	.71	12	**123**	**1**	**.63**	**1**	.43	5	**4.75**
AFM C+Ln	.77	8	**139**	**2**	.60	6	.35	9	6.25
AFM A+PC+Ln	**.82**	**3**	284	6	.59	8	**.47**	**2**	**4.75**
AFM B+PC+Ln	.75	9	**141**	**3**	**.62**	**3**	**.49**	**1**	**4.00**
AFM C+PC+Ln	.78	5	161	4	.58	10	.40	6	6.25

* Null and Rasch models are not ranked and given as a reference

It is particularly interesting whether accuracy, data requirements, and validation conditional probabilities correlate. Naturally, accuracy grows with the data necessary to fit the model and explains 35% of its variance. The average of four conditional probabilities is negatively related to the accuracy and explains 71% of its variance. However, despite the fact that the average of negative support for going from pass to fail and positive support for going from fail to pass, respectively correlates with model accuracy negatively, the percent of variance explained is low.

6. DISCUSSION

In this work we investigated the value of using student models for programming domain without a priori conceptualization of the problem domain. We hypothesized that, thanks to the inherent structure of the programming language, it could be possible to skip tedious development of a concept vocabulary overall.

Serving as a basis for navigation support, the models of student knowledge that we built could be used for recommending the next problem to solve. However, an arguably more interesting feature is to reuse the models for within-problem support. As we have shown in our validation, even in the absence of a formal conceptual domain structure, just relying on the code parser and concept selection and filtering algorithms, our models can be useful.

Based on the model accuracy, data requirement, and validation, we were able to select a model that has a promise to be accurate both modeling student knowledge and suggesting students what concepts to address in their code. The choice, however, has a number of tradeoffs. Depending on model accuracy, computational complexity of model fitting (size of the data required), and validation characteristics (potential accuracy of recommendation) one might opt to select a different model. The trade-off between modeling accuracy and validation accuracy is particularly sharp, because these two metrics are negatively correlated.

In our models, we only accounted for the presence of programming language constructs in the code, completely ignoring the number of times they were used. One particular roadblock that exists on the path toward incorporating problem-concept counts is that it would be not possible to use the PC algorithm anymore. The PC algorithm is intended for binary data only (passing of the test and presence of the concept). There are few empirically verified structural search algorithms that use block-of-conditional-independence-tests that handle hybrid data (binary and count data together).

In addition, when looking at the code, we only looked at the list of concepts and not at the structure of the code. We were able to detect certain strategies that students employed while solving the problems. In our future work, we plan to exploit these findings to improve our model's prediction and validation scores.

7. REFERENCES

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. Journal of the Learning Sciences, 4(2), 167-207.

[2] Cen, H., Koedinger, K.R., Junker, B. (2008) Comparing Two IRT Models for Conjunctive Skills. In 9th International Conference On Intelligent Tutoring Systems. (pp. 796–798). Springer, Heidelberg

[3] Hosseini, R., & Brusilovsky, P. (2013). JavaParser: A Fine-Grain Concept Indexing Tool for Java Problems. In The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013) (pp. 60-63).

[4] Hsiao, I.-H., Sosnovsky, S., and Brusilovsky, P. (2010) Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming. Journal of Computer Assisted Learning, 26 (4), 270-283.

[5] van der Linden, W.J., Hambleton, R.K. (eds.): Handbook of Modern Item Response Theory. (1997) Springer, New York

[6] Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. International Journal of Artificial Intelligence in Education, 13(2-4), 173-197.

[7] Sosnovsky, S. A., Dolog, P., Henze, N., Brusilovsky, P., & Nejdl, W. (2007). Translation of Overlay Models of Student Knowledge for Relative Domains Based on Domain Ontology Mapping. In 13th International Conference on Artificial Intelligence in Education (pp. 289-296)

[8] Spirtes, P., Glymour, C., and Scheines, R. (2000) Causation, Prediction, and Search, 2nd Ed. MIT Press, Cambridge MA.

[9] Vihavainen, A., Luukkainen, M., & Kurhila, J. (2012). Multi-faceted support for MOOC in programming. In Proceedings of the 13th annual conference on Information technology education (pp. 171-176).

[10] Vihavainen, A., Vikberg, T., Luukkainen, M., & Pärtel, M. (2013). Scaffolding students' learning using test my code. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education (pp. 117-122)

[11] Yudelson, M. & Brusilovsky, P. (2005). NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples. In 12th International Conference on Artificial Intelligence in Education (pp. 710-717).

Mining Gap-fill Questions from Tutorial Dialogues

Nobal B. Niraula
Institute for Intelligent Systems
University of Memphis
nbnraula@memphis.edu

Vasile Rus
Institute for Intelligent Systems
University of Memphis
vrus@memphis.edu

Dan Stefanescu
Institute for Intelligent Systems
University of Memphis
dstfnscu@memphis.edu

Arthur C. Graesser
Institute for Intelligent Systems
University of Memphis
graesser@memphis.edu

ABSTRACT

Gap-fill questions are fill-in-the-blank questions which consist of a sentence with one or more gaps (blanks) and a number of choices for each gap. Such questions play crucial roles in creating test materials and tutorial dialogues. In this paper, we present a system that automatically generates such questions by exploiting previously recorded student-tutor interactions with an Intelligent Tutoring System. Our method is novel because it relies on mining questions' distractors, i.e. tempting incorrect answers, from tutorial dialogues unlike most of the existing approaches that rely on instructional contents. Experimental results show that the proposed system can generate high quality gap-fill questions.

Keywords

Question Generation, Tutoring System, Dialogue Systems

1. INTRODUCTION

Test construction is an expensive and time-consuming process for instructors and educational researchers. Computer-assisted test construction can dramatically reduce costs associated with such test construction activities [8]. As a result, particular attention has been paid by researchers to automatically generate several types of questions such as *gap-fill questions* that can be used in assessment instruments [4]. The more general problem of question generation has been systematically addressed via shared tasks [11].

In this paper, we present a novel method that mines *gap-fill* questions from tutorial dialogues. *Gap-fill* questions are *fill-in-the-blank* questions which consist of a sentence/paragraph with one or more gaps (blanks). Gap-fill questions can be of two types: with alternative options (*key and distractors*) and without choices. The former ones are called *cloze* questions and the latter ones are called *open-cloze* questions. In this paper, we use the term gap-fill questions to refer to the cloze questions. Consider the following *gap-fill* question:

Newton's ____ law is relevant after the mover doubles his force as we just established that there is a non-zero net force acting on the desk then.
(a) third (b) second (c) first (d) heating

One of the options in a *gap-fill* question is the correct answer to the question, called the *key.* The rest of the choices are the *distractors*, i.e. incorrect answers that are tempting less proficient students who often confuse them with the *key.* The question sentence containing gap(s) is also known as the *stem.* In the gap-fill question above, the question sentence contains a gap and there are four potential choices for the gap. The *key* is *second* and *first*, *third* and *heating* are three distractors. Two of distractors are very close to the key while another, *heating*, is quite remotely related.

The attractiveness of gap-fill questions is that they are well-suited for automatic marking because the correct answer is simply the original word corresponding to the gap in the original sentence. Furthermore, gap-fill questions are effective at diagnosing and assessing students' knowledge [5]. Many automatic gap-fill question generation techniques are reported in the literature [7, 12]. These techniques have been successfully used even in large scale evaluations (e.g. TOEFL[1] and TOEIC[2]) to measure learners' proficiency at various tasks, e.g. assessing second language learners' skills of the target language. Gap-fill questions are also important in Intelligent Tutoring Systems (ITSs)[2], a category of advanced educational systems that emphasizes interaction, active learning, and adaptation to the learner. Specifically, ITSs use such questions for assessing students' knowledge level and learning gains as part of their assessment. Furthermore, ITSs use such questions for scaffolding in their practice modules. We explain next the role of gap-fill questions for scaffolding purposes in dialogue-based ITSs.

In dialogue based or conversational ITSs, students typically solve problems (a.k.a. instructional tasks) with the help of the system. That is, during a tutoring session students are prompted to provide complete solutions to various problems. If some of the steps in their solutions are missing, the computer tutor will provide appropriate scaffolding through

[1] http://www.ets.org/toefl
[2] http://www.ets.org/toeic

the use of hints, some in the form of *open-cloze* questions, to help students articulate missing or vague parts. Table 1 shows a fragment of a real student-tutor interaction from the intelligent tutoring system DeepTutor [9]. The first student response (to a previous hint - not shown) is incorrect and therefore the system decides to provide a more informative hint in the form of a *open-cloze* hint/statement/question.

Table 1: A fragment of student-tutor interactions while solving a *task*.

... ...
STUDENT: *Netwon's first law*
TUTOR: Let me give you a hint. The decomposition principle says that the analyses of forces and motion along two ______ directions, such as horizontal and vertical, can be done ______.
STUDENT: *perpendicular, separately*

In this paper, we propose a novel method to automatically generate *gap-fill* questions by exploiting recorded data from massive online education environments such as DeepTutor [10]. In such massive online courses (MOOCs) or massive online ITSs (MOITSs) instructional *tasks or problems* are solved by many students. Consequently, many student responses to hints in the form of questions, some of which are *open-cloze* questions, are collected and recorded in log files. Our approach here exploits this richness of information available in recorded tutorial dialogues from massive online training with ITSs. An advantage of mining these tutorial dialogues is the fact that we have access to actual students answers to *open-cloze* questions. That is, students' responses to these questions are words that they think best fill in the gaps in the *open-cloze* questions. Because not all responses are correct, we consider the incorrect responses as potential candidates for distractors. We rank these candidate distractors to find the best set of distractors. We will show later that this simple idea generates very good distractors for gap-fill questions.

2. RELATED WORKS

Before presenting the most related previous work, we describe the four main steps needed to generate gap-fill questions with choices from instructive texts or content-related documents. Understanding the four main steps will help better appreciate related work. The four main steps are : a) Selecting useful sentences from the text b) Identifying gaps (i.e. words to be deleted) in the selected sentences c) Generating distractor candidate list and d) Ranking the distractors in the list. The literature of gap-fill question generation contains methods that go through each of the steps or focus on particular steps.

Mitkov et al. [4] proposed a computer-aided procedure to generate multiple-choice questions from textbooks that goes through all the four steps. They find key terms by using regular expressions and thresholds. Hypernyms and coordinates of the terms are considered the distractors. The ranking of distractors is done using semantic similarity functions on the assumption that a distractor should be as semantically close to the key as possible. Agarwal and Mannem [1] also go through all four steps to automatically generates gap-fill questions from textbooks for reading comprehension tests.

Hoshino and Nakagawa[3] modeled the problem of generating multiple-choice questions as a learning problem. To decide whether a given word can be left blank in the declarative stem, they trained classifiers using a training data set. The distractors were random words from the same article excluding punctuation and the same word. Sumita *et al.* [13] generated gap-fill questions considering verbs as gaps in a sentence. Thesaurus was used to obtain distractors for the keys of the gaps. To rank distractors, they took each distractor, filled the gap using it, and searched the Web to get the hit counts of the sentence. Smith *et al.* [12] generated cloze questions in English language learning. They used distributional thesaurus to find distractors.

As one may note, most of these solutions require instructional texts such as textbook chapters and encyclopedia entries in addition to thesauri to generate gap-fill questions. Our method is unique because it is based on a generative approach, i.e. the potential distractors are generated by students themselves. Thus, our approach which works by mining questions and distractors from recorded dialogues complements the existing literature. Furthermore, this is the first approach, to the best of our knowledge, that relies on actual student answers to generate distractors.

3. THE METHODOLOGY

Since we do not start with instructional texts but with students' responses to *open-cloze* questions, we only need to generate distractors and rank them in order to generate gap-fill questions.

3.1 Generating Distractor Candidates

Finding plausible distractors that separate knowledgeable students from knowledge-poor students is one of the major challenges for cloze question generation. A good distractor is a concept that is semantically similar at some extent to the key but it is not a correct answer [4].

As already mentioned, we use student responses to *open-cloze* questions during tutorial dialogues as a source of distractors. When same open-cloze questions are answered by many students, there is a large pool of candidate distractors from which to select. We show in Table 2 an open-cloze question together with student responses and their *counts* or *votes*, i.e. the number of students that give the same response as the answer to the same question. Some open-cloze questions may not have enough student responses. In such cases, we follow some of the existing techniques for finding distractor candidates, e.g. we use WordNet as in [4]: extract the hypernyms and coordinated concepts (concepts with the same hypernym) of the key and consider them as the distractor candidates for the key.

3.2 Ranking Distractors

We used the following criteria to rank candidate distractors:
R1: *Use a semantic similarity score between the key and distractors.* This idea was used in the past by Mitkov *et al.* [4]. According to them, a good distractor is very related but not identical to the key. We used a Latent Semantic

Table 2: Students' responses and their frequencies (i.e. votes) to an open-cloze question.

While the wind is blowing, the shape of the sled's path will be ____.

curved => 4	no => 1	idk => 1
straight => 3	linear => 1	a triangle => 1
diagonal => 2	uhm no => 1	west => 1

Analysis (LSA) based similarity measure [6] to compute the similarity between a key and its distractors.

R2: *Use votes.* We rank the candidates based on their votes/counts (the higher, the better). We break the tie using the semantic similarity score with the key.

4. EXPERIMENTS AND RESULTS

We mined a collection of tutorial dialogues obtained from two of our experiments with the DeepTutor system ([9]). From the first experiment, we extracted tutorial dialogues for 297 students who solved 32 tasks (problems). Since a task was solved by zero or more students, we had 2,687 task sessions altogether (i.e. 9 tasks per student on average). Similarly, from the second experiment, we extracted 4,430 task sessions corresponding to 349 students and 37 tasks (i.e. 13 tasks per student on average). A total of 102 unique single-gap open-cloze questions were also mined. It is noted that some of the questions received a large number of responses while some others only a few. All of the single gap open-cloze questions received at least two responses, 82.85% of the questions received at least three responses, and 72.38% of questions received at least 4 responses.

4.1 Relation between a Response's Similarity and its Rank

We define the frequency rank (*FR*) of a student response i to a hint in the form of a open-cloze question q as : $FR(i) = \frac{100 * f_i}{\sum f_i}$ where f_i is the number of students who typed i as the answer to open-cloze question q (i.e. votes or counts of i). Then for each response, i.e. which could be either a correct response or candidate distractor, we computed its similarity with the corresponding key as well as its *FR* score. We discarded the student responses that were misspelled or contained emoticons. We used a small lexicons of emoticons for this purpose. Next, we computed correlation coefficients between the similarities and FR scores at different levels of response frequencies (see Table 3). The correlation coefficients for all responses (i.e. minimum frequency >=1) and for responses generated by at least two students (i.e. minimum frequency >=2) were 0.682 and 0.720 respectively. Similarly, the coefficients for responses with minimum frequencies of 3, 4, and 5 were 0.737, 0.733 and 0.754 respectively.

The positive correlation coefficients indicate that there is clearly a positive relation between the frequency of a response and its semantic similarity score. As we noticed, the correlation coefficients increased as we increased the minimum frequency. These results suggest that ranking student responses by their semantic similarity scores with the key can be approximated by their vote counts, i.e. how many students generated the answer. The higher the counts, the more similar the response is to the key. As the distractors for a key should be as semantically close to the key as possible, we can rank the responses by their votes and utilize them as potential distractors.

Table 3: Correlation between *Sim(key,responses)* & *FR(responses)* for responses with *freq >= Min Freq*

Min Freq	1	2	3	4	5
Correlation_LSA	0.682	0.725	0.737	0.733	0.754

4.2 Evaluation of Distractor selection

We conducted two evaluations to determine the quality of the distractors generated by our automated method. In each evaluation, we asked two annotators to rate each distractor with one of the following quality ratings: *good*, *ok*, and *bad*. The *good* distractors are ideal distractors, the *ok* distractors can be considered as potential distractors but are not as appropriate as the good distractors. The *bad* distractors do not make sense as a distractor or have the exact meaning with the key.

In the first evaluation, we considered questions that had at least three different student responses and had at least two votes per response. There were 23 questions that satisfied this condition. We ranked the distractor candidates by using *R2* as presented in Section 3.2 and chose the top 3 candidates as distractors. We rejected the candidates if they were synonyms of the key. We considered a key and distractor synonyms when their semantic similarity score was above or equal to 0.9. We also removed duplicate distractors in the final list. To reduce the annotation bias, we introduced a random word from a Wikipedia article as the fourth distractor. The order of the four distractors were randomized.

Next, we asked the annotators to annotate the instances, each consisting of a question sentence, its key and the distractors. A typical annotated instance is showed in the Table 4. The inter-rater agreement using the unweighted version of the Cohen's kappa statistic was 0.64 when we considered *good*, *ok* and *bad* groups separately. It increased to 0.86 when we merged *good* and *ok* groups into a single group. The detailed annotation results are presented in Table 5. The proportion of the good questions is the highest for both annotators. Since we introduced one bad distractor per question and we had 23 questions, we expected at least 23 bad distractors per annotator. Discounting this number in the table, one can notice that we can achieve very good distractors using the voting scheme.

Table 4: Sample Annotation

Question	The force of gravity exerted by the Earth on the cat is ___ all the time.			
Key	constant			
Distractors	relative	horizontal	zero	smile
Annotation	good	good	good	bad

Table 5: Annotation results for 23 questions with 4 distractors each

	good	*ok*	*bad*	*expected bad*
Annotator 1	46	11	35	23
Annotator 2	41	16	35	23

In a second evaluation, we addressed the case when we could not get sufficient distractors for a key due to too few responses available in our tutorial dialogue dataset. We had 100 such questions in our corpora. In such cases, we generated distractor candidates for a question from three different sources: student responses corresponding to the question, different questions with the same key, and WordNet. For each candidate, we checked whether its parts-of-speech matched with that of the key. If matched, we marked the candidate as a potential distractor for the key. Once we had three potential distractors, we stopped. The fourth distractor was a random word from Wikipedia. Annotation results showed that WordNet-based approach could generate distractors out of context. For example, it generated *one*, *two*, and *three* as the three distractors for the key *zero* for the question: *The net force is* __. The three candidates may look good but for the given question, they are bad distractors. Since the students' responses are mostly contextual,they are preferred over the WordNet-based distractors.

4.3 Error Analysis

The most challenging issue was finding similarities between student answers and the key. Although word pairs such as (*is*, *equals*), (*vertical*, *y-direction*), (*identical*, *constant*) include words with same meaning in the context of Newtonian Physics, LSA failed to find that due to lack of domain knowledge. Use of numbers (e.g. 1st for first, 9.8m/s for constant acceleration) and misspellings of the words (e.g. *seperately* for *separately*, *thirrd* for *third*, *on* for *no*) in students responses were other factors limiting the performance of the proposed approach.

5. CONCLUSION AND FUTURE WORK

We presented in this paper a unique method to generate gap-fill questions. We also proposed different ranking functions to prioritize the list of potential distractor candidates. Since we exploit the students responses corresponding to a problem, our approach would be particularly useful for scalable ITSs and MOOCs and where thousands of students solve the same problem. In future, we exploit the open-cloze questions with multiple gaps to generate more gap-fill questions.

6. ACKNOWLEDGMENTS

This research was supported in part by Institute for Education Sciences under awards R305A100875. Any opinions, findings, and conclusions or recommendations expressed in this material are solely the authors'.

7. REFERENCES

[1] M. Agarwal and P. Mannem. Automatic gap-fill question generation from text books. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–64. Association for Computational Linguistics, 2011.

[2] A. C. Graesser, S. D'Mello, X. Hu, Z. Cai, A. Olney, and B. Morgan. Autotutor. In P. M. McCarthy and C. Boonthum-Denecke, editors, *Applied Natural Language Processing: Identification, Investigation and Resolution*, pages 169–187. PA: IGI Global, 2012.

[3] A. Hoshino and H. Nakagawa. A real-time multiple-choice question generation for language testing: a preliminary study. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 17–20. Association for Computational Linguistics, 2005.

[4] R. Mitkov, L. A. Ha, and N. Karamanis. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2):177–194, 2006.

[5] R. Mitkov, L. A. Ha, A. Varga, and L. Rello. Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 49–56. Association for Computational Linguistics, 2009.

[6] N. Niraula, R. Banjade, D. Ştefănescu, and V. Rus. Experiments with semantic similarity measures based on lda and lsa. In *Statistical Language and Speech Processing*, pages 188–199. Springer, 2013.

[7] J. Pino, M. Heilman, and M. Eskenazi. A selection strategy to improve cloze question quality. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada*, pages 22–32, 2008.

[8] M. Pollock, C. Whittington, and G. Doughty. Evaluating the costs and benefits of changing to caa. In *Proceedings of the 4th CAA Conference*, 2000.

[9] V. Rus, S. D'Mello, X. Hu, and A. C. Graesser. Recent advances in conversational intelligent tutoring systems. *AI Magazine*, 34(3), 2013.

[10] V. Rus, D. Stefanescu, N. Niraula, and A. C. Graesser. Deeptutor: Towards macro- and micro-adaptive conversational intelligent tutoring at scale. In *Work in Progress Learning At Scale*, 2014.

[11] V. Rus, B. Wyse, P. Piwek, M. Lintean, S. Stoyanchev, and C. Moldovan. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 251–257. Association for Computational Linguistics, 2010.

[12] A. K. S. Smith, P. Avinesh, and A. Kilgarriff. Gap-fill tests for language learners: Corpus-driven item generation. In *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*, 2010.

[13] E. Sumita, F. Sugaya, and S. Yamamoto. Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 61–68. ACL, 2005.

Online Optimization of Teaching Sequences with Multi-Armed Bandits

Benjamin Clement Didier Roy Pierre-Yves Oudeyer

Manuel Lopes
Inria Bordeaux Sud-Ouest,
France
{FirstName.LastName}@inria.fr

ABSTRACT

We present an approach to Intelligent Tutoring Systems which adaptively personalizes sequences of learning activities to maximize skills acquired by each student, taking into account limited time and motivational resources. At a given point in time, the system tries to propose to the student the activity which makes him progress best. We introduce two algorithms that rely on the empirical estimation of the learning progress, one that uses information about the difficulty of each exercise **RiARiT** and another that does not use any knowledge about the problem **ZPDES**.

The system is based on the combination of three approaches. First, it leverages recent models of intrinsically motivated learning by transposing them to active teaching, relying on empirical estimation of learning progress provided by specific activities to particular students. Second, it uses state-of-the-art Multi-Arm Bandit (MAB) techniques to efficiently manage the exploration/exploitation challenge of this optimization process. Third, it leverages expert knowledge to constrain and bootstrap initial exploration of the MAB, while requiring only coarse guidance information of the expert and allowing the system to deal with didactic gaps in its knowledge.

1. INTRODUCTION

Intelligent Tutoring Systems (ITS) have been proposed to make education more accessible, more effective and simultaneously as a way to provide useful objective metrics on learning. Recently, online learning systems have further raised the interest in these systems and several recent projects started on Massive Open Online Course (MOOC) for web-based teaching of university level courses. For a broad coverage on the field of ITS see [9] and [13].

According to [13], there are four main components of an ITS: i) a *cognitive model* that defines the domain knowledge or which steps need to be made to solve problems in a particular domain; ii) a *student model* that considers how students learn, what is the evolution of their cognitive state depending on particular teaching activities; iii) a *tutoring model* that defines, based on the cognitive and the student model, what teaching activities to present to students and iv) a *user interface model* that represents how the interaction with the students occurs and how problems are proposed to the learners.

In this work we are more focused on the *tutoring model*, that is, how to choose the activities that provide a better learning experience based on the estimation of the student competence levels and progression, and some knowledge about the cognitive and student model. We can imagine a student wanting to acquire many different skills, e.g. adding, subtracting and multiplying numbers. A teacher can help by proposing activities such as: multiple choice questions, abstract operations to compute with a pencil, games where items need to be counted through manipulation, videos, or others. The challenge is to decide what is the optimal sequence of activities that maximizes the average competence level over all skills.

There are several approaches to develop a *Tutoring Model*. A first approach is based on hand-made optimization and on pedagogical theory, experience and domain knowledge. There are many works that followed this line, see the recent surveys on the field by [9, 13]. A second approach considers particular forms of knowledge to be acquired and creates didactic sequences that are optimal for those particular classes of problems [2, 6, 7]. A third approach, and more relevant for our work, is that the optimization is made automatically without particular assumptions about the students or the knowledge domain. The framework of partial-observable Markov decision process (POMDP) has been proposed to select the optimal activities to propose to the students based on the estimation of their level of acquisition of each KC [14].

Our ITS aims at providing to each particular student the activities that are giving the highest learning progress. We do not consider that these activities are necessarily the ones defined a-priori in the cognitive and student model, but the ones that are estimated, at runtime and based on the students results, to provide the maximum learning gain. This approach has three main advantages:

Weaker dependency on the cognitive/student model
In most cases the tutoring model incorporates the student model inside. Given students' particularities, it is often highly difficult or impossible for a teacher to understand

all the difficulties and strengths of individual students and thus predict which activities provide them with maximal learning progress. Also, typically, these models have many parameters, and identifying all such parameters for a single student is a very hard problem due to the lack of data, the intractability of the problem and the lack of identifiability of many parameters that often results in models which are inaccurate in practice [3]. It has been shown that a sequence that is optimal for the average student is often suboptimal for most students, from the least to the most skilled [11].

We consider that it is important to be as independent as possible of the cognitive and student model when deciding which activities to propose. This requires that the ITS explores and experiments various activities to estimate their potential for learning progress for each student. The technical challenge is that these experiments should be not just sufficiently informative about the student's current competence but also to evaluate the effectiveness of each exercise to improve those competences (a form of stealth assessment [16]).

Efficient Optimization Methods We will rely on methods that do not make any specific assumptions about how students learn and only require information about the estimated learning progress of each activity. We make a simple assumption that activities that are currently estimated to provide a good learning gain, must be selected more often. A very efficient and well studied formalism for these kind of problems is Multi-Armed Bandits [5]. Following a casino analogy, at each step we can choose a slot-machine and we get to observe the payback we get, the goal it to find the best arm, but while we are trying to discover it we have to bet to test them.

More Motivating Experience Our approach considers that, at each time instance, the exercises that are providing the higher learning progress must be the ones proposed. This allows not only to use more efficient optimization algorithms but also to provide a more motivating experience to students. Several strands of work in psychology [4] and neuroscience [8] have argued that the human brain feels intrinsic pleasure in practicing activities of optimal difficulty or challenge, i.e. neither too easy nor too difficult, but slightly beyond the current abilities, also known as the zone of proximal development [10].

Our main contributions, when compared to other ITS systems, are: the use of highly performing Multi-Armed Bandit algorithms [5]; a simpler factored representation of the cognitive model that maps activities to the minimum necessary competence levels; and considering that the acquisition of a KC is not a binary variable but defined as the level of comprehension of that KC. The advantage of using MAB is that they are computationally efficient and require a weaker dependency between the tutoring and the cognitive and student models. Other contributions include an algorithm to estimate student competence levels; and the empirical learning progress of each activity. An extended version of this article is available at [12] including an initial user study.

2. ITS WITH MULTI-ARMED BANDITS

2.1 Relation between KC and pedagogical activities

In general, activities may differ along several dimensions and may take several forms (e.g. video lectures with questions at the end, or interactive games or exercises of various types). Each activity can provide opportunities to acquire different skills/knowledge components (KC), and may contribute differentially to improvement over several KCs (e.g. one activity may help a lot in progressing in KC_1 and only little in KC_2). Vice versa, succeeding in an activity may require to leverage differentially various KCs. While certain regularities of this relation may exist across individuals, it will differ in detail for every student.

First, we model here the competence level of a student in a given KC as a continuous number between 0 and 1 (e.g. 0 means not acquired at all, 0.6 means acquired at 60 percent, 1 means entirely acquired). We denote c_i the current estimation of this competence level for knowledge unit KC_i. In what we call a R Table, for each combination of an activity $\mathbf{a}$ and a KC_i, the expert then associates a $q-$value ($q_i(\mathbf{a})$) which encodes the competence level required in this KC_i to have maximal success in this activity $\mathbf{a}$. This in turn provides a upper and lower bound on the competence level of the student: below $q_i(\mathbf{a})$ in case of mistake; above $q_i(\mathbf{a})$ in case of answering correctly.

We start by assuming that each activity is represented by a set of parameters $\mathbf{a} = (a_1, ..., a_{n_a})$. The R Table then uses a factorized representation of activity parameters, where instead of considering all $(\mathbf{a}, KC_i)$ combinations and their corresponding $q_i(\mathbf{a})$, we consider only (a_j, KC_i) combinations and their corresponding $q_i(a_j)$ values, where $q_i(a_j)$ denotes the competence level in KC_j required to succeed entirely in activity $\mathbf{a}$ which $j-th$ parameter value is a_j. This factorization makes the assumption that activity parameters are not correlated. The alternative would require a larger number of parameters and would also require more exploration in the optimization algorithm. We use the factorized R Table in the following manner to heuristically estimate the competence level $q_i(\mathbf{a})$ required in KC_i to succeed in an activity parameterized with $\mathbf{a}$: $q_i(\mathbf{a}) = \prod_{j=1}^{n_a} q_i(a_j)$

2.2 Estimating the impact of activities over students' competence level in knowledge units

Key to the approach is the estimation of the impact of each activity over the student's competence level in each knowledge unit. This requires an estimation of the current competence level of the student for each KC_i. We do not want to introduce regular tests that might interfere negatively with the learning experience of the student. Thus, competence levels need to be inferred through stealth assessment [16] that uses indirect information from the results on the exercises.

When doing an activity $\mathbf{a} = (a_1, ..., a_{n_a})$, the student can either succeed or fail. In the case of success, if the estimated competence level c_i in knowledge unit i is lower than $q_i(\mathbf{a})$, we are underestimating the competence level of the student in KC_i, and so should increase it. If the student fails and $q_i(\mathbf{a}) < c_i$, then we are overestimating the competence level of the student, and it should be decreased. For these two first cases we can define a reward:

$$r_i = q_i(\mathbf{a}) - c_i \quad (1)$$

and use it to update the estimated competence level of the student according to $c_i = c_i + \alpha r_i$ where α is a tunable parameter that allows to adjust the confidence we have in each new piece of information.

A crucial point is that the quantity $r_i = q_i(\mathbf{a}) - c_i$ is not only used to update c_i, but is used to generate an internal reward $r = \sum r_i$ to be cumulatively optimized for the ITS (details below). Indeed, we assume here that this is a good indicator of the learning progress over KC_i resulting from doing an activity with parameters $\mathbf{a}$. The intuition behind this is that if you have repeated successes in an activity for which the required competence level is higher than your current estimated competence level, this means you are probably progressing.

2.3 RiARiT: Right Activity at Right Time

To address the optimization challenge for ITS, we will rely on multi-arm bandit techniques (MAB)[5]. A particularity here is that the reward (learning progress) is non-stationary, which requires specific mechanisms to track its evolution. Indeed, here a given exercise will stop providing reward, or learning progress, after the student reaches a certain competence level. Also we cannot assume that the rewards are i.i.d. as different students will have different preferences and many human factors, i.e. distraction, mistakes on using the system, create several spurious effects. Thus, we rely here on a variant of the EXP4 algorithm [1, 5]. We consider a set of filters that track how much reward each exercise parameters is giving. Then the algorithm selects stochastically the teaching activities proportionally to the expected learning progress for each parameter.

Expert knowledge can also be used by incorporating *coarse* global constraints on the ITS. Indeed, for example the expert knows that for most students it will be useless to propose exercises about decomposition of real numbers if they do not know how to add simple integers. Thus, the expert can specify minimal competence levels in given KC_i that are required to allow the ITS to try a given parameter a_j of activities.

2.4 ZPDES: Zone of Proximal Development and Empirical Success

Our goal is to reduce the dependency on the cognitive and student models and so we will try to simplify further the algorithm. Our simplification will take two sources of inspiration: **zone of proximal development** and the **empirical estimation of learning progress**.

As discussed before focusing teaching in activities that are providing more learning progress can act as a strong motivational cue. Estimating explicitly how the success rate on each exercise is improving will remove the dependency on the R table. For this we replace Eq. 1 with $r = \sum_{k=1}^{t} \frac{C_k}{t} - \sum_{k=1}^{t-d} \frac{C_k}{t-d}$ where $C_k = 1$ if the exercise at time k was solved correctly. The equation compares the $d+1$ more recent success with all the previous past, providing an empirical measure of how the success rate is increasing. We no longer estimate the competence level of the student, and directly use the reward estimation.

The other inspiration is the concept of the zone of proximal development [10] that considers that activities that are slightly beyond the current abilities of the learner are the more motivating. This concept will provide three advantages: improve motivation; further reduce the need of quantitative measures for the educational design expert; and provide sequence of activities that follow a more sequential order. A first point is that there are some parameters that have a clear relation of increasing complexity (such as the parameter exercise type) and should be treated differently than other parameters that do not have such ordering (for instance the complexity in the modality presentation will change depending on each student and not on the problem itself). A final point is that we are choosing exercises based on the estimated (recent) past learning progress, and if we know which exercise is next in terms of complexity then we can use that one. This information, if correct, allows the MAB to propose the more complex exercises without requiring to estimate their value first. Providing a more predictive behavior and not just relying on the recent past.

This algorithm is identical to RiARiT but we treat the parameters that have a clear relation of increasing complexity differently. For the parameter i, when the expected learning progress of parameter j is below the level of the more complex parameter value, $w_i(j) < w_i(j+1)/\theta$, and the success rate is higher than a pre-defined threshold : $\sum_{k=1}^{t} \frac{C_k(j)}{t} > \omega$, we allow the parameter value $j+3$ to be chosen and initiate it with: $w_i(j) = 0$ and $w_i(j+3) = w_i(j+2)$.

3. TEACHING SCENARIO

We will now describe a specific teaching scenario about learning how to use money, typically targeted to students of 7-8 years old. The parameters of the activities are commonly used in schools for acquiring these competences and there are already well studied teaching sequences validated in several studies [15].

In each exercise, one object is presented with a given tagged price and the learner has to choose which combination of bank notes, coins or abstract tokens need to be taken from the wallet to buy the object, with various constraints depending on exercises parameters. The five Knowledge Components aimed at in these experiments are: **KnowMoney**: Global skill characterizing the capability to handle money to buy objects in an autonomous manner; **SumInteger**: Capability to add and subtract integer numbers; **DecomposeInteger**: Capability to decompose integer numbers into groups of 10 and units; **SumCents:** Capability to add and subtract real numbers (cents); **DecomposeCents**: Capability to decompose real numbers (cents); **Memory**: Capability to memorize a number which is presented and then removed from visual field.

The various activities can be parameterized with the following properties: **Exercise Type** depending on the complexity of decomposing a price[1] that can be read directly by making the correspondence to a real note/coin $a = (1, 2, 5)$ and those that need a decomposition that requires more than one item $b = (3, 4, 6, 7, 8, 9)$. The exercises will be generated by choosing prices with these properties in a set of six levels of increasing difficulty and picking an object that is priced realistically.; **Price Presentation**: i) written and spoken; ii) written; iii) spoken; **Cents Notation**: i) x€x; ii) x, x€; **Money Type**: i) Real euros; ii) Money Tokens.

[1] In the euro money system the money items (bills and coins) have the values 1, 2 and 5 for the different scales.

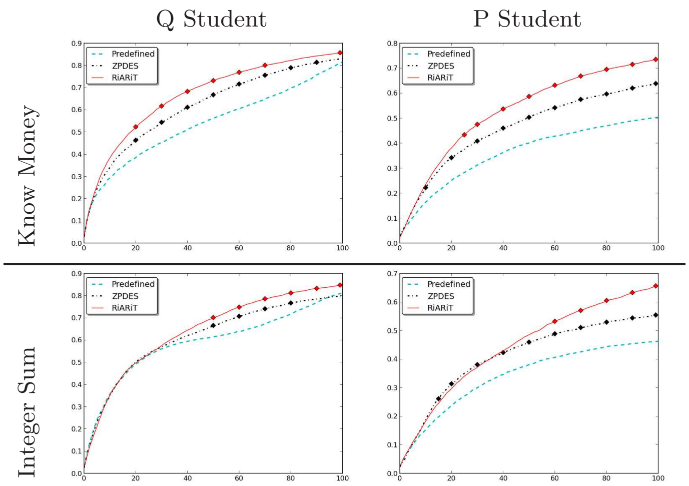


Figure 1: The evolution of the comprehension of two knowledge components with time for population . Markers on the curve mean that the difference is significative.

4. SIMULATIONS

We present a set of simulations with virtual students. We consider two populations. A population ”Q” where the students have different learning rates and maximum comprehension levels for each KC and another population ”P” where, in addition to this, the students have limitations in the comprehension of specific parameterizations of the activities. We expect that in the population ”Q” an optimization will not provide big gains because all students are able to use all exercises to progress. On the other hand, the population ”P” will require that the algorithm finds a specific teaching sequence for each particular student. We note that the algorithm itself is not provided with any a-priori information about the properties of the students. We present here the results showing how fast and efficiently our algorithms estimate and propose exercises at the correct level of the students. Each experiment considers a population of 1000 students generated using the previous methods and lets each student solve 100 exercises.

Figure 1 shows the skill’s levels evolution during 100 steps. For Q student, learning with RiARiT and ZPDES is faster than with the predefined sequence, but at the end, Predefined catch up with ZPDES. For P simulations, as students can not understand particular parameter values, they block on stages where the predefined sequence does not propose exercises adequate to their level, while ZDPES, by estimating learning progress, and RiARiT, by considering the estimated level on all KC and parameter’s impact, are able to propose more adapted exercises.

5. CONCLUSIONS AND FUTURE WORK

In this work we proposed a new approach to intelligent tutoring systems. We showed through simulations and empirical results that a very efficient algorithm, that tracks the learning progress of students and proposes exercises proportionally to the learning progress, can achieve very good results. Using as baseline a teaching sequence designed by an expert in education [15], we showed that we can achieve comparable results for homogeneous populations of students, but a great gain in learning for populations of students with larger variety and stronger difficulties. In most cases, we showed that it is possible to propose different teaching sequences that are fast to adapt and personalized. We introduced two algorithms RiARiT that uses some information about the difficulty about the task, an another algorithm ZPDES that does not use any information about the problem. It is expected that RiARiT, as it uses more information, behaves better when the assumptions are valid, while ZPDES, without any information can not achieve as high performance in well behaved cases but is surprisingly good without any information. Even when compared with a hand optimized teaching sequence ZPDES shows better adaptation to the particular students’ difficulties.

6. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

[2] F. J. Balbach and T. Zeugmann. Recent developments in algorithmic teaching. In *Inter. Conf. on Language and Automata Theory and Applications*, 2009.

[3] J. E. Beck and X. Xiong. Limits to accuracy: How well can we do at student modeling? In *Educational Data Mining*, 2013.

[4] D. Berlyne. *Conflict, arousal, and curiosity.* McGraw-Hill Book Company, 1960.

[5] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Stochastic Systems*, 1(4), 2012.

[6] M. Cakmak and M. Lopes. Algorithmic and human teaching of sequential decision tasks. In *AAAI Conference on Artificial Intelligence*, 2012.

[7] J. Davenport, A. Rafferty, M. Timms, D. Yaron, and M. Karabinos. Chemvlab+: evaluating a virtual lab tutor for high school chemistry. In *Inter. Conf. of the Learning Sciences (ICLS)*, 2012.

[8] J. Gottlieb, P.-Y. Oudeyer, M. Lopes, and A. Baranes. Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in Cognitive Sciences*, 17(11):585–593, 2013.

[9] K. R. Koedinger, E. Brunskill, R. S. Baker, E. A. McLaughlin, and J. Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 2013.

[10] C. D. Lee. Signifying in the zone of proximal development. *An introduction to Vygotsky*, 2:253–284, 2005.

[11] J. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Inter. Conf. on Educational Data Mining*, 2012.

[12] M. Lopes, B. Clement, D. Roy, and P.-Y. Oudeyer. Multi-armed bandits for intelligent tutoring systems. *arXiv:1310.3174 [cs.AI]*, 2013.

[13] R. Nkambou, R. Mizoguchi, and J. Bourdeau. *Advances in intelligent tutoring systems*, volume 308. Springer, 2010.

[14] A. Rafferty, E. Brunskill, T. Griffiths, and P. Shafto. Faster teaching by pomdp planning. In *Artificial Intelligence in Education*, 2011.

[15] D. Roy. Usage d’un robot pour la remédiation en mathématiques. Technical report, 2012.

[16] V. J. Shute. Stealth assessment in computer-based games to support learning. *Computer games and instruction*, 55(2):503–524, 2011.

Predicting MOOC Performance with Week 1 Behavior

Suhang Jiang
School of Education
University of California, Irvine
Irvine, CA 92697
suhangj@uci.edu

Adrienne E.Williams
School of Biological Sciences
University of California, Irvine
Irvine, CA 92697
adriw@uci.edu

Katerina Schenke
School of Education
University of California, Irvine
Irvine, CA 92697
kschenke@uci.edu

Mark Warschauer
School of Education
University of California, Irvine
Irvine, CA 92697
markw@uci.edu

Diane O'Dowd
School of Biological Sciences
University of California, Irvine
Irvine, CA 92697
dkodowd@uci.edu

ABSTRACT

Prior studies on Massive Open Online Courses (MOOCs) suggest that there is a significant decrease in student participation after one week of instruction [8]. This paper uses a combination of students' Week 1 assignment performance and social interaction within the MOOC to predict their final performance in the course. The study also examines the role external incentives in final MOOC performance. Using logistic regression as a classifier, we are able to predict the probability of students earning certificates for completion of the MOOC, as well as the type of certificate (i.e. Distinction and Normal) earned, with high accuracy.

Keywords

MOOC; Performance Prediction; Incentive

1. INTRODUCTION

In less than two years, Massive Open Online Courses (MOOCs) have attracted millions of learners to enroll in courses such as the History of Chinese Architecture, Information Visualization, and Healthcare Innovation and Entrepreneurship. These online courses provide a diverse set of learners with opportunities to engage in lifelong learning. Institutions of higher education are in a unique position concerning MOOCs; many universities such as MIT and Stanford supply the content of MOOCs, and many university students are encouraged to take MOOCs to expand their existing knowledge base or access courses that are not available at their home institution. However, against a background of thriving enrollment, the completion rate of MOOCs is staggeringly low [3]. Previous research indicates that fewer than 7% of learners who enroll in a MOOC actually complete it [6]. Of even more concern is that the significant decrease in participation usually takes place by the second week of the course [8]. If institutions of higher education want to take advantage of the potential of MOOCs in student learning, it is important to understand what variables affect the completion of these courses as well as at what kinds of interventions can be designed to encourage persistence. As such, this paper uses learner's behavior in the first week of a Biology MOOC to predict performance at the end of the course.

2. RELATED WORK

A number of studies have explored students' behaviors and learner's engagement patterns in MOOCs to understand issues of persistence. In one study, Balakrishnan [1] looked at student retention in a MOOC offered by UC Berkeley. He used variables related to the (1) cumulative time students spent in watching video, (2) the number of posts viewed in the forum, (3) the number of posts created on the forum, and (4) the amount of time spent on the progress page. Using these variable and Hidden Markov Models, Balakrishnan was able to predict students' likelihood to drop out of the MOOC. These measures of student engagement are likely factors in predicting student retention in MOOCs. While learner engagement seems to be a core component in answering the question of persistence in MOOCs, there still remain questions about other variables involved in understanding the complex problem of MOOC persistence. Patterns of student behavior in MOOCs can tell us something about the types of activities that are known to be engaging. For example, Kizilcec and his colleagues [4] identified four prototypical engagement patterns in a MOOC that consisted of watching videos and taking quizzes. These patterns were: (1) students who completed the majority of assessment, (2) students who engaged mainly in terms of watching videos, (3) students who did assessment at the beginning of the course, and (4) students who only watched videos for one or two assessment periods. The completing category of students is respectively 27%, 8% and 5% in the three sampled courses.

In addition to thinking about individual characteristics that predict persistence of MOOCs, an understanding of situated theories of learning is useful in examining learner persistence in MOOCs [5]. Situated learning theory posits that knowledge resides primarily in social interactions and only secondarily in the individual [2]. In the context of MOOCs, the social interactions learners have with one another via social networks could offer additional explanations for persistence in MOOCs that transcend individual learner characteristics. The role of social networking in MOOCs has been explored in a few MOOC studies. In an initial study, Yang and her colleagues [7] modeled how students' social positioning predicts their dropout using survival analysis. Nevertheless, little research integrates both the predictors of academic assessment and social interaction in modeling students' performance.

Finally, the larger context of incentives to complete a MOOC adds yet another dimension to our understanding of learner persistence. In traditional university environments, grades provide learners with a huge incentive to attend class and engage with the material. Such an incentive does not exist in MOOCs, where completion commonly results in a digital badge or a certificate that often contains little value. Often times, learners enroll in MOOCs out of intrinsic interest in the material and not because the MOOC is required for their undergraduate or graduate degree. The current study is uniquely situated within a context in which a subsample of the learners are incentivized to participate in the MOOC by their University. The study will examine how students who receive additional external incentives performed differently than the general population.

3. DATASET

We are analyzing an online course titled “The Preparation for Introductory Biology,” offered by professors from University of California, Irvine (UCI) and hosted in Coursera. The duration of the course was four weeks, and comprised of three units. Each unit was composed of short videos, three to four quizzes, and three to four peer assessments. These quizzes were in the multiple-choice format with automatic, immediate feedback. Additionally, learners were able to re-take each quiz with new questions up to three times in order to improve their scores. This course offered two study tracks: a Basic track, which was based on the performance of ten quizzes and resulted in a normal certificate, and a Scholars track, which included peer assessments and resulted in a distinguished certificate. Students did not pre-select a track, but rather were considered to have completed the Scholars track if they fulfilled the requirements.

The online course was created with a primary goal of preparing incoming freshman for the onsite Bio 93 course at UCI. The failure rate for this onsite course is 15%, which results in a proportion of undergraduates having to retake the course in the following term. At UCI, students must obtain a Mathematics SAT score of at least 550 (out of 800) to be eligible for the Biological Sciences major. Students who enter UCI with a score below 550 enter the Undeclared major and must instead pass a full year of biology and chemistry before being eligible to enroll as a Biological Sciences major. In this particular year, Undeclared major students were incentivized to enroll and complete the Preparation for Introductory Biology MOOC. If Undeclared major students successfully completed the MOOC with Distinction, they would be able to enter the major after just one term instead of waiting a year. This provides us with a unique opportunity to investigate the effects of providing an incentive to a group of learners enrolled in a MOOC.

The data mainly come from the SQL file exported from the Coursera database, which include time-stamped datasets of learners’ assignment (quiz and peer assessment) performance, scores, forum activities, and final performance. Another source of data come from UCI registrar, which records students who are enrolled in the onsite Bio 93 course. A reported 37,933 students signed up for the course and 551 students obtained Distinguish certificate for completing the Scholars track while 1,971 students obtained Normal certificate for completing the Basics track. Of the students who signed up, 35,411 students did not complete the course. 232 UCI Biology students and 172 Undeclared major students were identified to have signed up for the online course at the time of the study; we are still waiting to confirm a small proportion of these students.

4. DATA ANALYSIS

Our analysis consists of two logistic regression models, which predict student performance at the end of the course.

4.1 Feature Set

The predicted variable for the first model is the certificate the learner gets, i.e. a Distinction certificate or a Normal certificate. The predicted variable for the second model is whether the students get a Normal certificate or do not complete the MOOC, and thus receive no certificate.

The first predictor is the average quiz score learners obtained in the first week of the course. There are four quizzes in Unit 1 and the quiz score ranged from 0 to 6.

The second predictor is the number of peer assessments students completed in Week 1. We did not include the scores that students received from their peer assessors because those scores were not available until the second or the third week of the MOOC and therefore were not thought to affect MOOC persistence at week one.

The third predictor is learners’ social network degree in the first week, which measures the level of social integration. The social network degree measures the local centrality of learners in the online learning community. It is calculated as the number of edges to which the node is connected. In this scenario, we treat learners as nodes and making comments to another learner’s post is regarded as a directed edge from the commenter to the poster. The degree value is the number of connections that each learner has. Learners who did not participate in forums are assigned with 0 for their social network degree.

The fourth predictor is whether or not a learner is an incoming UCI Undeclared major student. This subgroup of students will go on to take the Bio 93 onsite course and have received external incentive to participate in the online course. We identified students as Undeclared by matching their school email addresses with Coursera accounts.

4.2 Logistic Regression

Two logistic regression models were run separately. The first logistic regression model predicts the type of certificate a learner received (Distinction or Normal certificates). The second logistic regression predicts whether students receive Normal certificates, or none at all. In the first model, the number of peer assessments taken in Week 1 is a strong predictor for achieving Distinction. For every unit increase in the number of peer assessments taken in Unit 1, the odds of getting Distinction are over 7 times larger than getting Normal, holding other predictors constant. Learners who are more active and well connected in the forum in the first week are more likely to receive Distinction than Normal certification, holding the number of peer assessments taken constant. For students taking the same number of peer assessments, the odds of UCI Undeclared major students getting Distinction are 89.4% higher than the rest of the learners. Table 1 indicates the odds ratio for the five predictors in the two logistic regression models.

Table 1 Odds Ratio

Odds Ratio	Class	
Variables	Distinction vs Normal	Normal vs None
Average Quiz Score	--	2.416***
Number of Peer Assessment	8.745**	1.054
Social Network Degree	1.192*	1.123
UCI Undeclared Major	1.894*	2.282**

Note. *** p<0.001 ** p<0.01 * p<0.05 .p< 0.1

In the second model, the average quiz scores in Unit 1 strongly predicted whether learners get Normal certificate or none. Learners' activity in the forum is no longer statistically significant in the predictive model. The odds of a UCI Undeclared major students getting Normal certificates are 2.282 times non-UCI-Undeclared major students, holding the average quiz score, the number of peer assessments completed and social network degree at fixed values.

Tenfold cross-validation was employed to estimate the predictive model. The first model predicting Distinction and Normal certificate earners achieved 92.6% accuracy. The second model predicting Normal certificate and no certification earners achieved 79.6% accuracy. Table 2 shows the evaluation of the predictive models.

Table 2 Model Evaluation

Evaluation	Modelling Distinction and Normal earners	Modelling Normal and None earners
Accuracy	0.926	0.796
ROC Area	0.947	0.851
Precision Positive	0.779	0.703
Precision Negative	0.978	0.911
Recall Positive	0.924	0.907
Recall Negative	0.927	0.713
Measure Positive	0.846	0.792
Measure Negative	0.952	0.800

5. DISCUSSION AND FUTURE WORK

The models indicate that assignment performance in Week 1 is a strong predictor of students' performance at the end of the course. The degree of social integration in the learning community in Week 1 is positively correlated with the achievement of Distinction certificates. Students with external incentive are more likely to complete the course compared to students in general, even in comparison with students who have similar backgrounds. However, this research is limited because we cannot control more variables that influence students' performance in the MOOC. Future research should focus on how to increase students' social integration and interaction in the online learning community, as these factors have been shown to influence student participation in MOOCs. More investigation into providing external incentive and increasing course relevance for the target audience is still needed. It is also worth exploring the effectiveness of the integration of online education and traditional face-to-face education in more depth.

Additionally, more experimentation and research into the relationship between quality of online courses and students' engagement and performance is recommended. Research from disciplines such as Education, Human Computer Interaction, and Computer Science should collaborate and redesign online education. The low engagement and completion rates reflect the existent opportunities for the improvement of online education.

6. ACKNOWLEDGMENTS

We are very indebted to the Digital Learning Lab, University of California, Irvine.

7. REFERENCES

1. Balakrishnan, G. and Coetzee, D. 2013.Predicting Student Retention in Massive Open Online Courses using Hidden Markov Models. http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-109.pdf.
2. Greeno, J.G.1989.A perspective on thinking. *American Psychologist 44*, 2 (1989), 134–141.
3. Ho, A.D., Reich, J., Nesterko, S.O., et al. 2014. *HarvardX and MITx: The First Year of Open Online Courses, Fall 2012-Summer 2013*. Social Science Research Network, Rochester, NY.
4. Kizilcec, R.F., Piech, C., and Schneider, E. 2013. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, ACM (2013), 170–179.
5. Lave, J. and Wenger, E. 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press.
6. Parr, Chris. 2013. New study of low MOOC completion rates | Inside Higher Ed. *Inside Higher Ed*. http://www.insidehighered.com/news/2013/05/10/new-study-low-mooc-completion-rates?utm_source=feedly.
7. Yang, D., Sinha, T., Adamson, D., and Rose, C.P.2013. Anticipating student dropouts in Massive Open Online Courses.
8. Hill, P. 2013. Emerging Student Patterns in MOOCs: A Graphical View. *e-Literate*. http://mfeldstein.com/emerging_student_patterns_in_moocs_graphical_view/.

Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software

Maria Ofelia Z. San Pedro[1], Jaclyn L. Ocumpaugh[1], Ryan S. Baker[1], Neil T. Heffernan[2]
[1]Teachers College Columbia University, 525 W 120th St. New York, NY 10027
[2]Worcester Polytechnic Institute, 100 Institute Rd. Worcester, MA 01609

mzs2106@tc.columbia.edu, ocumpaugh@tc.columbia.edu, baker2@exchange.tc.columbia.edu, nth@wpi.edu

ABSTRACT

The worldwide increase in demand for qualified workers in science, technology, engineering, and mathematics (STEM) fields has resulted in a greater focus on preparing students to enroll in postsecondary STEM programs. The processes that lead students to become interested in and equip them for STEM careers begin years earlier. Previous research indicates that family background, financial resources, and prior family academic achievement can be used to predict whether a student will enroll in a STEM major. In this paper, we consider another class of factors that may be predictive while being more actionable. In this paper, we use predictive analytics, based on previously-validated automated detectors of student learning and engagement, to predict which students will choose a STEM major. With data from 363 college students who used ASSISTments during their regular middle school math classes, we develop a model that can successfully distinguish 66% of the time if a student will choose a STEM major or a non-STEM major when they enter college. In doing so, we offer steps towards providing educators with more actionable information on the STEM trajectories of individual students.

Keywords

STEM, Affect Detection, Knowledge Modeling, Educational Data Mining, Predictive Analytics, Gaming the System

1. INTRODUCTION

Science, technology, engineering, and mathematics (STEM) jobs have played a significant role in driving the modern economy, with growth as high as three times faster than that of non-STEM jobs in the United States over the last decade [13]. Many STEM jobs require a postsecondary degree or other advanced technical training. However, research shows a gap between the number of students who express interest in STEM degree programs and the number who actually enter them, which is driven by inadequate preparation for higher level STEM skills and other aspects of college readiness [21]. This lack of preparation often begins as early as middle school. For instance, the National Mathematics Advisory Panel argues that difficulties with concepts like fractions hinder students from further achievement in mathematics, including algebra [15].

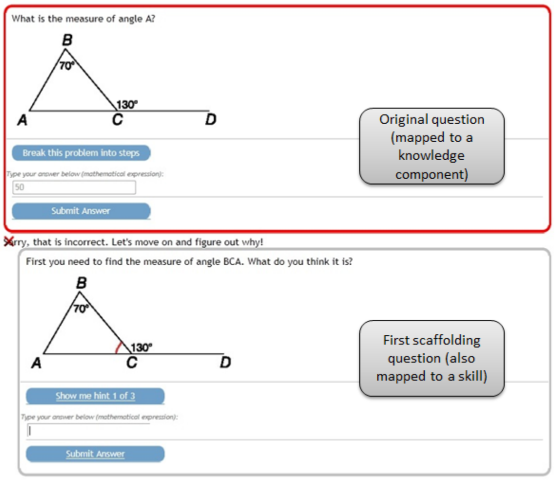
Since the motivation and interest that guides students to enter STEM careers can often be traced to middle school [12], it may be valuable to work on creating better understanding of the factors and processes in middle school students' learning and engagement that connect to eventual decisions to pursue STEM degrees and careers. Studies show that family background, financial resources, and prior family academic achievement have a significant impact to a student's interest and intention to major in STEM [21]. However, current predictive models are generally insufficient to help classroom teachers identify which students are on track, which need further support, and what types of interventions are likely to have the greatest impact [14]. Part of the challenge is one of getting the right data – predictive models have typically relied on course-level data like grades [4] or high-level indicators of general student interest in STEM careers [12, 21], making it difficult to make predictions actionable before a student is already significantly off-track.

Recent work, however, has taken advantage of the increasing deployment of educational software that logs student behavior (often in fine-grained detail), developing automated detectors that assess student learning and engagement [3, 8, 10, 18]. This development creates new opportunities to assess students on a broader range of constructs than previously possible and to predict long-term student outcomes, such as college enrollment several years after using a learning system [20]. This study builds on these models, finding that enrollment in STEM degree programs (among those in college) can be inferred from learning and engagement during middle school mathematics learning. Using previously developed automated detectors of knowledge, affect, and disengaged behavior, we develop a prediction model to distinguish whether or not students who attend college will enroll in a STEM major. By identifying these constructs, we argue, we can better identify which students are most in need of interventions, helping educators to better serve their students.

2. METHODOLOGY

2.1 The ASSISTments System

This study predicts student outcomes from their interactions with the ASSISTments system [19], a free web-based mathematics tutoring system for middle-school mathematics, provided by Worcester Polytechnic Institute (WPI). ASSISTments *assesses* a student's knowledge while *assisting* them in learning, providing teachers with formative assessment of students as they acquire specific knowledge components. Within the system, each mathematics problem maps to one or more cognitive skills. When students answer correctly, they proceed to the next problem. When they answer incorrectly, the system scaffolds instruction by dividing the problem into component parts, stepping students through each before returning them to the original problem (see Figure 1). Once the original problem is correctly answered, the student advances to the next.



What is the measure of angle A?

Type your answer below (mathematical expression):

50

✗rry, that is incorrect. Let's move on and figure out why!

First you need to find the measure of angle BCA. What do you think it is?

Type your answer below (mathematical expression):

B
70°
130°
A
C
D
Break this problem into steps
Submit Answer
Original question (mapped to a knowledge component)
B
70°
130°
A
C
D
Show me hint 1 of 3
Submit Answer
First scaffolding question (also mapped to a skill)

Figure 1. Example of an ASSISTments problem.

2.2 Data

2.2.1 Post-High School Survey Data

For this study, over 2,500 students who had used ASSISTments during their middle school mathematics classes were invited to participate in a survey about their post-high school academic and career achievements. As of this writing, a total of 425 students have responded, for a retention rate of about 20%. Students were drawn from school districts in the Northeastern US who used ASSISTments during the 2004-2005 to 2006-2007 school years (with a few continuing usage for more than one year). Of the 3 districts targeted for this research, 1 was a low-performing district in an urban area, primarily serving English language learners eligible for free or reduced-price lunches; the other 2 districts drew from suburban middle-class populations.

The survey asked students to specify what degree program(s) they were enrolled in, whether they were engaged in full or part-time employment, and what their current employment was. Out of the 425 respondents, 363 were enrolled in college and will be analyzed within this paper. Student answers were coded to reflect their enrollment or participation in a STEM major as defined by the National Science Foundation [16].

2.2.2 ASSISTments Data

Action log files from ASSISTments were obtained for each of the 363 respondents, generating a total of 329,565 actions within the system, across a total of 166,188 original and scaffolding problems. (Actions include answering questions or requesting help.) On average, this resulted in 457 problems per student. Knowledge, affect, and behavior models were applied to this dataset, generating features used for our predictive model of STEM major enrollment.

2.3 Feature Distillation

The features used to predict college major classifications (STEM vs. non-STEM) were generated using a discovery with models approach, leveraging automated detectors of student engagement and learning that were previously developed and validated for ASSISTments. These included existing models of student knowledge, disengaged behaviors (carelessness, gaming the system, and off-task behavior), educationally-relevant affective states (boredom, engaged concentration, confusion, frustration), and other information about student usage (the proportion of correct actions and the total number of actions made by the student, a proxy for overall usage).

Corbett and Anderson's [9] Bayesian Knowledge Tracing (BKT) model, a proven knowledge-estimation model used in a number of ITS systems, was applied to the data for this study by employing a brute-force grid search. BKT infers students' latent knowledge from their performance on previous problems involving the same set of skills. Each time a student attempts a problem or problem step for the first time, BKT recalculates the estimates of that student's knowledge for the skill (or knowledge component) involved in that problem. BKT estimates were calculated at the student's first response to each problem and were applied to each of the student's subsequent attempts on that problem.

To obtain assessments of affect and disengaged behaviors, we leverage existing detectors of student affect and behavior within the ASSISTments system [17, 18]. These included boredom, engaged concentration, confusion, frustration off-task behavior, gaming the system, and carelessness. Data from students who attended urban schools were labeled using affect models optimized for students in urban schools [17, 18], and data from students who attended suburban schools were labeled using affect models optimized for students in suburban schools [17].

Except for carelessness (explained below), the affect and behavior detectors were developed in a two-stage process. First, student affect labels were acquired from BROMP field observations, which records them using HART, an Android app (reported in [18]). Then those labels were synchronized with the log files generated by ASSISTments. This process resulted in automated detectors that can be applied to log files at scale, specifically the data set used in this project (action log files for the 363 students). The detectors were constructed using only log data from student actions within the software occurring concurrently or prior to each BROMP observation, achieving state-of-the-art model goodness [17, 18], and were applied to the data set used in this paper to produce confidence values for each construct for each student actions. Detector confidences were rescaled in order to correct for bias caused by resampling during training [18, 20].

Carelessness is operationalized using contextual slip estimates—the probability that despite knowing the skill to answer an item, a specific incorrect action made by the student for that item is the result of slip or carelessness (see [2]). The probability of carelessness/slip is assessed contextually and is different depending on the context of the student error. As such, the estimate of probability of carelessness/slip is different for each student action. This study uses carelessness models that were previously constructed for ASSISTments [18].

2.4 Modeling STEM Major Enrollment

Within this paper, we develop a logistic regression model predicting STEM major enrollment from combinations of features. Using logistic regression allows for relatively good interpretability of the resultant model, while matching the statistical approach used in much of the work predicting long-term transitions from K-12 education to college [3, 6, 11, 20].

For each of the assessments (learning, affect, and disengaged behaviors), aggregate student-level predictor variables were created by taking the average of the predictor feature values for each student. (In other words, taking the average boredom per student, average confusion per student, etc.) A simple backward elimination feature selection, based on each parameter's statistical significance was used. All predictor variables were standardized using z-scores to increase interpretability of the resulting odds ratios. (Note that this does not impact model goodness or predictive power in any fashion.)

3. RESULTS

First, we looked at our original, non-standardized features and how their values compare between those who were pursuing a STEM major in college and those who were not (Table 1).

Table 1. Feature comparison for STEM Major students (1, n=194) and Non-STEM Major (0, n=169).

	STEM Major	Mean	Std. Dev.	t-value	Cohen's d
Carelessness	0	0.204	0.118	-4.437	0.460
	1	**0.267**	**0.154**	(p < 0.01)	
Student Knowledge	0	0.340	0.196	-4.853	0.508
	1	**0.447**	**0.223**	(p < 0.01)	
Correctness	0	0.418	0.171	-5.184	0.547
	1	**0.508**	**0.161**	(p < 0.01)	
Boredom	0	0.222	0.072	0.286	0.030
	1	0.219	0.078	(p = 0.78)	
Engaged Concentration	0	0.660	0.064	1.500	0.162
	1	0.652	0.044	(p = 0.14)	
Confusion	0	0.085	0.058	0.636	0.067
	1	0.081	0.062	(p = 0.53)	
Frustration	0	0.171	0.078	1.602	0.166
	1	0.155	0.101	(p = 0.11)	
Off-Task	0	0.206	0.086	-0.709	0.076
	1	0.212	0.062	(p = 0.48)	
Gaming	**0**	**0.181**	**0.174**	5.269	0.573
	1	0.100	0.108	(p < 0.01)	
Number of Actions	**0**	**1049.5**	**1569.2**	1.984	0.218
	1	784.53	794.65	(p < 0.05)	

An independent samples t-test (Table1) shows that students in STEM majors had higher mean values for average student knowledge, average carelessness, and average correctness, while students in non-STEM majors had higher mean values for average gaming and average number of actions. Effect sizes for these features were computed using Cohen's d which measures the standardized mean difference of the features between two groups – in this paper, the students pursuing a STEM major and those taking a non-STEM major. As shown in Table 1, gaming the system has the largest effect size (d=0.573), indicating that students who took a non-STEM major had a mean gaming percentage 0.573 standard deviations higher during middle school than students who took a STEM major. It is worth noting that the effect size of gaming for predicting STEM major is substantially larger than the effect size of gaming for predicting whether students attended college or not, where d was 0.293 [43].

These observations align with the individual effects of each feature on the prediction of STEM major enrollment. For example, there is a strong positive relationship between enrolling in a STEM major and average correct answers, indicating that success in mathematics using ASSISTments is associated to higher probability of pursuing a STEM major. The same strong positive relationship is seen between STEM major enrollment and student knowledge estimate as the student learns with ASSISTments. Two non-intuitive results are found in these data. The first concerns the relationship between carelessness and STEM major enrollment. Taken by itself, the more a student becomes careless, the more likely the student is to choose a STEM major, evidence in keeping with past results that careless errors are characteristic of more successful students [7]. The second non-intuitive result concerns the amount of interaction the student has had with the system. Our results show that the number of actions per student is negatively related to majoring in a STEM program, perhaps indicative of struggling students whose actions consist mostly of help requests and scaffolded attempts (which indicate that the student got many problems wrong on the first try). Additionally, the more a student games the system, the less likely that student is to enroll in a STEM major – a result compatible with past evidence that gaming is associated with poorer learning in mathematics [8].

A model for STEM Major enrollment including a combination of data features was developed using Logistic Regression and cross-validated at the student level (6-fold). Our final model (Table 2) achieves a cross-validated A' of 0.663 and a cross-validated Kappa of 0.257. This model is statistically significantly better than the null model, χ^2 (df = 2, N = 363) = 38.010, p < 0.001 and achieved a fit of R^2 (Cox and Snell) = 0.099, R^2 (Nagelkerke) = 0.133, indicating that its predictors explain 9.9-13.3% of the variance of those who attended college. As seen in Table 2, the predictors (student knowledge and gaming) maintained the same directionality as they demonstrated individually (Table 1).

Table 2. Model of STEM major enrollment

Features	Coefficient	Chi-Square	p-value	Odds Ratio
Student Knowledge	0.357	8.859	0.003	1.429
Gaming	-0.492	13.792	< 0.001	0.611
Intercept	0.133	1.418	0.234	1.142

4. DISCUSSION AND CONCLUSION

This paper presents a logistic regression model which indicates that a combination of features of student engagement and student success in ASSISTments can distinguish a student who will take a STEM major 66.3% of the time. Success within middle school mathematics (indicated by correct answers and high probability of knowledge in ASSISTments) is positively associated with STEM major enrollment, a finding aligned with studies that conceptualize high performance and developing aptitude during early schooling as a sign of STEM major readiness and predictor of later enrollment in STEM programs [21]. The disengaged behavior of gaming the system during middle school mathematics is found to be negatively associated with pursuing a STEM degree. Previous research has shown that that gaming negatively impacts learning [8], but it is also a particularly strong indicator of disengagement with mathematics, suggesting that students' lack of interest in STEM careers may manifest early. It has been shown that gaming behaviors can be successfully remediated either through alternate opportunities to learn the material that students bypassed or through metacognitive interventions which explain why gaming is ineffective for learning [1, 8]. The relationship between gaming and the choice of college major is relatively large, larger than its relationship to whether a student attends college [20], suggesting that gaming remediation could be an important component of efforts to encourage more students towards STEM degree programs.

Our model also finds that affective states are not particularly strong predictors of whether a student will pursue a STEM major, in contrast to work which found that affective states were predictive of college attendance [20]. A possible explanation is that student affect may be less relevant for college major choice than how students respond to that affect (e.g. a student who just becomes careless when he or she gets bored might be more likely to maintain the STEM track than a student who games the system in response to his or her boredom). It also may be that affect during schooling largely plays a role in determining whether students choose higher education at all; once we analyze only the students who choose higher education (e.g. the current sample), affect plays a much smaller role than domain-specific learning or choices. Negative affective states should still be attended to, as they impact both learning outcomes and college attendance [18, 20]. It may be a valuable area of future work to explore whether the interactions of affective states with other factors can influence these predictions. For example, gaming the system and carelessness may be mediating some of the relationships between affect and college major selection.

One possible use of these findings is to give educators and career counselors a new lens on early indicators of disinterest or disengagement from STEM content and instruction, allowing them to develop counseling strategies that will sustain student interest in pursuing STEM degrees and careers. In doing so, it is important to note that despite considerable current societal emphasis on encouraging students to pursue STEM majors, some students will have other interests and goals. At the same time, the demand for STEM professionals considerably outstrips supply and there is value in a citizenry that has basic STEM literacy [5], regardless of their career choices. If these predictions, based on interactions within a mathematics tutor, can be used to provide targeted help to students that builds on their strengths and strengthens their weaknesses, we stand the chance of both identifying students who are particularly gifted in mathematics and creating greater options for students who struggle for one reason or another. As online learning spreads to other domains of K-12 education, we will be able to provide similar support within other subject domains, supporting all students in reaching their maximum potential.

5. ACKNOWLEDGMENTS

This research was supported by the NSF (#DRL-1031398, #SBE-0836012) and the Bill & Melinda Gates Foundation (#OPP1048577). We also thank Kevin Ramirez for his support in data collection and Irina Shklovski for her advice.

6. REFERENCES

[1] Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Mehranian, H., Fisher, D., Barto, A., Mahadevan, S. and Woolf, B. 2007. Repairing disengagement with non-invasive interventions. In *Proc. AIED*, 195–202.

[2] Baker, R.S.J.d., Corbett, A.T., and Aleven, V. 2008. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In *Proceedings of the 9th ITS Conference*, 406-415.

[3] Baker, R.S., Corbett, A.T., and Koedinger, K.R. 2004. Detecting Student Misuse of Intelligent Tutoring Systems. In *Proceedings of the 7th ITS Conference*, 531-540.

[4] Bowers, A.J., Sprott, R., and Taff, S.A. 2013. Do we Know Who Will Drop Out? A Review of the Predictors of Dropping out of High School: Precision, Sensitivity and Specificity. *The High School Journal*. 96(2), 77-100.

[5] Bybee, R. W. (1997). *Achieving scientific literacy: From purposes to practices*(Vol. 88). Portsmouth, NH: Heinemann

[6] Cabrera, A. F. 1994. Logistic regression analysis in higher education: An applied perspective. *Higher Education: Handbook of Theory and Research*, 10, 225-256.

[7] Clements, K. 1982. Careless errors made by sixth-grade children on written mathematical tasks. *Journal for Research in Mathematics Education*, 13, 136–144.

[8] Cocea, M., Hershkovitz, A., and Baker, R.S.J.d. 2009. The Impact of Off-task and Gaming Behaviors on Learning: Immediate or Aggregate? In *Proc. AIED*, 507-514

[9] Corbett, A.T., and Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 4, 253-278.

[10] D'Mello, S.K., Craig, S.D., Witherspoon, A. W., McDaniel, B. T., and Graesser, A. C. 2008. Automatic Detection of Learner's Affect from Conversational Cues. *User Modeling and User-Adapted Interaction*, 18 (1-2), 45-80.

[11] Eccles, J. S., Vida, M. N., and Barber, B. 2004. The relation of early adolescents' college plans and both academic ability and task-value beliefs to subsequent college enrollment. *Journal of Early Adolescence*, 24, 63-77.

[12] Hayden, K., Ouyang, Y, Scinski, L., Olszewski, B., and Bielefeldt, T. 2011. Increasing student interest and attitudes in STEM: Professional development and activities to engage and inspire learners. *Contemporary Issues in Technology and Teacher Education*, 11(1), 47-69.

[13] Langdon, D., McKittrick, G., Beede, D., Khan, B., and Doms, M. 2011. STEM: Good jobs now and for the future. *ESA Issue Brief# 03-11*. US Department of Commerce.

[14] Lent, R. W., Lopez Jr, A. M., Lopez, F. G., and Sheu, H. B. 2008. Social cognitive career theory and the prediction of interests and choice goals in the computing disciplines. *Journal of Vocational Behavior*, 73(1), 52-62.

[15] National Mathematics Advisory Panel. 2008. *Foundations for success: The final report of the National Mathematics Advisory Panel*. US Department of Education.

[16] National Science Foundation, National Center for Science and Engineering Statistics. 2013. Science and Engineering Degrees: 1966–2010. *Detailed Statistical Tables NSF* 13-327. Arlington, VA.

[17] Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., Heffernan, C. (in press) Population validity for Educational Data Mining models: A case study in affect detection. To appear in the *British Journal of Educational Technology*.

[18] Pardos, Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M. 2013. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Proceedings of the 3rd LAK Conference*, 117-124

[19] Razzaq, L. M., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K. R., Junker, B., ... & Rasmussen, K. P. (2005, May). Blending Assessment and Instructional Assisting. In *AIED* (pp. 555-562).

[20] San Pedro, M.O.Z., Baker, R.S.J.d., Bowers, A.J., Heffernan, N.T. 2013. Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. In *Proceedings of the 6th EDM Conference*, 177-184.

[21] Wang, X. 2013. Why Students Choose STEM Majors Motivation, High School Learning, and Postsecondary Context of Support. *American Educational Research Journal*, 50(5), 1081-1121.

Quantized Matrix Completion for Personalized Learning

Andrew S. Lan
Rice University
mr.lan@sparfa.com

Christoph Studer
Cornell University
studer@sparfa.com

Richard G. Baraniuk
Rice University
richb@sparfa.com

ABSTRACT

The recently proposed SPARse Factor Analysis (SPARFA) framework for personalized learning performs *factor analysis* on ordinal or binary-valued (e.g., correct/incorrect) graded learner responses to questions. The underlying factors are termed "concepts" (or knowledge components) and are used for *learning analytics (LA)*, the estimation of learner concept-knowledge profiles, and for *content analytics (CA)*, the estimation of question–concept associations and question difficulties. While SPARFA is a powerful tool for LA and CA, it requires a number of algorithm parameters (including the number of concepts), which are difficult to determine in practice. In this paper, we propose *SPARFA-Lite*, a convex optimization-based method for LA that builds on matrix completion, which only requires a *single* algorithm parameter and enables us to automatically identify the required number of concepts. Using a variety of educational datasets, we demonstrate that SPARFA-Lite (i) achieves comparable performance in predicting unobserved learner responses to existing methods, including item response theory (IRT) and SPARFA, and (ii) is computationally more efficient.

Keywords

Personalized learning, learning analytics, content analytics, factor analysis, matrix completion, convex optimization.

1. INTRODUCTION

Recent advances in machine learning propel the design of personalized learning systems (PLSs) that mine learner data (e.g., graded responses to tests or homework assignments) to automatically provide timely feedback to individual learners. Such automated systems have the potential to revolutionize education by delivering a high-quality, personalized learning experience at large scale.

1.1 SPARse Factor Analysis (SPARFA)

The recently proposed SPARse Factor Analysis (SPARFA) framework introduces models and machine learning algorithms for learning and content analytics [16, 17]. *Learning analytics* (LA) stands for the analysis of the knowledge of each learner, while *content analytics* (CA) stands for the analysis of all learning resources, i.e., textbooks, lecture videos, questions, etc. SPARFA analyzes binary-valued (1 for a correct answer and 0 for an incorrect one) or quantized (ordinal-valued, e.g., partial credits) graded responses of N learners to Q questions, in the domain of a course/exam. The key assumption of SPARFA is that the learners' responses to questions are governed by a small number of K ($K \ll N, Q$) latent factors, called "concepts," which are also known as knowledge components [11]. SPARFA performs the joint estimation of (i) question–concept associations, (ii) learner concept knowledge profiles, and (iii) question difficulties, solely from binary-valued graded learner responses. Provided this analysis, SPARFA enables a PLS to provide automated feedback to learners on their individual concept knowledge and to course instructors on the content organization of the analyzed course.

SPARFA, as well as other factor analysis methods, inevitably suffer from the lack of a principled and computationally efficient way to select the appropriate values of the algorithms' parameters, especially the number of latent concepts K. The choice of the number of concepts K is important for two reasons: First, it affects the performance in predicting unobserved learner responses. Second, it determines the interpretability of the estimated concepts, which is key for a PLS to provide human-interpretable feedback to learners. Rule-based intelligent tutoring systems [24] rely on domain experts to manually pre-define the value of K. Such an approach turns out to be labor-intensive and is error prone, which prevents its use for applications in massive open online courses (MOOCs) [20]. SPARFA utilizes cross-validation to select K, as well as all other algorithm parameters [17]. Such an approach is computationally extensive as it requires multiple SPARFA runs to identify appropriate values for all algorithm parameters.

1.2 Contributions

In this work, we propose SPARFA-Lite, a convex optimization-based LA algorithm that automatically selects the number of latent concepts K by analyzing graded learner responses in the domain of a single course/assessment. SPARFA-Lite leverages recent results in quantized matrix completion [15] to analyze quantized graded learner responses, which accounts for the fact that responses are often graded on an ordinal scale (partial credit). Since SPARFA-Lite only has a single algorithm parameter, it has low computational complexity as compared to existing methods such as IRT or conventional SPARFA. We demonstrate the effectiveness of SPARFA-Lite in (i) predicting unobserved learner responses and (ii) performing LA on a variety of real-world educational datasets.

1.3 Related work

Factor analysis has been used extensively to analyze graded learner response data [19, 23]. While some factor analysis methods treat binary-valued graded learner responses as real numbers [2, 13], others use probabilistic models to achieve superior performance in predicting unobserved learner responses. These methods include the additive factor model (AFM) [9], instructional factors analysis (IFA) [10], and learning factor analysis (LFA) [22], which all assume that the number of concepts K to be known a priori. Collaborative filtering IRT (CF-IRT) [5] and SPARFA [17] both use cross-validation to select K, as well as all other tuning parameters, by identifying the best prediction performance on unobserved learner responses. This approach is computationally extensive and does not scale to MOOC scale applications, where the dimension of the problem is large and immediate feedback is required. The authors of [4] proposed to select K by applying an SVD to the binary-valued graded learner response matrix and examining the decay of its singular values, which is not an automated approach.

Matrix completion (MC) aims to recover a low-rank matrix from incomplete, real-valued observations [6, 8], and has been used extensively in collaborative filtering applications. More recently, 1-bit MC [12], and its generalization, quantized MC [15] have been proposed for the recovery of low-rank matrices from incomplete binary-valued and quantized (or ordinal) observations, respectively. Since the graded learner responses in educational scenarios are typically binary-valued or ordinal, we next investigate the applicability of quantized MC [15] to educational scenarios.

2. SPARFA-LITE STATISTICAL MODEL

SPARFA-Lite aims at recovering the unknown, low-rank matrix $\mathbf{Z}$ that governs the learners' responses to questions, solely from quantized (ordinal) graded learner responses. Suppose that we have N learners answering Q questions. Let the $Q \times N$ matrix $\mathbf{Z}$ be the underlying low-rank matrix that we seek to recover. Let $Y_{i,j} \in \mathcal{O}$ denote the quantized observed graded response of the j^{th} learner, with $j \in \{1, \ldots, N\}$, to the i^{th} question, with $i \in \{1, \ldots, Q\}$. $\mathcal{O} = \{1, \ldots, P\}$ is a set of P ordered labels. Inspired by [15], we use the following model for the graded learner response $Y_{i,j}$:

$$\begin{aligned} &Y_{i,j} = \mathcal{Q}(Z_{i,j} + \epsilon_{i,j}),\ (i,j) \in \Omega_{\text{obs}}, \\ &\epsilon_{i,j} \sim \textit{Logistic}(0,1)\,. \end{aligned} \tag{1}$$

Here, $\textit{Logistic}(0,1)$ represents the Logistic distribution with zero mean and unit scale [14]. The set $\Omega_{\text{obs}} \subseteq \{1, \ldots, Q\} \times \{1, \ldots, N\}$ contains the indices associated to the observed learner responses $Y_{i,j}$. The function $\mathcal{Q}(\cdot) : \mathbb{R} \to \mathcal{O}$ represents a scalar quantizer, defined as

$$\mathcal{Q}(x) = p \quad \text{if } \omega_{p-1} < x \leq \omega_p,\ p \in \mathcal{O},$$

where $\{\omega_0, \ldots, \omega_P\}$ is a set of quantization bin boundaries, with $\omega_0 \leq \omega_1 \leq \cdots \leq \omega_{P-1} \leq \omega_P$. We will assume that the set of quantization bin boundaries $\{\omega_0, \ldots, \omega_P\}$ is known a priori. In situations where these bin boundaries are unknown, they can be estimated directly from data (see, e.g., [15, 16] for the details).

In terms of the likelihood of the observed graded learner responses $Y_{i,j}$, the model in (1) can be written equivalently as

$$p(Y_{i,j} = p \mid Z_{i,j}) = \Phi(\omega_p - Z_{i,j}) - \Phi(\omega_{p-1} - Z_{i,j})\,, \tag{2}$$

where $\Phi(x) = \frac{1}{1+e^{-x}}$ corresponds to the inverse logit link function. For this paper, we will be using only the inverse logit link function as it leads to algorithms with lower computational complexity comparing to the inverse probit link function [15].

The goal of the SPARFA-Lite algorithm detailed next is to recover the unknown low-rank matrix $\mathbf{Z}$ given the observed binary-valued graded learner responses $Y_{i,j}$, $(i,j) \in \Omega_{\text{obs}}$.

3. THE SPARFA-LITE ALGORITHM

To recover the low-rank matrix $\mathbf{Z}$ from binary-valued graded learner responses, we minimize the negative log-likelihood of the observed graded learner responses $Y_{i,j}$, $(i,j) \in \Omega_{\text{obs}}$, subject to a low-rank promoting constraint on $\mathbf{Z}$. In particular, we seek to solve the following convex optimization problem:

$$(\text{P}) \begin{cases} \underset{\mathbf{Z} \in \mathbb{R}^{Q \times N}}{\text{minimize}} & f(\mathbf{Z}) = -\sum_{i,j:(i,j) \in \Omega_{\text{obs}}} \log p(Y_{i,j} | Z_{i,j}) \\ \text{subject to} & \|\mathbf{Z}\| \leq \lambda. \end{cases}$$

Here, the constraint $\|\mathbf{Z}\| \leq \lambda$ is used to promote low-rank solutions $\mathbf{Z}$ [8] and the parameter $\lambda > 0$ is used to control its rank. In practice, one can use the nuclear norm constraint $\|\mathbf{Z}\|_* \leq \lambda$, which is a convex relaxation of the (non-convex) low-rank constraint $\operatorname{rank}(\mathbf{Z}) \leq r$ [6, 8]; alternatively, one can use the max-norm constraint $\|\mathbf{Z}\|_{\max} \leq \lambda$ (see [18] for the details). We select the only algorithm parameter $\lambda > 0$ via cross-validation. We emphasize that this parameter-selection process of SPARFA-Lite is much more efficient than that for regular SPARFA, which has three algorithm parameters.

Since the gradient of the negative log-likelihood of the inverse logit link function can be computed efficiently, (P) can be solved efficiently via the FISTA framework [3]. Starting with an initialization of the matrix $\mathbf{Z}$, at each inner iteration $\ell = 1, 2, \ldots$, the algorithm performs a gradient step that aims at reducing the objective function $f(\mathbf{Z})$, followed by a projection step that makes the solution satisfy the constraint $\|\mathbf{Z}\| \leq \lambda$. Both steps are repeated until convergence.

The *gradient step* is given by $\widehat{\mathbf{Z}}^{\ell+1} \leftarrow \mathbf{Z}^{\ell} - s_\ell \nabla f$, where s_ℓ is the step-size at iteration ℓ (see [15] for the details on step-size selection). The gradient of the objective function $f(\mathbf{Z})$ with respect to $\mathbf{Z}$ is given by

$$[\nabla f]_{i,j} = \begin{cases} \frac{\Phi'(L_{i,j} - Z_{i,j}) - \Phi'(U_{i,j} - Z_{i,j})}{\Phi(U_{i,j} - Z_{i,j}) - \Phi(L_{i,j} - Z_{i,j})} & \text{if } (i,j) \in \Omega_{\text{obs}} \\ 0 & \text{otherwise,} \end{cases}$$

where the derivative of the inverse logit link function corresponds to $\Phi'(x) = \frac{1}{2+e^{-x}+e^{x}}$. The $Q \times N$ matrices $\mathbf{U}$ and $\mathbf{L}$ contain the upper and lower bin boundaries corresponding to the measurements $Y_{i,j}$, i.e., we have $U_{i,j} = \omega_{Y_{i,j}}$ and $L_{i,j} = \omega_{Y_{i,j}-1}$.

The *projection step* imposes low-rankness on $\mathbf{Z}$. For the nuclear norm constraint case $\|\mathbf{Z}\|_* \leq \lambda$, this step requires a projection onto the nuclear norm ball with radius λ, which can be performed by first computing the SVD of $\mathbf{Z}$ followed by projecting the vector of singular values onto an ℓ_1-norm ball with radius λ (the details can be found in [6]). The resulting projection step corresponds to

$$\mathbf{Z}^{\ell+1} \leftarrow \widetilde{\mathbf{U}}\text{diag}(\mathbf{s})\widetilde{\mathbf{V}}^T, \ \text{ with } \ \mathbf{s} = \text{P}_\lambda(\text{diag}(\mathbf{S})), \tag{3}$$

where $\widetilde{\mathbf{U}}\mathbf{S}\widetilde{\mathbf{V}}^T$ denotes the SVD of $\widehat{\mathbf{Z}}^{\ell+1}$. The operator $\text{P}_\lambda(\cdot)$ denotes the projection of a vector onto the ℓ_1-norm ball with radius λ, which can be computed at low complexity [15]. For the max-norm constraint $\|\mathbf{Z}\|_{\max} \leq \lambda$, the projection step can be calculated efficiently by following the method put forward in [18]. We emphasize that SPARFA-Lite is guaranteed to converge to a global optimum, since the problem (P) is convex.

4. SPARFA-LITE LEARNING ANALYTICS

We now demonstrate how SPARFA-Lite can be used to perform LA. To this end, we assume that there is tag information available for each question, i.e., there are a set of M user-defined labels (tags) associated with the Q questions, with each question associated with at least one tag. We define the $Q \times M$ question-tag matrix $\mathbf{T}$ with $T_{i,m} = 1$ if tag m is associated to question i, and $T_{i,m} = 0$ otherwise. We also define the $Q \times N$ matrix $\mathbf{A}$ with $A_{i,j} = \Phi(Z_{i,j}) \in [0,1]$, which is the de-noised and completed version of the (partially observed) graded learner response matrix $\mathbf{Y}$. Using both matrices $\mathbf{T}$ and $\mathbf{A}$, we can compute the $N \times M$ learner tag knowledge matrix $\mathbf{B}$ with the entries $B_{j,m} = (\sum_{i=1}^{Q} T_{i,m})^{-1} \widetilde{B}_{j,m} \in [0,1]$, where $\widetilde{\mathbf{B}} = \mathbf{A}^T\mathbf{T}$. The entries $B_{j,m}$ represent the de-noised concept knowledge of learner j on tag m; large values represent good knowledge of tag m, whereas small values represent poor tag knowledge. This tag knowledge information is crucial for a PLS to perform LA.

5. EXPERIMENTS

We now compare SPARFA-Lite against existing factor analysis methods for predicting unobserved learner responses, using real-world educational datasets and demonstrate its efficacy in performing LA. All algorithm parameters are selected through cross-validation. All results are averaged over 25 independent Monte–Carlo trials.

5.1 Predicting unobserved learner responses

We first compare the performance of SPARFA-Lite in predicting unobserved graded learner responses with two state-of-the-art factor analysis algorithms.

Datasets. In this experiment, we use five different educational datasets for: (1) an undergraduate course on fundamentals of electrical engineering, consisting of $N = 92$ learners answering $Q = 203$ questions, with 99.5% of the answers observed; (2) an undergraduate course on signals and systems, consisting of $N = 41$ learners answering $Q = 143$ questions, with 97.1% of the answers observed; (3) an undergraduate course on introduction to probability and statistics, consisting of $N = 57$ learners answering $Q = 107$ questions, with 68.9% of the answers observed; (4) a university entrance exam, consisting of $N = 1706$ learners answering $Q = 60$ questions, with 60.9% of the answers observed; and (5) another university entrance exam, consisting of $N = 1564$ learners answering $Q = 60$ questions, with 70.8% of the answers observed. The undergraduate course datasets are collected via OpenStax Tutor [21]; see [25] for the details on the university entrance exam dataset. Note that all of these datasets contain binary-valued graded learner responses, which is a special case of the general, quantized model proposed above (with $P = 2$ and $\{\omega_0, \omega_1, \omega_2\} = \{-\infty, 0, \infty\}$). For simplicity, we will refer to the individual datasets as Dataset 1-to-5, respectively.

Experimental setup. We now compare SPARFA-Lite against CF-IRT [5] and SPARFA [17], two established factor analysis methods that perform well in terms of predicting unobserved graded learner responses. To assess prediction performance on unobserved learner responses, we randomly puncture each dataset by removing 20% of the observed learner responses in $\mathbf{Y}$ to form a test set. We then train all three algorithms on the rest of the observed learner responses and predict the unobserved responses in the test set. Since CF-IRT and SPARFA both have the number of concepts K as a tuning parameter, we run both algorithms using a range of possible values of K and select the value of K that achieves the best prediction performance. For SPARFA-Lite, we only need to select the value of the single algorithm parameter λ that controls K. To assess the prediction performance of all three algorithms, we use three well-established performance metrics: prediction accuracy (COR), prediction likelihood (LIK), and area under the receiver operation characteristic curve (AUC) [11]. The prediction accuracy corresponds to the percentage of correctly predicted responses. The prediction likelihood corresponds to the average the predicted likelihood of the unobserved responses, i.e., $\frac{\sum_{i,j:(i,j)\in\bar{\Omega}_{\text{obs}}} p(Y_{i,j}|Z_{i,j})}{|\bar{\Omega}_{\text{obs}}|}$, where $\bar{\Omega}_{\text{obs}}$ represents the set of learner responses in the test set. The area under curve is a commonly-used performance metric for binary classifiers (see [11] for the details).

Table 1: Performance comparison of SPARFA-Lite vs. CF-IRT and SPARFA on predicting unobserved ratings for five educational datasets. Bold numbers represent the best performance among the three algorithms. SPARFA-Lite achieves comparable performance to CF-IRT and SPARFA in all experiments and metrics at significantly lower computational complexity.

		CF-IRT [5]	SPARFA [17]	**SPARFA-Lite**
Dataset 1	COR	0.8687	0.8711	**0.8737**
	LIK	**0.7286**	0.7195	0.7235
	AUC	0.8247	0.8056	**0.8299**
Dataset 2	COR	0.8061	0.8096	**0.8181**
	LIK	0.6393	**0.6759**	0.6707
	AUC	0.7985	0.7285	**0.8047**
Dataset 3	COR	**0.7263**	0.7000	0.7200
	LIK	**0.5876**	0.5334	0.5699
	AUC	**0.7629**	0.7116	0.7372
Dataset 4	COR	0.6967	0.7015	**0.7019**
	LIK	0.5538	**0.5587**	0.5537
	AUC	0.7180	**0.7249**	0.7175
Dataset 5	COR	0.6866	0.6880	**0.6903**
	LIK	0.5506	**0.5536**	0.5505
	AUC	0.7457	**0.7478**	0.7472

Results and discussion. Table 1 shows the mean of the performance metrics over 25 trials. We see that SPARFA-Lite achieves comparable performance as CF-IRT and SPARFA. Note that it outperforms CF-IRT and SPARFA on the most important performance metric–prediction accuracy (COR), with the exception of Dataset 3.

We emphasize that SPARFA-Lite is computationally more efficient than CF-IRT and SPARFA, since it (i) has only a single algorithm parameter and (ii) can be solved efficiently as it is a convex optimization problem. CF-IRT and SPARFA, in contrast, have multiple tuning parameters (including K) [5, 17], which means one have to run them multiple times to conduct a grid search over all possible values of these parameters. In particular, one Monte–Carlo trial of SPARFA-Lite on Dataset 1 only takes 3 sec, while CF-IRT and SPARFA require roughly 2 min. and 10 min. respectively, in MATLAB on a standard desktop PC with a 3.07 GHz Intel Core i7 processor (corresponding to 40× and 200× speed up). One can further reduce the computational complexity of SPARFA-Lite by replacing the nuclear norm constraint with the max-norm constraint [7, 18].

5.2 SPARFA-Lite learning analytics

Dataset and experimental setup. In this experiment, we use data collected from a high-school algebra test conducted on Amazon's Mechanical Turk [1]. The dataset consists of the quantized (with $P = 4$ ordinal values) graded responses of $N = 99$ learners answering $Q = 34$ questions, and the learner responses are fully observed. A total of $M = 13$ tags have manually been assigned to the questions. We use SPARFA-Lite to perform learning analytics on this dataset as described in Sec. 4.

Results and discussion. Table 2 shows the tag knowledge profile for a set of selected learners on the tags "Simplifying expressions," "Geometry," and "Systems of equations." The first row of

Table 2: Tag knowledge of selected learners. SPARFA-Lite performs robust LA by estimating each learner's tag knowledge from ordinal graded response data.

	Simplifying expressions	Geometry	System of equations
Class average	69 %	64%	30%
Best learner	84%	79%	34%
Average learner	70%	63%	24%
Worst learner	32%	34%	43%

the table shows the mean tag knowledge of all learners (in precent), while rows 2–4 show the tag knowledge (in percent) for the best learner, an average learner, and the worst learner, respectively. Leveraging these tag knowledge profiles, a PLS can automatically provide personalized feedback to learners on their strengths and weaknesses, and automatically recommend learning resources for remedial studies. For example, for the average learner in Table 2, a PLS would alert them to focus on the tag "System of equations" and recommend them learning resources associated with this tag, because their tag knowledge is below the class average. Moreover, a PLS can use this analysis to provide feedback to course instructors on the average tag knowledge of the entire class, helping them to make timely adjustments to their future course plan.

6. CONCLUSIONS

SPARFA-Lite is an efficient method that analyzes an incomplete set of quantized graded learner responses to questions to perform learning analytics. SPARFA-Lite achieves comparable or superior performance in predicting unobserved graded learner responses compared to existing factor-analysis methods, with significantly reduced computational complexity.

7. REFERENCES

[1] Amazon Mechanical Turk. http://www.mturk.com/mturk/welcome, Sep. 2012.

[2] T. Barnes. The Q-matrix method: Mining student response data for knowledge. In *Proc. American Association for Artificial Intelligence Workshop on Educational Data Mining*, July 2005.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Science*, 2(1):183–202, Mar. 2009.

[4] B. Beheshti, M. Desmarais, and R. Naceur. Methods to find the number of latent skills. In *Proc. 5th Intl. Conf. on EDM*, pages 81–86, June 2012.

[5] Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. Seaton, and D. Pritchard. Model-based collaborative filtering analysis of student response data: Machine-learning item response theory. In *Proc. 5th Intl. Conf. on EDM*, pages 95–102, June 2012.

[6] J. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optimization*, 20(4):1956–1982, Mar. 2010.

[7] T. Cai and W. Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *J. Machine Learning Research*, 14:3619–3647, Dec. 2014.

[8] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, Dec. 2009.

[9] H. Cen, K. R. Koedinger, and B. Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, and T. W. Chan, editors, *Intelligent Tutoring Systems*, volume 4053 of *Lecture Notes in Computer Science*, pages 164–175. Springer, June 2006.

[10] M. Chi, K. Koedinger, G. Gordon, and P. Jordan. Instructional factors analysis: A cognitive model for multiple instructional interventions. In *Proc. 4th Intl. Conf. on EDM*, pages 61–70, July 2011.

[11] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, Dec. 1994.

[12] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters. 1-bit matrix completion. *arXiv preprint:1209.3672*, Sep. 2012.

[13] M. Desmarais. Conditions for effectively deriving a Q-matrix from data with non-negative matrix factorization. In *Proc. 4th Intl. Conf. on EDM*, pages 41–50, July 2011.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2010.

[15] A. S. Lan, C. Studer, and R. G. Baraniuk. Matrix recovery from quantized and corrupted measurements. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, May. 2014, to appear.

[16] A. S. Lan, C. Studer, A. E. Waters, and R. G. Baraniuk. Tag-aware ordinal sparse factor analysis for learning and content analytics. In *Proc. 6th Intl. Conf. on Educational Data Mining*, pages 90–97, July 2013.

[17] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *J. Machine Learning Research*, 2014, to appear.

[18] J. Lee, B. Recht, N. Srebro, J. Tropp, and R. Salakhutdinov. Practical large-scale optimization for max-norm regularization. *Advances in Neural Information Processing Systems*, 22:1297–1305, Dec. 2010.

[19] N. Li, W. W. Cohen, and K. R. Koedinger. A machine learning approach for automatic student model discovery. In *Proceedings of the 4th Intl. Conf. on Educational Data Mining*, pages 31–40, Jul. 2011.

[20] F. G. Martin. Will massive open online courses change how we teach? *Communications of the ACM*, 55(8):26–28, Aug. 2012.

[21] OpenStaxTutor. https://openstaxtutor.org/, Sep. 2013.

[22] P. I. Pavlik, H. Cen, and K. R. Koedinger. Learning factors transfer analysis: Using learning curve analysis to automatically generate domain models. In *Proc. 2nd Intl. Conf. on EDM*, pages 121–130, July 2009.

[23] J. C. Stamper, T. Barnes, and M. Croy. Extracting student models for intelligent tutoring systems. In *Proc. Natl. Conf. on Artificial Intelligence*, volume 22, pages 113–147, July 2007.

[24] K. VanLehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The Andes physics tutoring system: Lessons learned. *Intl. J. of Artificial Intelligence in Education*, 15(3):147–204, Sep. 2005.

[25] D. Vats, C. Studer, A. S. Lan, L. Carin, and R. G. Baraniuk. Test size reduction for concept estimation. In *Proc. 6th Intl. Conf. on Educational Data Mining*, pages 292–295, July 2013.

Reengineering the Feature Distillation Process: A Case Study in the Detection of Gaming the System

Luc Paquette, Adriana M.J.A. de Carvalho, Ryan S. Baker, & Jaclyn Ocumpaugh
Teachers College Columbia University
525 West 120th Street
New York, NY, 10027
paquette@tc.columbia.edu

ABSTRACT

As education technology matures, researches debate whether data mining (EDM) or knowledge engineering (KE) paradigms are best for modeling complex learning constructs. A hybrid paradigm may capture strengths from both approaches. In particular, recent work has argued that successful data mining depends on thoughtful feature engineering. In this paper, we explore the use of cognitive modeling (a form of knowledge engineering) to enhance the feature engineering process for detectors of gaming the system, one of the most studied complex constructs in EDM. Using this construct enables us to measure the extent to which our techniques improve performance over previous models.

Keywords

Gaming the system, Cognitive Tutor, feature engineering, cognitive modeling, cognitive task analysis

1. INTRODUCTION

Over the last ten years, researchers interested in student disengagement have sought to improve the detection of gaming the system, behavior where students attempt to solve problems in an educational environment by exploiting properties of the system [2]. Within intelligent tutors, gaming the system manifests in several ways, including help abuse (e.g. [1], [12], [17]) and systematic guessing (e.g. [12], [17]). However, the construct appears to be quite complex, and while human coders are capable of achieving good inter-rater reliability for this construct [2], its complexity is still a challenge for the modeling community.

Gaming the system has now been modeled in a variety of systems using techniques from both Educational Data Mining (EDM) and Knowledge Engineering (KE). Within an EDM approach, classification algorithms are used to match training labels generated from *in situ* field observations (as in [17]) or from text replays (e.g. [2], [3] and [4]). These models have been effective at predicting gaming, but critics of EDM techniques argue that the resultant models are difficult to interpret.

KE models of gaming offer greater interpretability, but may oversimplify a construct that can manifest in many different ways. Often KE models focus only on 1-2 patterns of gaming, (e.g. quick incorrect answers or specific types of help abuse in [12], [17]), and it is reasonable to question whether such a complex and ill-defined construct can be fully described by 2-3 simple rules. In particular, simple rules may indicate gaming when a student skips to bottom-out hints to obtain answers (a pattern typical of gaming), but then pauses to self-explain, a behavior associated with positive learning outcomes [16].

In this study, we leverage Cognitive Task Analysis (CTA) [5], a form of KE, to produce a better EDM model. In line with results suggesting that attention to feature construct validity improves model goodness [15], we enhance construct validity by constructing features based on the explicit patterns articulated by expert human judges for how they recognize gaming [13]. We find that this method generates features that better reflect the meaningful units of student behavior that trigger experts to recognize gaming; using these features within an EDM process leads to better goodness than a model developed using cognitive task analysis alone.

2. COGNITIVE MODELING OF TEXT REPLAY CODING

Many EDM studies of gaming that have leveraged human judgments have relied upon text replays ([3], [4]), a sequence (clip) of student actions displayed in textual form, used to reliably and rapidly label systematic patterns of student behavior. Each replay contains time-stamped information about the context the student is interacting with (elements of the learning environment, including the relevant skills being tested), the input entered by the student, and the system's assessment of that input (right, wrong, a "bug" or common misconception, or a help request). A trained human coder labels each text replay as "gaming" or "not gaming".

In the CTA presented in [13], researchers interviewed and observed a gaming expert who had coded over 20,000 text replays from Cognitive Tutor Algebra, eliciting information about which cues were meaningful during that process. [13]'s CTA showed that expert coding involved two main processes: interpreting the student's actions and using these interpretations to identify patterns indicative of gaming. In particular, CTA identified 19 different constituents, or units of behavior, used by the expert. Analysis shows that the expert relied heavily on pauses to assess students' reflection and engagement, but gaming labels were also dependent upon contextualized information about the student input (e.g., was the student entering several similar answers in a row). Table 1 shows a partial list of these constituents.

Further analysis [13] found that no constituent is independently sufficient for identifying gaming, but that certain combinations of constituents are. Expert interviews identified 13 substantive patterns of the 19 constituents, which we refer to as *pattern features*. In this paper, we build on this work, using the constituents of [13]'s CTA to generate new pattern features and then applying EDM techniques to improve model performance.

Table 1. Some pattern constituents indicative of gaming

Constituent Description		Interpretive Label
C1	Pause ≤ 5 seconds before a help request	*[did not think before help request]*
C2	Pause ≥ 4 and ≤ 8 seconds per help message after a help request	*[scanning help messages]*
C3	Pause ≤ 3 seconds per help message after a help request	*[searching for bottom out hint]*
C4	Pause ≤ 5 seconds before a step attempt	*[guess]*
C5	Pause ≤ 8 seconds after a bug	*[did not read error messag e]*
C6	Answer was the same as the previous action, but in a different context	*[same answer/diff. context]*
C7	Answer was similar to the previous one (Levenshtein [16] distance of 1 or 2)	*[similar answer]*
C8	Context of the current action is not the same as the context for the previous	*[switched context before right]*
C9	Context for the current action is the same as the context for the previous	*[same context]*
C10	Answer or context is not the same as the previous action	*[diff. answer AND/OR diff. context]*

3. METHODS

3.1 Data

This study relies on data from Cognitive Tutor Algebra that have been used to study three gaming models ([3], [12], [13]), the Pittsburgh Science of Learning Center DataShop "Algebra I 2005-2006 (Hampton only)" dataset [8], which contains data from 59 students over the course of an entire school year. For [3], the data from 12 different lessons were segmented into clips of at least five actions and 20 seconds in length, within a single problem. A total of 10,397 text replays were presented to the expert, who labeled 708 (6.8%) as gaming [3].

[13] divided these text replay clips into subsets so that 75% of the clips from each category (531 gaming and 7,267 not gaming) are randomly assigned to a training set. The remaining 25% (177 gaming and 2,422 not gaming) were held-out for testing to ensure against overfitting. In this study, we use the same division during feature distillation, but final models are trained using standard cross-validation techniques.

3.2 Feature Distillation

The 19 constituents identified through [13]'s cognitive task analysis were used as features for the detectors built in this study. Constituent labels were applied to each clip, and the number of times each constituent appeared was computed.

Whereas the cognitive modeling approach in [13] attempted to replicate the expert's decision process, this paper's hybrid model searches for pattern features beyond those the expert directly articulated. This enables us to test a broader range of patterns on the large number of clips coded by the expert coder.

In order to generate new patterns, each clip from the training set was tagged with the 19 constituents identified in [13]. Constituent labeling involved a multistep process. First, student actions were given the following 4 labels: *help*, *attempt* (an attempted answer regardless of its correctness), *incorrect* (bug or wrong attempt) and *bug*. Note that the range in specificity here allows more than one action label to be applied in some cases. Next, 15 2-action and 57 3-action sequences were created from these action labels. For example, the 2-action sequences included "help → attempt" and "help → incorrect," and 3-action sequences included "incorrect → help → attempt." Sequences of 2 consecutive help requests were not generated since these are collapsed in the log files. Next, these sequences were tagged with constituent labels. In order to reduce the number of possible combinations, constituents that were associated with "not gaming" in the CTA were excluded from this process. These labels were then used to generate patterns consisting of 0-2 constituents. Impossible combinations of constituents were excluded (e.g., a help request could not be tagged with both [*scanning help message*] (C2) and [*searching for bottom out hints*] (C3)), producing 496,944 possible pattern features.

As the feature set was now enormous (increasing the potential for over-fitting), 2 steps were used to reduce the number of features tested in our final model. First, Cohen's Kappa [6] was used to evaluate how individual pattern features predicted gaming. Pattern features with Kappa < 0.05 (Kappa of 0 indicates chance) were eliminated first, reducing the number of possible features from 496,944 to 29,294. Next, a modified forward selection process was applied to the remaining patterns.

Although Kappa is a popular indicator of performance, it is relatively poor at eliminating patterns that identify many true positives at the cost of also identifying a large numbers of false positives. Since a combination of more specific sub-patterns might detect just as many true positives while detecting fewer false positives, a combination of more specific pattern features could achieve a better performance, even though the single overly general pattern would be selected first by a forward selection process based solely on Kappa. In order to prevent high rates of false positives, our forward selection process gave more weight to pattern features with a higher ratio of true positives (TP) to false positives (FP), a metric similar to precision. For the first iterations of our forward selection process, only pattern features with a TP to FP ratio ≥ to 1 were considered. This threshold was then lowered in increments of .05 each time Kappa could no longer be improved at the current threshold. This process repeated until the threshold became 0 and Kappa did not improve.

The ratio of TP to FP was used during forward selection instead of precision to reduce over-fitting to the training set. Amongst the generated pattern features, many detect a small number of TP while not capturing any FP. Those pattern features are likely to be overly specific to the training set. For such patterns, the value for the precision metric will be 1, the highest possible value, whereas the ratio of TP to FP is undefined (and are treated as 0 in our approach). As such, when executing forward selection using precision, those overly specific patterns will be added early to the set of best patterns, over-fitting to the training set. By contrast, those patterns will only be considered as possible best patterns when using TP to FP ratio if they still contribute to the overall performance at the end of the forward selection process.

Performance was evaluated on both the training and the test set to ensure that our forward selection algorithm did not overfit to the training data. This process was executed on the training set, resulting in the selection of 60 pattern features.

In addition to constituent and pattern features, 6 features, which we term "count features," were also considered. For these, we counted the number of actions of specific types during the clip, including (1) help, (2) attempts, (3) right answers, (4) incorrect answers (whether just wrong or a bug), (5) wrong answers (incorrect but not a bug), and (6) bugs. Combined with the other 2 feature types, this resulted in 85 features that were considered during the construction of *CognitiveHybrid-PF*, our first hybrid detector.

3.3 Validation and Performance

Detectors of gaming the system were constructed in RapidMiner 5.3 [11], using J48, JRip, Step Regression and Naïve Bayes, four algorithms that have been successful for past educational data mining problems. Performance was assessed using two metrics: Cohen's Kappa and A' [7]. A' is the probability that given a pair of two clips, one coded as gaming the system and the other coded as non-gaming, the model can accurately detect which clip was coded as gaming. A' is equivalent to the area under the ROC curve in signal detection theory and the Wilcoxon statistic [7]. A detector with an A' of 0.5 performs at chance, and a detector with an A' of 1.0 performs perfectly. A' was computed at the clip level, using the code at http://www.columbia.edu/~rsb2162/edmtools.html.

Detector performance during RapidMiner's forward selection was evaluated using a 6-fold student-level cross-validation. By cross-validating at the student level, we increase the confidence that our detectors will generalize to new students.

4. Results

The detectors were refined in three stages.

4.1 *CognitiveHybrid-PF*: Pattern Feature Detector

Our first detector was built using all 85 features. The Naïve Bayes algorithm performed best under 6-fold student-level cross-validation. (Kappa = 0.477 and A' = 0.770), accurately diagnosing 411 (58.05%) of the gaming clips and misdiagnosing only 495 (5.11%) of the non-gaming clips. The resulting model (Table 2) contains 22 of 85 potential features: 20 pattern features, 1 constituent-based feature (F21), and 1 count feature (F22). Except for F22, each was associated with a higher probability of gaming by the Naïve Bayes detector.

A closer inspection of the model improves our understanding of the actions and constituents that typify gaming in Cognitive Tutor Algebra. Only 3 pattern features (F5, F8, F12) selected in this model contain help constituents (C1 and C3). Instead, the predominant label was *incorrect* (both bugs and other wrong answers). This action appeared in 19/20 pattern features, omitted only from F7, where further scrutiny shows that the more specific *bug* label was a component of this and 7 other pattern features. This suggests that incorrect answers typify gaming, but a contrasting result also emerges. Right answers were possible in 13 of 20 pattern features, perhaps because a student who is gaming the system might get the correct answer by systematically guessing.

Overall, *CognitiveHybrid-PF's* feature selection suggests gaming behaviors in this corpus are typified by fast, systematic guessing patterns (e.g., providing similar answers or the same answers in different contexts). The effect of context changes appears to be nuanced but highly predictive when combined with other factors. A student who repeatedly enters the same answer in different contexts is not engaged in learning, but neither is a student who persists within one context after multiple incorrect steps.

4.2 *CognitiveHybrid-C*: Constituent Detector

Although *CognitiveHybrid-PF* shows substantial improvements over previous models in terms of cross-validated Kappa, room for improvement remains, especially for A'. Within *CognitiveHybrid-PF*, A' may have been reduced by the binary way the pattern features were used, resulting in high confidences for clips matching one or more pattern features, but confidences approaching 0 for all other clips. To address this issue, we construct

Table 2. Features utilized by *CognitiveHybrid-PF*.

Selected features	
F1	**incorrect** → [*same answer/diff. context*] & **incorrect**
F2	[*diff. answer AND/OR diff. contex* t] & **incorrect** → [*similar answer*] & **incorrect** → [*similar answer*] & **incorrect**
F3	**bug** & [*did not read error message*] → [*similar answer*] & **incorrect** → [*diff. answer AND/OR diff. context*] & **attempt**
F4	**incorrect** → [*same answer/diff. context*] & **attempt** → **bug**
F5	[*similar answer*] & **incorrect** → [*guess*] & [*similar answer*] & **attempt** → [*did not think before help request*] & [*same context*] & **help**
F6	**incorrect** → [*guess*] & [*similar answer*] & **attempt** → [*switched context before right*] & **incorrect**
F7	**bug** → [*guess*] & diff. answer AND/OR diff. context] & **bug** → [*guess*] & [*diff. answer AND/OR diff. context*] & **attempt**
F8	[*did not think before help request*] & [*same context*] & **help** → **attempt** → [*guess*] & [*similar answer*] & **incorrect**
F9	**bug** → [*similar answer*] & **incorrect** → [*diff. answer AND/OR diff. context]* & **bug**
F10	[*guess*] & [*same context*] & **attempt** → [*same context*] & **incorrect** → [*guess*] & [*similar answer*] & **incorrect**
F11	[*guess*] & [*same context*] & **incorrect** → [*diff. answer AND/OR diff. context*] & **attempt** → [*switched context before right*] & **incorrect**
F12	[*guess*] & [*similar answer*] & **incorrect** → [*diff. answer AND/OR diff. context*] & **incorrect** → **help** & [*searching for bottom-out hint*]
F13	**incorrect** → [*similar answer*] & **bug** → [*same answer/diff. context*] & **attempt**
F14	[*guess*] & [*diff. answer AND/OR diff. context*] & **incorrect** → [*guess*] & **bug** → [*guess*] & [*diff. answer AND/OR diff. context*] & **attempt**
F15	[*same context*] & **bug** & [*did not read error message*] → [diff. answer AND/OR diff. context] & **attempt** → [*guess*] & **incorrect**
F16	[*guess*] & [*same answer/diff. context*] & **attempt** → **incorrect** → [*diff. answer AND/OR diff. context*] & **incorrect**
F17	**incorrect** → [*guess*] & [*diff. answer AND/OR diff. context*] & **bug** → [*diff. answer AND/OR diff. context*] & **incorrect**
F18	[*guess*] & [*same context*] & **incorrect** → [*diff. answer AND/OR diff. context*] & **incorrect** → [*similar answer*] & **incorrect**
F19	[*similar answer*] & **incorrect** → [*same context*] & **incorrect** → [*similar answer*] & **incorrect**
F20	[*similar answer*] & **incorrect** → [*guess*] & [*similar answer*] & **incorrect** → [*similar answer*] & **attempt**
F21	number of times that [*switched context before right*] occured in the clip
F22	number of **right answers** in the clip

CognitiveHybrid-C, which relies only on constituent and count features. As with *CognitiveHybrid-PF*, Naïve Bayes was selected as the best algorithm when performance was assessed using 6-fold student-level cross validation. The exclusion of pattern features improved A' (0.875) but also increases the false positive rate, lowering Kappa (0.332). The model accurately diagnosed 323 (45.62%) gaming clips but misdiagnosed 657 (6.78%) non-gaming clips. Compared to *CognitiveHybrid-PF*, *CognitiveHybrid-C* is more parsimonious, requiring only 2 constituent features ([same answer/diff. context] and [thought about error]) and 4 count features (wrong, bug, incorrect, and right). Except for (count of right), all were associated with higher probabilities of gaming.

4.3 *CognitiveHybrid-E*: Ensemble Detector

Both *CognitiveHybrid-C and CognitiveHybrid-PF* have strengths, but neither is ideal. *CognitiveHybrid-E* (our ensemble detector) leverages the better prediction confidences (A') of *C* and the better classifications (Kappa) of *PF* by ensembling the two. This is done by averaging the two models' confidences together, and setting a threshold of 0.5. *CognitiveHybrid-E*, when student-level cross-validated, achieves good Kappa (0.457) and A' (0.901), accurately diagnosing 392 (55.37%) gaming clips and misdiagnosing only 476 (4.91%) not-gaming clips.

CognitiveHybrid-E's performance (Kappa = 0.457, A' = 0.901) is better than previous detectors trained on the same data. Neither [3]'s decision tree detector (Kappa = 0.40) nor their latent response model (Kappa = 0.04) is cross-validated [3] and (unpublished) cross-validation drops the decision tree detector's performance to Kappa = 0.24. [13]'s cognitive model performed well on training data (Kappa = 0.430), but performance dropped when applied to a held-out test set (Kappa = 0.330). *CognitiveHybrid-E* also compares favorably to other published gaming detectors: [4], conducted in SQL-Tutor, reported student-level cross-validated Kappa = 0.36, A' = 0.770. In ASSISTments [14], a model of gaming achieved Kappa = 0.370 and A' = 0.802; an earlier model in ASSISTments [17] achieved Kappa = 0.181.

5. CONCLUSIONS AND DISCUSSION

In this study, we provide enhanced, automated models of gaming-the-system for Cognitive Tutor Algebra, improving model performance for a construct already well established in the literature ([3], [13]). Improvements were driven by a hybrid approach that leverages both KE and EDM techniques, using cognitive modeling of human experts during feature distillation and then applying EDM practices to combine these operators to predict gaming.

These results have implications for debates between KE and EDM approaches. They suggest that EDM researchers could substantially improve their feature engineering by employing KE techniques during feature distillation. At the same time, they also attest to limitations in relying solely on human experts to define the constructs in automated detectors. There are many constructs that humans can easily recognize but are still difficult to define. The detailed interview method used in [13] and built on here foregrounds the value of expert evaluations. By considering hundreds of thousands of possible patterns, EDM methods can improve performance. For modeling complex constructs, the combination of KE and EDM can be stronger than either method alone.

6. ACKNOWLEDGMENTS

This research was supported by a FRQNT (Fonds de recherche du Québec - Nature et technologies) post-doctoral fellowship and by National Science Foundation grant #SBE-0836012.

7. REFERENCES

[1] Aleven, V., McLaren, B. M., Roll, I., Koedinger, K. R. 2004. Towards Tutoring Help Seeking: Applying Cognitive Modeling to Meta-Cognitive Skills. *Proc of ITS 2004*, 227-239.

[2] Baker, R. S. J. d., Corbett, A. T., Wagner, A. Z. 2006. Human Classification of Low-Fidelity Replays of Student Actions. *Proc of EDM Workshop at ITS 2006*, 29-36.

[3] Baker, R. S. J. d., de Carvalho, A. M. J. A. 2008. Labeling Student Behavior Faster and More Precisely with Text Replays. *Proc of EDM 2008*, 38-47.

[4] Baker, R. S. J. d., Mitrovic, A., Mathews, M. 2010. Detecting Gaming the System in Constraint-Based Tutors. *Proc of UMAP 2010*, 267-278.

[5] Clark, R. E., Feldon, D., van Merriënboer, J., Yates, K., Early, S. 2008. Cognitive Task Analysis. In *Handbook of Research on Educational Communications and Technology (3rd ed.)*, 575-593.

[6] Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational & Psych Measurement*, 20,11, 37-46.

[7] Hanley, J., McNeil, B. 1982. The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143, 29-36.

[8] Koedinger, K. R., Baker, R., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. 2010. *A Data Repository for the Community: The PLSC DataShop*. CRC Press, Boca Raton.

[9] Koedinger, K. R., Corbett, A. T. 2006. Cognitive Tutors: Technology Bringing Learning Sciences to the Classroom. *The Cambridge Handbook of the Learning Sciences*, 61-77.

[10] Levenshtein, A. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10, 8, 707-710.

[11] Mierswa, I., Scholz, M., Klinkenberg, R., Wurst, M., Euler, T. 2006. Yale: Rapid Prototyping for Complex Data Mining Tasks. Proc of KDD 2006, 935-940.

[12] Muldner, K., Burleson, W., Van de Sande, B., VanLehn, K. 2011. An Analysis of Students' Gaming Behaviors in an Intelligent Tutoring System: Predictors and Impact. *User Modeling and User Adapted Interaction*, 21, 99-135.

[13] Paquette, L., de Carvalho, A. M. J. A., Baker, R. S. *accepted*. Towards Understanding Expert Coding of Student Disengagement in Online Learning. *Proc. Cog Science Society*.

[14] Pardos, Z.A., Baker, R.S., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M. *in press*. Affective States and State Tests: Investigating How Affect and Engagement During the School Year Predict End of Year Learning Outcomes. To appear in *Journal of Learning Analytics*.

[15] Sao Pedro, M., Baker, R., Gobert, J. 2012. Improving Construct Validity Yields Better Models of Systematic Inquiry, Even with Less Information. *Proc of UMAP 2012*, 249-260.

[16] Shih, B., Koedinger, K. R., Scheines, R. 2008. A Response Model for Bottom-Out Hints as Worked Examples. *Proc of EDM 2008*, 117-126.

[17] Walonoski, J. A., Heffernan, N.T. 2006. Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. *Proc of ITS 2006*, 382-391.

SketchMiner: Mining Learner-Generated Science Drawings with Topological Abstraction

Andy Smith[1], Eric Wiebe[2], Bradford Mott[1], James Lester[1]

[1]Department of Computer Science [2]Department of STEM Education
North Carolina State University, Raleigh, NC, USA
{pmsmith4, wiebe, bwmott, lester}@ncsu.edu

ABSTRACT

Mining learner-generated sketches holds significant potential for acquiring deep insight into learners' mental models. Drawing has been shown to benefit both learning outcomes and engagement, and learners' sketches offer a rich source of diagnostic information. Unfortunately, interpreting learners' sketches—even sketches comprised of semantically grounded symbols—poses significant computational challenges. In this paper we describe SketchMiner, an educational sketch mining framework that automatically maps learners' symbolic sketches to topology-based abstract representations that are then analyzed with graph similarity metrics to perform automated assessment and misconception discovery. SketchMiner has been used to mine a corpus of symbolic science sketches created by upper elementary students in inquiry-based drawing episodes as they interact with an intelligent science notebook in the domain of physical science. Results of a study with SketchMiner suggest that it can correctly assess learners' symbolic sketches.

Keywords

Student modeling, Sketch analysis

1. INTRODUCTION

Diagrams and sketching are fundamental to science education. From primary through post-secondary education, students use drawings and graphical representations to make sense of complex systems and as a tool to organize and communicate their ideas to others. Studies have shown that learning strategies focusing on learner-generated sketches can produce effective learning outcomes, such as improving science text comprehension and student engagement [12], facilitating the writing process [11], and improving the acquisition of content knowledge [3]. Furthermore, spatial ability has been recognized as a predictor of STEM success even when accounting for mathematical and verbal ability [17].

Unlike the well studied areas of how people learn from writing text, viewing graphics, and reading, relatively little is known about how the generation of scientific drawings affects learning. Van Meter and Garner [9] posit that students asked to draw a picture engage in three cognitive processes: selecting relevant information, organizing the information to build up an internal verbal model, and constructing an internal nonverbal representation to connect with the verbal representation. Others suggest that drawing can be a meaningful learning activity requiring both essential and generative processing to mentally connect multiple knowledge representations [14].

The benefits of learner-generated sketching can best be realized by thoughtfully designing activities within a well-designed curriculum, as the positive effects of drawing strongly depend on the quality of the learner-generated products and scaffolding [10]. The act of generating a visual representation is a cognitively demanding task and, as such, requires scaffolds to guard against excessive and extraneous cognitive load [16]. Effective scaffolds for drawing include providing cutout figures, guiding questions, and targeted drawing prompts [7,19].

From a computational perspective, learner-generated drawings pose significant challenges. Even in an environment with predefined symbolic elements, the generative nature of the task yields a very large solution space of unique drawings and configurations. The work presented here describes initial efforts to mine learner-generated science drawings. To automatically cluster and compare drawings, the proposed framework uses a multi-step process of translating trace sketch behavior data of student drawings into topological representations. This process consists of converting the drawn elements into a graph representation based on a topology derived from the domain and using a modified edit distance methodology for comparing the topological graphs. We show how these comparisons can be used to analyze drawings to detect misconceptions, as well as to cluster student solutions in a manner that exhibits high fidelity with respect to human categorization.

This paper is structured as follows. Section 2 discusses other approaches that have been used to analyze student sketching. Section 3 describes the tablet-based learning environment that was used to collect the symbolic sketch dataset from elementary students. Section 4 introduces SketchMiner, a sketch data mining systems that automatically analyzes and compares student drawings using topological graphs. Finally, Section 5 describes an application of SketchMiner to cluster student drawings compared to a human clustering.

2. RELATED WORK

Sketch analysis poses significant computational challenges, with a majority of prior work focused on sketch recognition. For example, sketch recognition frameworks have been designed for domains such as organic chemistry and circuits in which free-hand drawing is translated into domain-specific symbols [1]. Another system, Mechanix, combines free-hand recognition capabilities with error checking to create feedback for undergraduate engineering students enrolled in a statics course [15].

Bollen and van Joolingen's SimSketch merges sketching with modeling and simulation of science phenomena [2]. In SimSketch, user free-hand drawings are segmented into objects by the system, and then annotated by the user with a variety of behaviors and attributes. Students can then run a simulation based on their drawing and see the results before revising their sketch. SimSketch has been evaluated in a planetarium setting and been shown to be both a functionally useable and enjoyable system for visitors.

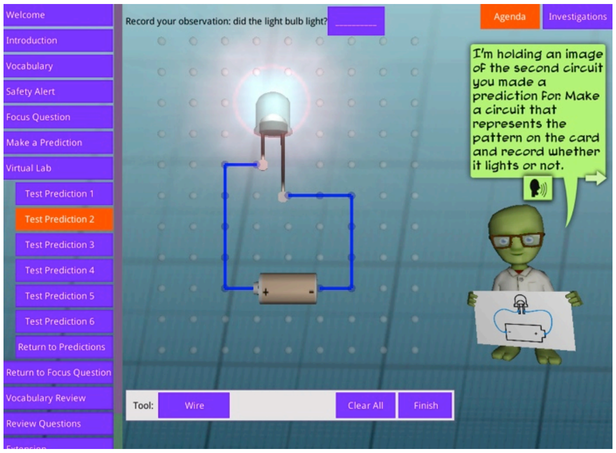
Another promising line of investigation for studying learner-generated drawing in educational settings centers on the CogSketch system [5]. CogSketch has been developed as an open-domain sketch understanding system. Sketch worksheets were built within CogSketch, and used in a study to collect and cluster undergraduate geology student sketches by an analogical generalization engine [4].

3. LEONARDO CYBERPADS

Recent years have witnessed growing interest in introducing science notebooks into elementary science classrooms [13]. Science notebooks capture students' inquiry-based activities in both written and graphical form, potentially providing a valuable source of both diagnostic and prognostic information. However, because elementary teachers have limited training in science pedagogy, they often struggle with effectively using science notebooks in classroom learning activities [18].

For the past three years our laboratory has been developing a digital science notebook, the LEONARDO CyberPad (Figure 1), which runs on tablet computing platforms. LEONARDO integrates intelligent tutoring systems technologies into a digital science notebook that enables students to graphically model science phenomena. With a focus on the physical and earth sciences, the LEONARDO PadMate, a pedagogical agent, supports students' learning with real-time problem-solving advice. LEONARDO's curriculum is based on that of the Full Option Science System [8] and is aligned with the Next Generation Science Standard goals in elementary school science education [20].

Throughout the inquiry process, students using the LEONARDO CyberPad are invited to create symbolic sketches, including electrical circuits. Given the challenges of machine analysis of freehand sketching, as well as concerns of excessive cognitive demand for elementary students working in such an unstructured space [18], LEONARDO supports symbolic drawing tasks. To preserve the generative processing hypothesized to be of great benefit for learner-generated drawings strategies, each activity begins with a blank page so that the representations must be created from scratch. Students then choose from a variety of semantically grounded objects and place them at various points in the drawing space. For example, objects for the electricity unit include light bulbs, motors, switches, and batteries. Students then place wires on the drawing space, connecting the various objects to simulate proper electrical behavior. This focuses the learning activity on choosing the appropriate circuit elements and creating the appropriate circuit topology. Drawing tasks vary in complexity from copying a picture of a circuit held up by the PadMate, to recreating a circuit made during a physical investigation, to creating more complex circuits designed to increase their understanding of series and parallel circuits.



Record your observation: did the light bulb light? __________

I'm holding an image of the second circuit you made a prediction for. Make a circuit that represents the pattern on the card and record whether it lights or not.

Figure 1. Screenshot of the LEONARDO CyberPad

4. TOPOLOGY-BASED SKETCH MINING

To analyze student drawings, SKETCHMINER first translates them into a more abstract representation. It takes as input trace logs from students' work in the CyberPad. From the trace logs it extracts student actions at a level of granularity capable of producing replay-quality representations of the drawing activities. From these actions it extracts the state of the student drawing at each point in the activity. For the analyses reported in this paper, we focus only on the final submitted sketches rather than the multiple drawings generated during the sketching process. The set of objects and locations are then utilized by a simulation engine that supports the querying of topological features of the drawing.

SKETCHMINER uses these topological features to generate a labeled graph representation of the drawing. Topological graphs provide two key representational benefits. First, they are very flexible and can be used across many domains. For the domain of circuits, our representation focuses on the electrical topology of the circuit drawing, which could be replaced or augmented by other features such as two-dimensional spatial topology. Second, graphs are easily visualized and interpretable by humans, which facilitates the interpretation of patterns and features extracted by automated analysis.

The first step in the translation from drawings to topological graphs is encoding the non-wire circuit elements. Circuit elements

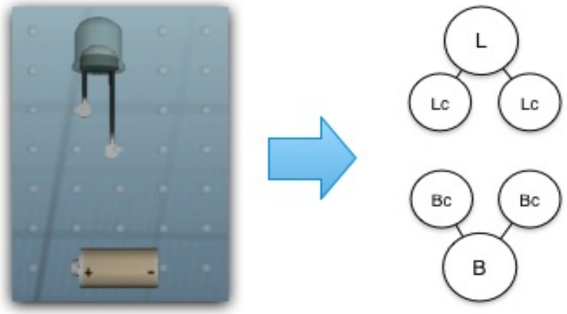


Figure 2. Circuit elements and corresponding topology

are represented as nodes in the graph. Because there are only two points where each node can interact with other objects in the drawing space, each node is connected to two nodes representing its contact points (Figure 2).

After creating the nodes of the graph, SKETCHMINER then generates the edges between them. For each contact point in the graph, the simulation engine uses a depth first search to return all other contact points reachable with a zero resistance path. If one or more paths exist between contact points, they are then connected with a single edge in the graph (Figure 3).

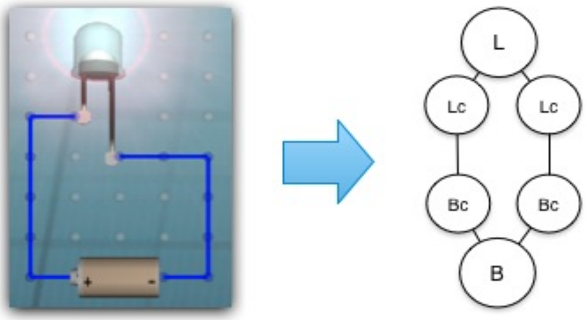


Figure 3. Connections encoded as edges

While multiple methods can be used to compare the similarity of graphs and trees, SKETCHMINER uses a method capable of numerically summarizing the difference between topographical states that also provides a description of how to transition from one state to the other. In particular, it uses a modified form of *edit distance*. Edit distance has been used to characterize errors in a variety of domains and is perhaps best known for its application in natural language spelling correction. Edit distance captures the difference between two representations as a series of edit operations. Additionally, these edit operations can be weighted, with the sum of necessary operations equaling the edit distance.

SKETCHMINER uses edit distance to measure the number of element additions, element deletions, edge additions and edge deletions needed to match two topologies. While traditional string edit distances tend to also utilize substitution, we chose to treat this instead as deleting an element, then adding a new one because this is the path a student would have to take to modify his or her drawing.

To determine the sequence of edit operations necessary to match two topologies, SKETCHMINER utilizes a guided search of possible actions to determine the lowest cost path through the operation space. While there are more efficient algorithms for graph edit distance, (e.g., see [6] for a survey), the greater complexity of these is not justified for the size of topological graphs generated from student sketches in this work.

Another design decision for SKETCHMINER considered how to weigh different edit operations for calculating the edit distance. An unweighted edit distance produces some undesirable effects. In particular, an unweighted score does not differentiate well between different types of errors. Consider a target drawing of a complete circuit featuring a battery and a motor. A blank submission and a complete circuit with the motor contacts short-circuited will both produce the same edit distance.

One approach to correcting for this is to adjust the weighting of actions. A subset of the student answers was analyzed with subject matter experts in an attempt to determine how the edit distance was aligning with curricular goals and assessment of different types of errors. A weighting scheme was generated to penalize missing elements at a cost of 4, extra elements at a cost of 2, and extra/missing edges at a cost of 1. SKETCHMINER uses this weighting scheme.

5. CORPUS ANALYSIS

For the analyses of SKETCHMINER reported here, a corpus of fourth grade symbolic drawings was collected with the LEONARDO CyberPads running on iPads in elementary classrooms in North Carolina and California. After data cleaning, drawing activities from 132 students were used for the analysis. Student drawings were scored in comparison to normative models constructed by the research team. Because there may be multiple correct solutions to a given exercise, student submissions were scored against multiple "correct" solutions and assigned the score of the closest match. These scores were then used to qualitatively analyze the student drawings as a basis for the distance metric for unsupervised clustering and for misconception detection.

To evaluate SKETCHMINER's edit distance's value as an assessment metric, we clustered student drawings using both the weighted and unweighted topographical edit distance as the distance metric. In order to evaluate the clusters, two independent coders from the project's education team developed a rubric (described in Table 2) and scored the student responses for a circuit involving a switch, motor, and battery connected in series. Based on the rubric, the drawings were independently classified into 4 clusters by the two coders ($\kappa = .9$), creating a gold standard clustering to validate our clusters against.

After the hand coding, we then ran an automated cluster analysis on the student drawings based on the SKETCHMINER generated codings. To cluster the drawings we utilized the WEKA toolkit implementation of k-means clustering with k=4 to align with the human coding. Because k-means can be dependent on initialization, the analysis was run 10 times with different random seeds and the results averaged.

Table 1. Classification accuracy

Distance Metric	Accuracy	Precision	Recall
Unweighted	.73	.56	.63
Weighted	.86	.74	.76

As shown in Table 1 above, SKETCHMINER produced strong alignment with the human classifications, with the weighted edit distance producing better results than unweighted. The improved accuracy is a result of the weighted edit distance outperforming the unweighted edit distance at separating the three error classes.

Table 2. Classification by class for weighted edit distance

Class	Accuracy	Precision	Recall
1 (Blank)	.89	.61	1
2 (No Structure)	.87	.66	.5
3 (Some Structure)	.86	.92	.6
4 (Correct)	.98	1	.96

Further analysis of the weighted edit distance classification reveals that the process produced near-perfect accuracy on correct answers (Class 4). Inspection of the misclassified correct student sketches showed one example where the student had created the correct circuit, and a smaller unrelated circuit on a different part of the drawing space which inflated its edit distance. The other human-coded correct answer misclassified by SKETCHMINER was due to the student creating the correct topology but using a light bulb instead of a motor.

For classifying errors, the clustering showed strong alignment with empty entries, but had difficulty separating Class 2 errors (elements present but with no structure) from empty submissions. One possible way of improving this in the future could be to treat absence-of-circuit elements as a special case error.

6. CONCLUSION

Understanding how students learn from drawing is a foundational problem in learning analytics. Tablet-based science notebooks, such as the one provided by the Leonardo CyberPad, offer an excellent "laboratory" for instrumenting the drawing process and afford significant opportunity for educational data mining techniques. In this paper we have introduced SketchMiner, which utilizes a graph-based representation of drawing topologies to automatically interpret learner-generated symbolic sketches. In an analysis of SketchMiner's application to a corpus of fourth grade student symbolic sketches, it was found that its assessment of student drawings aligns with human-provided assessments.

The results show promise as a means of automatically assessing learner drawings and suggest several lines of investigation for future research. First, while "distance to solution" is a valuable metric, SketchMiner's edit distance could also be used to compare errors to each other. Preliminary analysis using this technique has shown promise for identifying common error states that could be used in curriculum redesign or to generate targeted scaffolding for students.

Another area for future research is applying SketchMiner to more topologically complex domains. Because the topographical relations in the domain of circuits are somewhat sparse, SketchMiner's representations would need to be evaluated on more complex student drawings containing more diverse sets of elements and relationships with more complex topologies.

Perhaps the most promising area for analysis is investigating the drawing process itself. Topographical representations can be created at any point in the drawing process, allowing for analysis of sequences and patterns in student drawing. Models learned from corpora of learner drawing processes can be used to create more accurate models of learners' conceptual representations, as well as the basis for providing customized scaffolding to support a broad range of learner populations.

7. ACKNOWLEDGMENTS

This research is based upon work supported by the National Science Foundation under Grant No. DRL-1020229. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] Alvarado, C. and Davis, R. SketchREAD: a multi-domain sketch recognition engine. *Proceedings of the 17th annual ACM symposium on User interface software and technology*, ACM (2004), 23–32.

[2] Bollen, L. and Joolingen, W. van. SimSketch: Multi-Agent Simulations Based on Learner-Created Sketches for Early Science Education. *IEEE Transactions on Learning Technologies 6*, 3 (2013), 208–216.

[3] Britton, L. and Wandersee, J. Cutting up Text to Make Moveable, Magnetic Diagrams: A Way of Teaching & Assessing Biological Processes. *The American Biology Teacher 59*, (1997), 288–291.

[4] Chang, M. and Forbus, K. Clustering Hand-Drawn Sketches via Analogical Generalization. *Proceedings of the 25th Annual Conference on Innovative Applications of Artificial Intelligence*, (2013).

[5] Forbus, K., Usher, J., Lovett, A., Lockwood, K., and Wetzel, J. CogSketch: Sketch Understanding for Cognitive Science Research and for Education. *Topics in Cognitive Science 3*, 4 (2011), 648–666.

[6] Gao, X., Xiao, B., Tao, D., and Li, X. A survey of graph edit distance. *Pattern Analysis and Applications 13*, 1 (2009), 113–129.

[7] Lesgold, A.M., Levin, J.R., Shirmon, J., and Guttman, J. Pictures and young children's learning from oral prose. *Journal of Education Psychology*, 67 (1975), 636–642.

[8] Mangrubang, F. Preparing elementary education majors to teach science using an inquiry-based approach: The Full Option Science System. *American Annals of the Deaf 149*, 3 (2004), 290–303.

[9] Van Meter, P. and Garner, J. The Promise and Practice of Learner-Generated Drawing: Literature Review and Synthesis. *Educational Psychology Review 17*, 4 (2005), 285–325.

[10] Van Meter, P. Drawing construction as a strategy for learning from text. *Journal of Educational Psychology 93*, 1 (2001), 129–140.

[11] Moore, B. and Caldwell, H. Drama and drawing for narrative writing in primary grades. *The Journal of Educational Research 87*, (1993), 100–110.

[12] Rich, R.Z. and Blake, S. Using pictures to assist in comprehension and recall. *Intervention in School and Clinic 29*, 5 (1994), 271–275.

[13] Ruiz-Primo, M. a., Li, M., Ayala, C., and Shavelson, R.J. Evaluating students' science notebooks as an assessment tool. *International Journal of Science Education 26*, 12 (2004), 1477–1506.

[14] Schwamborn, A., Mayer, R.E., Thillmann, H., Leopold, C., and Leutner, D. Drawing as a generative activity and drawing as a prognostic activity. *Journal of Educational Psychology 102*, 4 (2010), 872–879.

[15] Valentine, S., Vides, F., Lucchese, G., et al. Mechanix: A Sketch-Based Tutoring System for Statics Courses. *Proceedings of the Twenty-Fourth Conference on Innovative Applications of Artificial Intelligence*, (2012).

[16] Verhoeven, L., Schnotz, W., and Paas, F. Cognitive load in interactive knowledge construction. *Learning and Instruction 19*, 5 (2009), 369–375.

[17] Wai, J., Lubinski, D., and Benbow, C.P. Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology 101*, 4 (2009), 817–835.

[18] Wiebe, E.N., Bedward, J.C., and Madden, L.P. Graphic Representations in Science Notebooks : A Vehicle for Understanding Science Inquiry in the Elementary Classroom. *Presented at AERA*, (2009).

[19] Zhang, H. and Linn, M. Using drawings to support learning from dynamic visualizations. *Proceedings of the 8th International Conference of the Learning Sciences*, (2008), 161–162.

[20] Achieve. In *Next Generation Science Standards - First Public Draft*. Achieve, Inc, 2012.

Teachers and Students Learn Cyber Security: Comparing Software Quality, Security

Shlomi Boutnaru
School of Education
Tel Aviv University
ISRAEL
boutnaru@mail.tau.ac.il

Arnon Hershkovitz
School of Education
Tel Aviv University
ISRAEL
arnonhe@tauex.tau.ac.il

ABSTRACT

In recent years, schools have added cyber security to their computer science curricula. While doing so, existing teachers are trained with the new material. In this study we explore differences in teachers' and students' learning of cyber security, implementing a multi-way, data-driven approach by comparing measures of software quality and security. Our findings suggest that teachers' codes have a better quality than the students', and that the students' codes are slightly better secured than the teachers'. The findings imply on the teachers benefit from their prior knowledge and experience. Also, findings shed light on the difference between quality and security in today's programming paradigms.

Keywords

Cyber security, code metrics, software quality, software security, hierarchical clustering, decision tree.

1. INTRODUCTION

Educational systems worldwide often adopt "hot", emerging topics to their curricula. Usually, teachers from within the system are quickly trained to teach the new material. The current study aims on understanding the differences in teachers' and students' learning of new material in the case of cyber security (also known as computer security, IT security), that is, the practice of protecting computer systems from unauthorized access, change or destruction. This understanding might contribute to the pedagogy of teaching new materials, as well as to teacher development, shedding light on previous findings regarding novices' and experts' programing knowledge [1,9]. Studies in this field had usually used various measures to assess experts' and novices' programming skills and knowledge, mostly based on qualitative data collection (mainly programming-related tasks and interviews), rather than on assessment of programs written. Our approach is to use automatically extracted software quality and security features.

Explicit metrics for measuring different dimensions of code quality have been developed from the late 1960s, shortly after the development of the then-new domain of software engineering [3,5]. Metrics were defined with their automation in mind. As setting a numerical value for metrics might be time-consuming, subjective and expensive, "one would prefer for large programs an automated algorithm which examine the program and produces a metric value" [3; p. 596]. In recent years, EDM/LA methods have been used along with software metrics, allowing complex structure-based features, as well as adding variables measuring student-computer interaction [e.g., 2,7]. In this paper, we take an EDM approach, together with a comprehensive set of software metrics for both quality and security. We use both quality and security metrics.

As the main purpose of the current study is to explore the way novice students and experienced teachers learn cyber security – most commonly involves learning Python – we chose to focus on software metrics derived from the standards of that language. Therefore, we used the *Style Guide for Python Code (PEP 8)*[1] as the basis to the metrics. (These metrics were not accessible to the participants, and were only assessed in retrospect.) As there is no yet a Python standard for security, and based on the similarities between Python and C++, we based our quality metrics on the *C++ Secure Coding Standard*, by Carnegie Mellon University's CERT[2]. Both these standards are widely used in code evaluation

2. METHODOLOGY

Participants in this study were 31 11th- and 12th-grade students from two Israeli high-schools, 17-18 years old; and 18 high-school computer science teachers from different parts of Israel, 31-53 years old. Two of the latter were the teachers of the participant students.

Each of the participant teachers attended one of two cyber security programs (June 2012 – March 2013 or September 2013 – January 2014). The participant students took a curriculum-based cyber security program, as part of their computer science studies, during 2012/3 school year. Solutions (in Python) to tasks assigned during these programs were collected and analyzed.

Overall, 109 source files were collected – 68 teachers' and 41 students'. The teachers were assigned with four different exercises (writing a UDP echo server, a basic TCP command server, an advanced TCP command server, and a Web server); the students were assigned with three different tasks (writing a UDP echo server, an advanced TCP command server, and a TCP-based Chat).

Number of actual participants in the analysis was decreased to 17 teachers (with 60 files) and 15 students (with 27 files), as sometimes teachers/students worked in pairs or triples. When the same pair/triple had submitted all of the exercises, we arbitrarily left only one of the group in the data set. When pairs/triples had changed over the course of the program, we arbitrarily assigned chose only one representative for each submission.

[1] This guide was co-authored by Python creator, Guido van Rossum. Available on http://legacy.python.org/dev/peps/pep-0008 [accessed 3 May 2014].

[2] Available at https://www.securecoding.cert.org [accessed 3 May 2014].

2.1 Feature Engineering

Features were evaluated at the code-level; for the participant-level analysis (descriptive statistics, hierarchical tree), feature values were averaged across each participant's source files.

2.1.1 General Features (6 features)

For each source file, the general features are the following:

- *Number of Statements* (code size);
- *Number of Comment Lines*;
- *Documentation Rate* (= *Comment Lines* / *Statements*);
- *Number of Lines* (statements, comments and empty lines);
- *File Name Length* [characters; excluding the extension .py];
- *File Name Meaningfulness* (1 – file name is not meaningful at all; 2 – partly meaningful; 3 – very meaningful).

2.1.2 Quality Features (20 features)

These were automatically extracted by running *Pylint* (http://pylint.org), a common source code bug and quality checker for Python which follows PEP 8 style guide. Pylint defines five categories of standard violations/errors:

1. **Convention (C; 18 measures)**. Recommendations of software structural quality. Convention measures indicate standard violations (e.g., function/variable name does not match a regular expression defined in the standard);
2. **Warning (W; 61 measures)**. Python-specific problems that do not follow Python's best practices and may cause run time bugs (e.g., an unused import from wildcard import);
3. **Error (E; 32 measures)**. Probable bugs in the code that relate to general programming concepts (e.g., the use of a local variable before its assignment);
4. **Refactor (R; 15 measures)**. A "bad smell" code (derived from the term refactoring. the process of restructuring existing computer code without changing its external behavior). Such violation might be indicated when a function takes too many variables as input;
5. **Fatal**. This are errors in Pylint processing and not in the source file itself, hence were excluded.

Pylint scans the code and returns a list of measures for which violations/errors found, along with their count (we consider 0 for the measures that were not triggered by Pylint). Based on Pylint output, the following features were computed for each category:

- *Mean Count (C/W/E/R)* – mean count of violations/errors across all the category's measures.
- *Normalized Mean Count (C/W/E/R)* – *Mean Count* divided by code size (*Number of Statements*);
- *Rate of Triggered Measures (C/W/E/R)* – number of triggered measures divided by total number of measures;
- *Triggered Category (C/W/E/R)* – indicating whether at least one measure of it was triggered.
- *Normalized Triggered Category (C/W/E/R)* – *Triggered Category* divided by code size (*Number of Statements)*.

2.1.3 Security Features (6 features)

These features – extracted using scripts written by the research team – are binary, indicating whether the relevant mechanism was implemented (1) or not (0).

- *Input Validation* (the process of ensuring that a program operates on clean, correct and expected input);
- *Anti-Spoofing Mechanism* (spoofing attack is a situation in which an attacker masquerades as another entity by sending specially crafted data that seems as it was send from the legitimate source);
- *Bound Checking* (checking whether a variable is within some range before it is used);
- *Checking for Errors* (not checking return codes for errors can cause logical security bugs/crashing of the program that can cause Denial of Service attacks);
- *Sensitive Data Encryption*;
- *Client-Side-Only Security* (when the server relies on protection mechanisms placed on the client side only);

Among these, *Client-Side-Only Security* is the only one for which a 0-value denotes a good behavior.

3. RESULTS

3.1 Descriptive Statistics

3.1.1 General Features

Means of four general metrics are significantly different between students and teachers: *Number of Statements*, *Number of Lines*, *File Name Length*, and *File Name Meaningfulness*; on average, students' programs were longer than the teachers', and teachers' file names were longer and more meaningful than the students'. The difference regarding code size (*Number of Statements* and *Number of Lines*) might hint that teachers have a better grasp of the concept of programming with Pyhton, as this language allows far fewer lines compared to other languages. No significant differences were found between the means of the two documentation-related features. Average *Documentation Rate* was 0.1, which shows a reasonable documenting practice in Python. Results are summarized in Table 1.

Table 1. Descriptive statistics, t-test results for *general features* (one decimal place representation unless mean<0.1)

Variable	Mean (SD) N=32	Mean (SD), Teach. N=17	Mean (SD), Stud. N=15	t(30)^{a}
Number of Statements	51 (28.3)	40.5 (19.7)	62.9 (32.3)	2.3*, df=22.6^{b}
Number of Comments	6.1 (7.4)	5.5 (7.8)	6.8 (7.0)	0.5
Documentation Rate	0.1 (0.1)	0.1 (0.2)	0.1 (0.1)	-0.4
Number of Lines	56.9 (29.6)	45.4 (23.8)	69.7 (30.9)	2.5*
Name Length	10.8 (5.1)	12.9 (4.0)	8.4 (5.2)	-2.8**
Name Meaning.	1.3 (0.5)	1.6 (0.4)	0.9 (0.5)	-4.3**

* $p<0.05$, ** $p<0.01$. ^{a} Unless otherwise stated, df=30.

^{b} Levene's test for equality of variance resulted with a significant result, hence equal variances not assumed.

3.1.2 Quality Features

Means of eight quality metrics of convention (C) and warning (Q) type are significantly different between students and teachers (see Table 2): *Mean Count*, *Normalized Mean Count*, *Rate of Triggered Measures* – for both C and W; *Trigged Category W*, and *Normalized Triggered Category C*. On average, students had more convention- and warning-type violations than the teachers. As convention guidelines improve code readability and maintainability, these differences might indicate on the teachers' smoother migration to programming in a new language.

Table 2. Descriptive statistics, t-test results for *quality features* (one decimal place representation unless mean<0.1 or difference needs to be shown)

Variable	Mean (SD) N=32	Mean (SD), Teach. N=17	Mean (SD), Stud. N=15	t(30)^{a}
Mean Count C	72.3 (56.7)	40.8 (28.8)	108.0 (60.0)	4.0**, df=19.6^{b}
Mean Count W	56.9 (70.9)	20.7 (37.6)	97.8 (78.4)	3.5**, df=19.5^{b}
Mean Count E	1.4 (1.5)	1.3 (1.0)	1.5 (2.0)	0.5, df=19.5^{b}
Mean Count R	0.2 (0.3)	0.1 (0.4)	0.2 (0.3)	0.6
Normalized Mean Count C	0.11 (0.04)	0.09 (0.04)	0.13 (0.03)	3.6**, df=26.7^{b}
Normalized Mean Count W	0.02 (0.03)	0.01 (0.02)	0.04 (0.03)	2.9**, df=21.6^{b}
Normalized Mean Count E	–^{c}	–^{c}	–^{c}	0.05
Normalized Mean Count R	–^{c}	–^{c}	–^{c}	1.1
Rate of Triggered Measures C	0.4 (0.1)	0.3 (0.1)	0.4 (0.1)	4.5**
Rate of Triggered Measures W	0.05 (0.04)	0.03 (0.03)	0.07 (0.04)	3.5**
Rate of Triggered Measures E	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	-0.1, df=21.6^{b}
Rate of Triggered Measures R	0.01 (0.02)	0.01 (0.02)	0.01 (0.01)	0.7
Triggered Category C	1 (0)	1 (0)	1 (0)	N/A
Triggered Category W	0.7 (0.4)	0.6 (0.4)	0.9 (0.3)	2.7*, df=28.1^{b}
Triggered Category E	0.4 (0.3)	0.4 (0.3)	0.4 (0.4)	-0.6, df=23.8^{b}
Triggered Category R	0.2 (0.3)	0.1 (0.3)	0.2 (0.3)	1.2
Normalized Triggered Category C	0.03 (0.02)	0.04 (0.02)	0.02 (0.01)	-3.1**
Normalized Triggered Category W	0.02 (0.01)	0.02 (0.02)	0.02 (0.01)	0.6
Normalized Triggered Category E	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	-1.1
Normalized Triggered Category R	–^{c}	–^{c}	–^{c}	0.2

* $p<0.05$, ** $p<0.01$. ^{a} Unless otherwise stated, df=30.

^{b} Levene's test for equality of variance resulted with a significant result, hence equal variances not assumed. ^{c} Value < 0.01.

Pay attention to the opposite direction difference between students and teachers in *Normalized Triggered Category C*. This is a direct result of *Triggered Category C* getting a 1-value for both students and teachers and of *Number of Statements* being larger for students that it is for teachers (*Normalized Triggered Category C* is a ratio of these two variables).

3.1.3 Security Features

Overall, both teachers and students showed low levels of implementing security mechanisms, as summarized in Table 3. Both implemented no security mechanism regarding *Anti-Spoofing Mechanisms* and *Sensitive Data Encryption*. As for *Input Validation* and *Checking for Errors* – on average, students statistically significantly implemented more mechanisms than teachers regarding these features. It might be that teachers, learning from their own fresh experience, emphasized these subjects to their students.

As for *Client-Side-Only Security*, recall that a 0-value for this feature denotes a proper security implementation. As seen in Table 3, teachers' mean value for this feature was 0; however, as they had barely implemented any security mechanism, this value cannot be interpreted as a good practice. The students, with relatively a high mean value (0.5), demonstrate poor security design that is focused mostly at the client-side.

Table 3. Descriptive statistics, t-test results for *security features* (one decimal place representation unless mean<0.1)

Variable	Mean (SD) N=32	Mean (SD), Teach. N=17	Mean (SD), Stud. N=15	t^{a}
Input Validation	0.06 (0.17)	0 (0)	0.13 (0.23)	2.3*, df=14.0
Anti-Spoofing Mechanism	0 (0)	0 (0)	0 (0)^{b}	N/A
Bound Checking	0.10 (0.20)	0.04 (0.12)	0.17 (0.25)	1.9, df=20.1
Checking for Errors	0.18 (0.35)	0.04 (0.12)	0.33 (0.45)	2.5*, df=15.8
Sensitive Data Encryption	0 (0)	0 (0)	0 (0)	N/A
Client-Side-Only Security	0.21 (0.42)	0 (0)^{c}	0.5^{2} (0.52)	3.3**, df=11.0

* $p<0.05$, ** $p<0.01$. ^{a} Levene's test for equality of variance resulted with a significant result, hence equal variances not assumed. ^{b} For this case, N=12. ^{c} For this case, N=16.

3.2 Hierarchical Clustering

A hierarchical cluster analysis was performed, using Ward's method for clustering by Pearson correlation. Features were standardized using Z-scores before clustering. Analysis was computed using SPSS 18. The results, presenting two clusters, are strikingly clear: One cluster (N=9) holds only teachers, the other (N=23) holds all the 15 students and 8 additional teachers.

Examining features' mean values between the two clusters adds to previous student-teacher comparison. The most striking difference is in refactor (R) features, which did not show up earlier: a) *Rate of Triggered Measures R*, with t(df=20.5)=2.2, at p<0.05; b) *Triggered Category R*, with t(df=24.4)=2.2, at p<0.05; and c) *Normalized Triggered Category R*, with t(df=20.0)=2.4, at p<0.05. Levene's test for equality of variance resulted with significant results, hence equal variances were not assumed. Means in the teachers-only cluster were lower than in the mostly-students cluster (i.e., the teachers had demonstrated better security design). Hence, it might be that teachers are more experienced than students in regulating their own programming and recognizing seemingly-suspicious code.

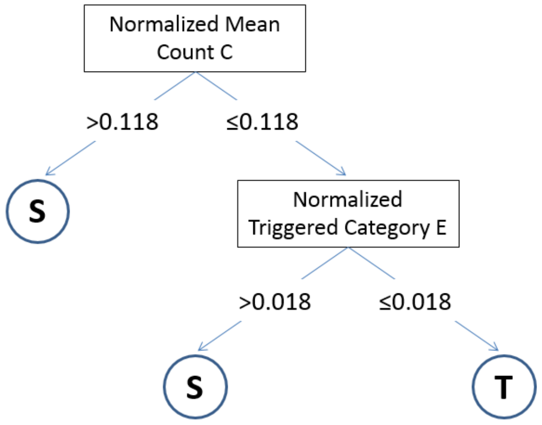
Bound Checking was also found significantly different between the two clusters, with t(df=19.1)=2.2, at p<0.05 (here also, equal variances were not assumed as for Levene's test significant result). Mean value for the teachers-only cluster is lower than the mostly-students cluster, in line with previous findings.

Some features' means were statistically significantly different when compared between teachers and students, but not different when comparing between clusters: *Number of Lines*, *Number of Statements*, and *Normalized Triggered Category C*. As *Normalized Triggered Category C* is the ratio of *Triggered Category C* – for which all of the participants got a value of 1 – to *Number of Statements*, and as *Number of Statements* and *Number of Lines* are highly correlated– with Pearson's r=0.983, at p<0.01 – it is enough to look at *Number of Statements*; therefore, we might conclude that the original difference in *Number of Statements* might have been arbitrary.

3.3 Prediction Model

Finally, we built a classifier at the code-level, predicting whether a program was submitted by a student or a teacher. 87 source codes were used. We ran a Decision Tree algorithm, using RapidMiner 5.3 (default parameters), with a manual forward feature selection. The best model found (with LOOCV kappa=0.751) is relatively simple, having only two features – *Normalized Mean Count C*, and *Normalized Triggered Category E* – three leaves and a total height of two (see Figure 1). It highlights the already known difference in convention violations between teachers and students. However, it adds an interaction of a convention feature with an error-related feature; the latter did not show up earlier. This interesting result suggests that students and teachers that are relatively good in convention-keeping might still pay attention differently to probable bugs.

Figure 1. Best prediction model (S=Student, T=Teacher)



4. DISCUSSION

Overall, we found that the teachers did better than students with regards to software quality metrics of a new programming language. However, the very existence of violations/errors in these metrics may hint that the teachers had struggled with the new material just like novices do. These findings support preliminary findings about computer science teachers being "regressed experts" when coping with new material [4]. Supporting computer science learners in improving their code might be relatively easily, by measuring software quality and security while writing the code and enabling a contextual feedback; this might produce a better code and, more importantly, a better learning [cf. 6, 8]. Popular IDEs (Integrated Development Environments) already provide integration with tools like Pylint (e.g., Emcas, VIM, Eclipse, Komodo, WingIDE, and gedit), so using such software might ease the measuring task.

As our results suggest, codes with higher software quality are not necessarily better secured. Overall, teachers' codes were of higher quality comparing to the students' codes, however with regards to the measurable security features – the opposite was true. If we want future software engineers to implement appropriate security mechanisms, we need to educate them in secure programming while teaching them programming practices.

5. REFERENCES

[1] Bateson, A.G., Alexander, R.A., and Murphy, M.D. 1987. Cognitive processing differences between novice and expert computer programmers. *Int. J. Man-Machine Studies, 26*(6), 649-660.

[2] Blikstein, P. 2011. Using learning analytics to assess students' behavior in open-ended programming tasks. *In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (Banff, AB)*, 110-116.

[3] Boehm, B.W., Brown, J.R., and Lipow, M. 1976. Quantitative evaluation of software quality. *In Proceedings of the 2nd International Conference on Software Engineering (San Francisco, CA)*, 592-605.

[4] Liberman, N., Ben-David Kolikant, Y., and Beeri, C. 2012. "Regressed experts" as a new state in teachers' professional development: lessons from Computer Science teachers' adjustments to substantial changes in the curriculum. *Computer Science Education, 22*(3), 257-283.

[5] McCall, J.A., Richards, P.K., and Walters, G.F. 1977. *Factors in software quality*. General Electric Company, Technical Report RADC-TR-77-369.

[6] Truong, N., Roe, P., and Bancroft, P. 2005. Automated feedback for "fill in the gap" programming exercises. *In Proceedings of the 7th Australasian Computing Education Conference, (Newcastle, NSW, Australia)*, 117–126.

[7] Vihavainen, A., Luukkainen, M., and Kurhila, J. 2013. Using students' programming behavior to predict success in introductory mathematics course. *In Proceedings of the 6th International Conference on Educational Dada Mining (Memphis, TN)*, 300-303.

[8] Wang, T., Su, X., Ma, P., Wang, Y., and Wang, K. 2011. Ability-training-oriented automated assessment in introductory programming course. *Computers & Education, 56*(1), 220-226.

[9] Wiedenbeck, S. 1985. Novice/expert differences in programming skills. *Int. J. Man-Machine Studies, 23*(4), 383-390.

Testing the Multimedia Principle in the Real World: A Comparison of Video vs. Text Feedback in Authentic Middle School Math Assignments

Korinn S. Ostrow
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
(508) 373 - 2676
ksostrow@wpi.edu

Neil T. Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
(508) 831 - 5569
nth@wpi.edu

ABSTRACT

As technology evolves and design options for web-based homework support systems expand, researchers are left with questions regarding best practices. These platforms often provide students correctness feedback meant to guide learning and offer dynamic tutoring to help students solve difficult problems. Feedback typically consists of bland text and worked examples, but as hypermedia gains prevalence, researchers are turning their focus to the appropriate use of such elements in e-learning environments. The following study assesses the effects of feedback medium within a randomized controlled trial conducted using ASSISTments, an adaptive math tutor. Results suggest that video feedback enhances learning outcomes and is well perceived by student users. These findings are of particular interest to the Learning Sciences, with intent to optimize e-Learning design.

Keywords

E-Learning, Cognitive Theory, Multimedia Principles, Feedback, Adaptive Tutoring, ASSISTments, Randomized Controlled Trial.

1. INTRODUCTION

A leader in the field of e-Learning, Richard Mayer has defined various multimedia principles for the optimal design of technology supported learning environments such as web-based homework support systems [3]. Rooted in cognitive theory, these principles call for the design of learning environments that are driven by an active learning process and that take the restraints of cognitive load and working memory into consideration [3][7]. Still, researchers seeking to enhance student engagement, motivation, and persistence, they are left questioning how to optimize the learning environment without overloading learners.

Mayer also posits that learners utilize separate information processing channels to internalize information; under the *redundancy principle*, material offered through one channel (i.e., a narrated passage) should not be simultaneously presented through another (i.e., text accompanying the narration) [3]. When such circumstances occur, the learner's attention is split across redundant content, depressing intake from both channels and hampering learning. Further, the *modality effect* suggests that learning gains are greater for narrated content than for content presented as text [7]. Based on these principles, the use of video, when presented without redundant textual explanation should appeal to both auditory and visual processing channels without risking overload.

Video is not novel to education, and it is growing increasingly popular due to the concept of the "flipped classroom," which often parallels the use of web-based homework support systems. While the quality of evidence for the flipped classroom has not yet proven impressive [4], the trend speaks to the growing accessibility of technological resources in education. Self-recorded video lectures and feedback offer teachers the opportunity to be deeply involved in student learning while simultaneously enhancing ownership of the technology [5].

Contrary research has suggested that video is not universally successful in promoting learning gains. In his early work on the effect of educational movies, Pane [9] noted mixed results as a function of content, offering evidence that the use of video may improve the speed of immediate recall, yet potentially harm long-term learning. Negative effects of video may include prolonged time-on-task that potentially leads to boredom or frustration, the inability to appropriately convey abstract content material, and the likelihood of technological difficulties that prevent students from adequately accessing materials.

In the present study, the ASSISTments platform is used to compare the delivery methods of feedback messages within a mathematics e-Learning environment. Prior research has found that dynamic graphics are more effective than static graphics in mathematics realms [7], and thus, we hypothesize that video will have a positive effect on learning gains within this system. Since its inception, ASSISTments has delivered significant results surrounding the use of textual feedback in the form of scaffolding and hints [10][11][12]; the present study serves as a preliminary exploration into replacing textual feedback with video.

Thus, we pose the following research questions:

1. Are learning outcomes enhanced when scaffold feedback is delivered using video rather than text?
2. Can we determine if students disproportionately internalize feedback based on the medium, given next question performance and response time?
3. Based on self-report measures, do students respond positively to the addition of video to their assignment?

2. METHODS

2.1 Participants

A set of six questions requiring students to use the Pythagorean theorem was assigned to 139 8th grade students using

ASSISTments. This student population was comprised of four classes of differing skill levels that spanned four suburban middle schools in Massachusetts and Ohio. All students were familiar with ASSISTments and used the system on a regular basis as part of classwork and homework assignments.

2.2 Design

The Pythagorean theorem problem set was derived from pre-existing ASSISTments certified material, based on Common Core State Standards and chosen in an attempt to match 8th grade fall curriculum. The structure of the problem set relied on three questions with text feedback (A, B, C) and three isomorphic questions with video feedback (A*, B*, C*). Each question and its morph were of similar difficulty and were therefore considered interchangeable (i.e., A and A*). The questions are available at [8] for further comparison.

The fixed question patterns depicted in Table 1 were rooted in the intention to allow all students an equivalent opportunity to experience both feedback styles. Thus, the four groups were designed to house fixed question patterns from which we could assess the impact of video versus text at various points throughout the problem set. Random assignment was attained by allowing ASSISTments to allocate students into one of the four groups at the start of the assignment. As depicted in Table 1, students assigned to Group 1 received video feedback if they answered question 1 incorrectly, text feedback if they answered question 2 incorrectly, and so on.

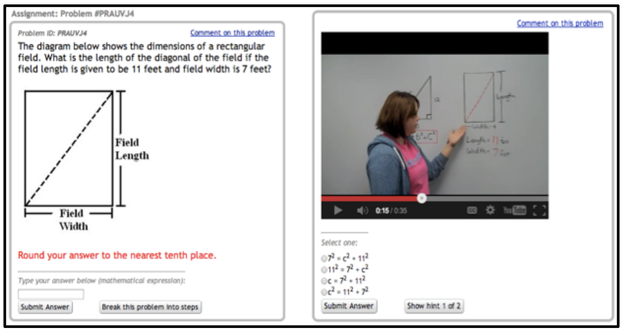
Table 1. Group design

Linear Order	1	2	3	4	5	6
Group 1	A*	B	C*	A	B*	C
Group 2	A	B*	C	A*	B	C*
Group 3	A*	B*	C	A	B	C*
Group 4	A	B	C*	A*	B*	C

*Depicts question morph with video feedback

Video content was designed to mirror textual feedback in an attempt to provide identical assistance through both mediums. Each video simply featured the lead researcher reading a feedback message while referring to the question content on a whiteboard. Figure 1 depicts question C* with video feedback, while Figure 2 depicts the question morph (C) with text feedback. All video material can be accessed at [8].

Both types of feedback were set to load incrementally with incorrect responses or if the student requested to break the question down into steps. Videos were set to play automatically, allowing students to gain information with equal efficiency regardless of feedback medium, and perhaps making it harder or more inconvenient to "game the system," or click through the scaffold steps in rapid succession.


Assignment: Problem #PRAUVJ4

Problem ID: PRAUVJ4

The diagram below shows the dimensions of a rectangular field. What is the length of the diagonal of the field if the field length is given to be 11 feet and field width is 7 feet?

Round your answer to the nearest tenth place.

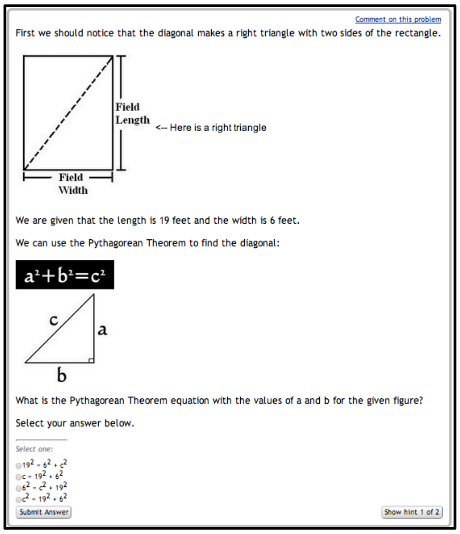
Type your answer below (mathematical expression):

Select one:

○ $7^2 = c^2 + 11^2$
○ $11^2 = 7^2 + c^2$
○ $c = 7^2 + 11^2$
○ $c^2 = 11^2 + 7^2$

Figure 1. Video feedback for question C*

For each group, four post-test survey questions asked students to judge, using a simple three-measure Likert scale (i.e., not at all, somewhat, a lot), if they felt video feedback was helpful, if it was enjoyable, if they would prefer similar videos in future assignments, and what effect video feedback had on their focus. For the entire student experience, see [8].


First we should notice that the diagonal makes a right triangle with two sides of the rectangle.

We are given that the length is 19 feet and the width is 6 feet.

We can use the Pythagorean Theorem to find the diagonal:

What is the Pythagorean Theorem equation with the values of a and b for the given figure?

Select your answer below.

Select one:

○ $19^2 = 6^2 + c^2$
○ $c = 19^2 + 6^2$
○ $6^2 = c^2 + 19^2$
○ $c^2 = 19^2 + 6^2$

Figure 2. Text feedback for question C

2.3 Procedure

The problem set was assigned to students in the manner consistent with their teacher's usual use of ASSISTments (i.e., as either classwork or homework). Students were free to work at their own pace and were not required to complete the assignment in one sitting. Log data was compiled for each student's performance, including elements such as first action, correctness, response time, attempts, and hints requested. Delegating random assignment to the tutor produced results that were less than optimal, as significantly fewer students were assigned to Group 2 and Group 4. However, assessment of the code controlling ASSISTments' random assignment function concluded that this anomaly was not influenced by any student attribute or system characteristic.

Table 2 explains initial group assignment and the process for excluding students from analysis. A total of 139 students were originally assigned (OA) the problem set. Six students failed to log enough progress to initiate a group assignment and were therefore excluded. Of the remaining 133 students, 13 students did not complete the problem set (I), and 31 students tested out (TO) (these students answered each question correctly and failed to receive feedback of either style). A disproportionate number of students tested out of Group 3, likely as a function of random assignment and small sample size.

Table 2. Students excluded from analysis

	OA	I	TO	G	**Remaining**
Group 1	35	4	7	0	**24**
Group 2	29	3	6	4	**16**
Group 3	43	4	11	2	**26**
Group 4	26	2	7	4	**13**
Total	**133***	**13**	**31**	**10**	**79**

*Six students failed to initiate a condition.

In prior research, "gaming the system" within ASSISTments has been defined as consistent answer seeking behavior displayed in rapid succession (i.e., clicking through all hints or scaffolds for completion) [1]. As such, "gamers" were operationalized as any student who clicked through question A (or A*) and its four scaffolds, regardless of feedback medium, at a rate faster than five seconds per response. By this loose definition, a total of 10 students qualified as "gamers" (G) and were removed prior to analysis as shown in Table 2.

Our primary analysis assessed student performance on the second question as a function of the feedback medium they experienced after incorrectly answering the first question. For question 1, Groups 1 and 3 were presented video feedback (A*), while Groups 2 and 4 were presented text feedback (A). We were therefore able to collapse these groups when analyzing second question performance. Based on Table 2, the removal of gamers significantly differs when the Groups are collapsed: for Groups 1 and 3, only 7.1% of students are removed from the remaining sample, while Groups 2 and 4 lose 43.5% of the remaining students. Considering our operational definition of gamers, and noting that Groups 2 and 4 received text feedback upon incorrectly answering question 1, the discrepancy found here suggests that video feedback may deter gaming. To better understand this bias, the proceeding analysis is carried out both with and without gamers for comparison.

3. RESULTS

3.1 Second Question Analysis

After considering the aforementioned exclusion methods, 79 students were remaining for analysis (89 when gamers were included). To address our initial research question, we assessed second question performance in students who had received feedback on question 1, as summarized in Table 5. Learning outcomes were enhanced for students who received video feedback (M = 0.77, SD = 0.43) rather than text feedback (M = 0.63, SD = 0.50), approaching significance at p = 0.143, with an effect size[1] of 0.32, 95% CI [-0.28, 0.91]. When gamers were included to analyze the effect of the selection bias, the improvement for students who had received video (M = 0.76, SD = 0.44) versus text (M = 0.52, SD = 0.51) became statistically significant, p < .05, with an effect size of 0.50, 95% CI [-0.03, 1.03].

Table 5. Summary of second question analysis

	Video	N	Text	N	ES[1]	95% CI
Performance						
w/o Gamers	0.77 (0.43)	35	0.63 (0.50)	16	0.32	[-0.28, 0.91]
w/ Gamers	0.76 (0.44)	37	0.52 (0.51)	23	0.50*	[-0.03, 1.03]
Resp. Time						
w/o Gamers	134.86 (118.76)	35	421.77 (1122.27)	16	-0.45	[-1.05, 0.15]
w/ Gamers	129.72 (117.46)	37	307.33 (943.50)	23	-0.30	[-0.82, 0.23]

Note. Time is depicted in seconds as: mean (standard deviation). *p < .05.

Further analysis of second question performance suggested that response time was faster for students who had received video (M = 134.86, SD = 118.76) rather than text (M = 421.77, SD = 1122.27), approaching significance at p = 0.068, with an effect size of -0.45, 95% CI [-1.05, 0.15]. When gamers were included for comparison, students who had incorrectly answered the first question and received video feedback performed faster (M = 129.72, SD = 117.46) than those receiving a text scaffold (M = 307.33, SD = 943.50), but results were not significant and the effect size dropped to -0.30, 95% CI [-0.82, 0.23]. As gaming was defined as rapidly clicking through questions and feedback, it is not surprising that time measures would drop in this manner. While these results portray consistent trends approaching significance, they should be taken with caution, as the number of students who received text feedback was disproportionately smaller than the number of students who received video feedback.

3.2 Response Time Within Feedback

To address our second research question, we examined students' overall experience within each type of feedback. Students saw a total of 186 scaffold levels of video feedback, and 171 scaffold levels of text feedback while completing their assignment. On average, response time during video feedback (M = 202.51, SD = 337.99) was longer than response time during text feedback (M = 35.18, SD = 28.74) approaching significance at p = 0.085, with an effect size of 0.68, 95% CI [0.47, 0.90]. When gamers were included for comparison, students saw a total of 241 levels of video feedback, and 231 levels of text feedback. Average response times dropped within both feedback styles, yet response time during video feedback remained longer (M = 169.28, SD = 268.44) than response time during text feedback (M = 28.38, SD = 21.67), approaching significance at p = 0.076, with an effect size of 0.73, 95% CI [0.54, 0.92].

These results suggest that there was no significant difference in the overall number of feedback levels experienced by students as a function of feedback medium. On average, students spent 3 minutes and 23 seconds within video feedback and only 35 seconds within text feedback. When gamers were considered, less time on average was spent within each feedback style, with students spending 2 minutes 49 seconds within video feedback and only 28 seconds within text feedback. Thus, students consistently spent more time within video feedback, suggesting that they actually took the time to watch the videos and internalize the content whereas they seemed to gloss over text feedback.

3.3 Survey Response Analysis

Of all students available for analysis, 53 answered the four post-test survey questions. Student responses are proportioned in Table 8. Taken together, we consider the survey results to suggest that video feedback is well perceived by students. Essentially, 83% of students reported that they would at least somewhat prefer ASSISTments to use video more often. Coupled with the student performance findings discussed above, we feel that video may be a beneficial tool for ASSISTments and that further exploration regarding the long-term effect on learning is required.

Table 8. Student responses to post-test survey questions

	Not at all	Somewhat	A lot
How helpful did you find the videos as you completed your assignment?	14%	43%	43%
How much did you enjoy the videos?	17%	57%	26%
Would you like it if more of your ASSISTments assignments used videos?	17%	43%	40%
Did you feel more focused on your assignment when the question had videos?	30%	38%	32%

[1] Effect sizes are reported throughout using Hedges correction [2].

4. CONTRIBUTIONS AND DISCUSSION

Although Mayer's work has been a predominant influence on the field of multimedia infused learning, much of his research has assessed college undergraduates in psychology labs. Thus, his results suggest a massive and seemingly unrealistic effect when compared to most real-world educational interventions. According to recent research detailing average effect sizes in educational settings, Lipsey, et al. [6] note that at the middle school level, researcher developed studies with specialized topics tend to show strength with effect sizes of approximately 0.43. The present study is on par with this trend, with effect sizes for second question analysis ranging from 0.32 to 0.51. We argue that these results provide a contribution to the Learning Sciences and help establish a basis for future research.

Based on our findings, we feel that video feedback may be a significantly beneficial tool for e-Learning. Immediate learning gains, represented by second question performance after receiving feedback on question 1, were significantly greater in students who experienced video feedback. Our results suggest that the use of video forces the learner to slow down and internalize the concept that is being taught, as depicted by consistent trends for response times within the feedback experience. Although text feedback consistently provides a faster alternative for skilled readers, perhaps adaptively slowing the pace more closely mimics the actions of a human tutor.

It should be noted that video feedback appears to have deterred gaming behavior. This may have been due in part to novelty, but was likely a function of the automatic nature of video playback. When a student tried to game through a question, each scaffold level would present another video until they were all playing simultaneously. A slightly more qualitative inspection of gaming behavior within this problem set suggested that at least three of the students labeled as gamers corrected their behavior after being exposed to video feedback. Future research is required to determine if video feedback provides a beneficial intervention for this population in general.

Regardless of the cause, let us assume for a moment that these effects are valid and reliable, and that video feedback significantly enhances student performance. With the growing popularity of web-based homework support systems and the ubiquitous nature of video servers such as YouTube and SchoolTube, teachers and instructional designers may be overlooking a valuable tool. The videos used in this study were of low production quality, shot in a single take, and featured a non-professional actress reading from a script. Teachers with years of expertise in providing feedback could arguably record a short video on their smartphone or tablet that would outperform the content used in this study. The use of video within e-Learning environments has the potential to streamline the process of repetitive one-to-one tutoring and boost the teacher's efficiency in the classroom. While pedagogical agents have become a popular tool for feedback delivery within e-Learning environments, the same messages may carry significantly more power when delivered by the student's teacher. A multitude of brief interactions offering personalized and appropriately timed feedback, guidance, and motivation, could become an important step toward truly adaptive tutoring.

Future implementations of this study should utilize a more powerful pre/post-test design with additional far transfer items and the use of open-ended survey response options to gauge student feedback. We also suggest that future endeavors compare a purely video condition to a control containing only textual feedback, perhaps using an AB design with multiple content topics to maintain fair treatment. Future work should also attempt to pinpoint critical elements driving the effects of video, such as motivation, novelty, personalization, and engagement.

5. ACKNOWLEDGEMENTS

We acknowledge funding from NSF (#1316736, 1252297, 1109483, 1031398, 0742503), ONR's "STEM Grand Challenges," and IES (#R305A120125, R305C100024). Thanks to S.D.O. & L.P.B.O.

6. REFERENCES

[1] Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A. & Koedinger, K. 2008. Why students engage in "Gaming the System" behavior in interactive learning environments. *Journal of Interactive Learning Research*. 19, 2, 185-224.

[2] CEM. 2013. Effect Size Calculator. Centre for Evaluation & Monitoring, Durham University. Accessed 11/8/2013 at https://tinyurl.com/nt4snvo.

[3] Clark, R.C. & Mayer, R. E. 2003. *e-Learning and the science of instruction: proven guidelines for consumers and designers of multimedia learning.* San Francisco, CA: Pfeiffer.

[4] Goodwin, B. & Miller, K. 2013. Research says/evidence on flipped classrooms is still coming in. *Educational Leadership: Technology-Rich Learning*. 70, 6, 78-80.

[5] Kelly, K., Heffernan, N., D'Mello, S., Namias, J., & Strain, A. 2013. Adding teacher-created motivational video to an ITS. In Proceedings of the Florida Artificial Intelligence Research Society Conference (FLAIRS 2013). 503-508.

[6] Lipsey, M.W., Puzio, K., Yun, C., Hebert, M. A., Steinka-Fry, K., Cole, M.W., Roberts, M., et al. 2012. Translating the statistical representation of the effects of education interventions into more readily interpretable forms. *National Center for Special Education Research*. Washington, DC.

[7] Mayer, R.E. (Ed). 2005. *The Cambridge handbook of multimedia learning*. New York: Cambridge University Press.

[8] Ostrow, Korinn. 2013. Pythagorean Theorem Math Study. Accessed 11/8/2013.
   a. Student Experience by Group, Randomized Controlled Trial, Data: https://tinyurl.com/lr529jp
   b. Video Scaffolds: https://tinyurl.com/mcc4w8z

[9] Pane, J.F. 1994. Assessment of the ACSE science learning environment and the impact of movies and simulations. Carnegie Mellon University, *School of Computer Science Technical Report CMU-CS-94-162*, Pittsburgh, PA.

[10] Razzaq, L. & Heffernan, N.T. 2006. Scaffolding vs. hints in the ASSISTments system. In Ikeda, Ashley & Chan (Eds). *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems*. Springer-Verlag: Berlin. 635-644.

[11] Singh, R., Saleem, M., Pradhan, P., Heffernan, C., Heffernan, N., Razzaq, L. Dailey, M. O'Connor, C. & Mulchay, C. 2011. Feedback during web-based homework: the role of hints. In Biswas et al. (Eds). *Proceedings of the Artificial Intelligence in Education Conference*. 328–336.

[12] Wang, Y. & Heffernan, N. 2011. The "assistance" model: leveraging how many hints and attempts a student needs. In Proceedings of the *Florida Artificial Intelligence Research Society Conference* (FLAIRS 2011).

The Importance of Grammar and Mechanics in Writing Assessment and Instruction: Evidence from Data Mining

Scott Crossley
Georgia State University
34 Peachtree Ave, Ste 1200
Atlanta, GA 30303
01+404-413-5179
scrossley@gsu.edu

Kris Kyle
Georgia State University
34 Peachtree Ave, Ste 1200
Atlanta, GA 30303
01+404-413-5200
kkyle3@gsu.edu

Laura Allen
Arizona State University
PO Box 872111
Tempe, AZ 85287
01+404-413-5200
LauraKAllen@asu.edu

Danielle S. McNamara
Arizona State University
PO Box 872111
Tempe, AZ 85287
01+404-413-5200
dsmcnamara1@gmail.com

ABSTRACT

The current study examined relationships between expert human judgments of text quality and grammar and mechanical errors in student writing. A corpus of essays (*N* = 100) written by high school students in the W-Pal system was collected, coded for grammar and mechanical errors, and scored by expert human raters. Results revealed weak relations between grammar errors and holistic essay scores and stronger relations between mechanics and holistic essay scores. Implications for essay scoring algorithms and providing feedback to writers are discussed.

Keywords

Intelligent tutoring systems, grammar and mechanics, automated feedback, automatic essay scoring

1. INTRODUCTION

The Writing Pal (W-Pal; [7, 9, 10]) is an intelligent tutoring system (ITS) that provides students with instruction and game-based practice on how to use writing strategies. The system also gives students opportunities to write essays, receive automated feedback on these essays, and revise the essays. The purpose of this study is to examine the importance of errors in grammar and mechanics (e.g., punctuation and spelling) for predicting holistic scores of essay quality and how the relationship between grammar and mechanics and essay quality can be used to help develop instructional modules and feedback algorithms within W-Pal. Our particular interest in the consequences of considering grammar and spelling in instructional modules and in providing automated feedback to students stems primarily from concerns expressed by writing instructors who have used W-Pal in their classes. Currently, W-Pal focuses on providing students with feedback that centers on using strategies to more effectively compose essays, including strategies to plan essays, write more effective introductions, essay bodies, and conclusions, and to revise their essays. These strategies have proven successful; however, some teachers remain concerned that students primarily need feedback on lower level aspects of writing such as grammar, punctuation, and spelling.

Although research supports the teaching of mechanics to students [5, 8], meta-analyses of effective writing instruction have demonstrated that grammar instruction is among the least effective types of student interventions [6]. On the other hand, teachers report that correct grammar and mechanics are important elements of writing instruction and writing quality. For example, in a study by Cutler and Graham [3], over 75% of surveyed teachers indicated that they taught grammar skills at least several times a week at the expense of teaching essay writing, planning, and revising. Additional evidence for the perceived importance of grammar skills in the classroom can also be found in writing textbooks, which dedicate large sections to grammar instruction [8].

Our main design and pedagogical questions in the context of W-Pal are whether to include a module that explicitly teaches grammar and mechanics, whether to provide grammar and mechanics feedback to the students who use W-Pal, and whether to incorporate grammar and mechanic indices in our automatic scoring algorithms. Fully answering these questions will likely require behavioral or intervention studies. However, an initial step in assessing the importance of grammar and mechanics is to use data mining techniques to assess relationships between grammar and mechanical accuracy and essay quality. Thus, in this study, we examine a corpus of essays written by students who were provided instruction in W-Pal. The essays were scored by expert raters for overall essay quality as well as grammatical and mechanical accuracy. The essays were also coded for grammatical and mechanical accuracy by a separate set of expert raters. We specifically seek to address the following three research questions:

1. To what extent are expert analytic scores of grammar and mechanics related to holistic scores of essay quality?
2. To what extent are expert analytic scores of grammar and mechanics associated with the number and type of errors observed in an essay?
3. To what extent are expert scores of holistic essay quality associated with the number and type of errors observed in an essay?

Our underlying presumption is that the answers to these questions will enhance our understanding of essay writing and expert judgments of essay quality. In turn, these answers will aid in the design and development of W-Pal by providing information about the importance of grammar and mechanical errors in assessing writing quality. If grammar and mechanical errors are important indicators of writing quality, then there may be value in providing instructional modules that help students avoid making grammatical and spelling errors, in providing feedback to learners about the number and types of errors that occur in their writing, and in including automated measures of grammar and mechanics in the scoring algorithms used by W-Pal. The results of the study will also strengthen our understanding of the linguistic features that underlie writing quality.

2. METHODS

To address the research questions for this study, a corpus of essays was hand coded to identify grammar and mechanical errors and these errors were then regressed onto the expert ratings of grammar and mechanics and the expert judgments of essay quality. In addition, correlations were conducted between the expert ratings of grammar and mechanics and the holistic judgments of essay quality.

2.1 Corpus

We selected 100 essays from an on-line writing study conducted in the W-Pal ITS. The essays were written by public high school students in the metro Phoenix area. The students ranged in age from 14 to 19 and the majority of the students in the study (62%) were female; 56% of the students identified themselves as native speakers of English, with the remaining participants identifying themselves as non-native speakers of English. Participants attended 10 sessions (1 session/day) over a 2-4 week period. Participants wrote a pretest essay during the first session and a posttest essay during the last session. The essays were written on two prompts (on the value of competition and on the role of images/appearances). The prompts were counterbalanced across the pretest and posttest essays. The essays used in this study were selected from the pretest essays only.

Two expert raters with at least 4 years of experience teaching freshman composition courses at a large university rated the quality of the essays using a standardized SAT rubric and an analytic rubric that contained four subsections: introduction, body, conclusion, and correctness (see [2] for more details on the rubric). The correctness subsection consisted of one rating that asked reviewers to judge an essay's grammar and mechanical accuracy. Both the SAT and the analytic rubric generated a rating with a minimum score of 1 and a maximum of 6. Raters were informed that the distance between each score was equal. The raters were first trained to use the rubric with 20 similar essays taken from another corpus. The final interrater reliability for all essays in the corpus was $r > .70$. The mean score between the raters was used as the final value for the quality of each essay. The essays selected for this study had a scoring range between 1 and 4.5. The mean score for the essays was 2.9 and the median score was 3.0. The scores were normally distributed.

2.1 Hand-Coding of Errors

An error tag-coding scheme was developed to investigate the grammar, mechanics, word use, and spelling in the 100 selected essays. The coding scheme was based on an error-tagging manual reported in Dagneaux, Dennes, Granger, and Meunier [4]. The manual consists of subsections related to form (spelling and morphology), grammar (nouns, adjectives, and verbs), lexico-grammar (complementation, dependent prepositions), lexical choices (single, phrases, connectors, and conjunctions), and word problems (redundant and missing words). Two expert raters were trained on this manual. After reviewing a training set of essays and the manual, new codes were incorporated that related to punctuation, spelling, sentence fragments, and ambiguous referents. These codes were not available in the original coding scheme but errors in the essays necessitated them. After training was completed, the raters coded each essay independently and codes between raters were compared. Differences in coding were adjudicated between the two raters until agreement was reached. Final raw scores were provided for each essay for each code. In addition, a score based on text length was computed (a normalized score). Component scores were calculated for all form errors (spelling and morphology), all grammar errors, all lexico-grammar errors, all lexical choice errors, all word problem errors, and all punctuation errors. Lastly, a total count of all errors in the essay was computed.

2.2 Statistical Analyses

Statistical analyses using SPSS were conducted to investigate the role that grammar and mechanics play in explaining human scores of essay quality. A correlation was calculated between holistic essay scores and expert scores of grammar and mechanics to examine links between holistic and analytic scores. A regression model was then used to assess the accuracy of the expert scores for grammar and mechanics by investigating associations between the hand-coded error counts and the expert judgments. Finally, a regression model was used to examine the associations between the hand-coded errors and the expert scores for holistic essay quality. For both regression models, a training and test approach was used. SPSS syntax does not select an exact percentage for training and test sets and thus training sets in SPSS may range from 63-71% of the corpus.

3. Results

3.1 Expert Scores

A correlation was calculated between the expert ratings for the holistic score and the expert ratings for grammar and mechanics (the analytic score). The resulting correlation, $r(100) = .388$, $p < .001$, reflects a positive, medium effect between the holistic and analytic scores.

3.2 Grammar Scores

Correlations were calculated between the hand-coded errors and the expert scores for grammar and mechanics to examine the strength of the relationship between these two variables. Prior to this analysis, the hand-coded error scores were also checked for multi-collinearity. The analyses demonstrated that there were 26 hand-coded errors that demonstrated at least a small effect size ($r > .10$) with the expert ratings and did not demonstrate strong multi-collinearity with each other (defined as $r > .90$). The majority of these variables were related to overall errors and mechanics, but not to grammar.

A stepwise linear regression analysis was conducted including the 26 hand-coded errors in which these variables were regressed onto the raters' evaluations for the 71 essays randomly selected by SPSS for the training set. The linear regression using the 23 variables yielded a significant model, $F(2, 69) = 20.980$, $p < .001$, $r = .615$, $r^2 = .378$. The test set yielded $r = .653$, $r^2 = .426$. Two variables were significant predictors in the regression: total number of errors (raw) and punctuation errors (raw). The regression model for the training set is presented in Table 1.

Table 1: Regression analysis predicting expert grammar and mechanics scores

Entry	Variable added	r	R^2
Entry 1	Total errors raw	0.572	0.327
Entry 2	Punctuation errors raw	0.615	0.378

3.3 Holistic Scores

Correlations were calculated between the hand-coded errors and the expert holistic scores to assess the strength of the relationship between errors and the holistic rating of essay quality and to check for multi-collinearity between the hand-coded errors. These analyses showed that there were 22 hand-coded errors that demonstrated at least a small effect size with the expert ratings of essay quality and did not demonstrate strong multi-collinearity with each other. The majority of the errors that demonstrated medium or close to medium effect sizes were related to spelling, punctuation, and lexical errors.

A stepwise linear regression analysis was conducted with the 22 variables in which the variables were regressed onto the raters' evaluations for the 71 essays randomly selected by SPSS for the training set. The regression model for the training set is presented in Table 2. The linear regression using the 22 variables yielded a significant model, $F(2, 69) = 8.043$, $p < .010$, $r = .435$, $r^2 = .189$. The test set yielded $r = .456$, $r^2 = .208$. Two variables were significant predictors in the regression: total number of errors normalized and logical connector errors normalized.

Table 2: Regression analysis predicting expert holistic scores

Entry	Variable added	r	R^2
Entry 1	Total errors normalized	0.350	0.122
Entry 2	Logical connector errors normalized	0.435	0.189

3.4 Post-Hoc Analysis

We conducted a post-hoc analysis in which we removed the total errors variable. We conducted this analysis to examine if, in the absence of a total error count, errors related to grammar or mechanics (or both) were predictive of essay quality. As in the previous analyses, a stepwise linear regression analysis was conducted with the remaining 21 variables from the holistic score analysis. These 21 variables were regressed onto the raters' evaluations for the 64 essays randomly selected by SPSS for the training set. The linear regression using the 21 variables yielded a significant model, $F(1, 63) = 9.601$, $p < .010$, $r = .364$, $r^2 = .132$. The test set yielded $r = .293$, $r^2 = .086$. One variable was a significant predictor in the regression: form errors normalized (i.e., errors related to spelling and morphology errors normalized for text length). The remaining 20 variables, including all of the grammar variables, did not significantly add to the model and were left out. The regression model for the training set is presented in Table 3.

Table 3: Regression analysis predicting expert holistic scores without total errors

Entry	Variable added	r	R^2
Entry 1	Form errors normalized	0.364	0.132

4. Discussion

We have taken a corpus-based data mining approach to investigating the importance of grammatical and mechanical features in predicting the quality of students' essays. The results of this study indicate that expert ratings of grammar and mechanical accuracy are positively correlated to essay score and that the total number of errors and the number of punctuation errors in an essay are predictive of human judgments of grammar and mechanical accuracy. The findings also indicate that if grammatical errors in essays have any effect on expert judgments of essay quality, they are small. In contrast, errors related to spelling, punctuation, and lexical choices showed relatively strong correlations. These findings call into question the need to design instructional modules to teach grammar in W-Pal, as well as in other tutoring systems which focus on helping students to improve their writing quality.

In reference to relations between expert judgments of essay quality and expert judgments of grammar and mechanics, the findings report a moderate correlation that explains 15% of the variance in overall essay quality. Previous studies have shown similar results for the strength of grammar and mechanic judgments to predict essay quality [1, 2]. Importantly, these studies have indicated that human ratings of grammar and mechanics are the least predictive analytic ratings of essay quality (behind analytic judgments related to text organization, perspective, unity, conviction, and other elements). The regression analysis between coded errors in essays and human judgments of grammar and mechanic errors demonstrated that total errors and punctuation errors explained 43% of the variance in the human judgments for the test set. Such a finding indicates that expert ratings of grammar and mechanics are not solely based on overt errors in essays (i.e., over 50% of the variance in these judgments are not explained by grammar, spelling, and punctuation errors in the essay).

In reference to relations between grammatical errors and overall essay quality, the strongest correlation reported for a grammatical variable (article errors) demonstrated only a small effect size with holistic scores (and one that was not significant). In total, only four grammatical errors demonstrated at least small effect sizes with holistic scores of essay quality (i.e., article errors, verb morphology errors, noun errors, and verb errors). In no instances were grammatical error variables included in regression models that predicted essay quality. Thus, the findings point toward a weakness of grammatical errors in explaining writing quality and provide little evidence to support the inclusion of a grammar instruction module in the W-Pal system or include grammar indices in the automatic scoring algorithms contained in W-Pal. Additionally, since grammar errors in the essays are not strongly linked to overall scores of essay quality, there appears to be no strong evidence to provide feedback to W-Pal users concerning grammatical errors.

Correlations between holistic scores and the hand coded errors yielded the strongest associations for spelling errors. However, only a few spelling error variables showed medium effects sizes with essay quality and only one index of combined spelling and morpheme errors (form errors) was included in a regression model that explained essay quality (this index explained 13% of the variance in essay quality). The majority of mechanical errors demonstrated only small effects with human judgments of essay quality and most of these errors did not reach significance.

Thus, while the evidence for mechanical instruction is a bit stronger, the findings do not strongly support the need to design instructional modules to teach mechanics in W-Pal. From a practical standpoint, designing a module that covers all potential spelling and punctuation errors in English is also too ambitious for a single ITS. In addition, research has demonstrated that students learn to spell best when they correct their own mispellings under the guidance of a teacher. This is especially true for students who have developed spelling skills (such as the adolescent writers targeted by W-Pal); these students should be able to predict spelling difficulties and apply previous knowledge to correct present spelling errors [11]. Therefore, the results of this study combined with design limitations and previous studies suggest that explicit spelling instruction may not be beneficial or practical.

In contrast to grammar errors, however, relations between spelling errors and holistic essay scores do appear strong enough to justify changes to the W-Pal automatic scoring algorithms and to the automatic feedback system. Automatically counting the number and types of spelling errors in an essay may improve the accuracy of the current scoring algorithm. In addition, compiling incidence scores for the number and types of punctuation normalized by the number of clauses or sentences may also increase the accuracy of the scoring algorithm. From a feedback perspective, highlighting spelling errors for W-Pal users may allow them to correct mispelled words more naturally. If, after highlighting spelling errors, users cannot still correctly spell the word, a drop-down menu with suggested spellings could be provided. In this way, spelling feedback that resembles best practices could be provided to W-Pal users. Of course such feedback mechanisms need to be assessed experimentally to better understand the relationship between spelling feedback and essay quality.

5. Conclusion

The results from this study, in combination with previous research, indicate that the explicit instruction of grammar in an ITS like W-Pal is likely unnecessary. In addition, providing feedback to users in reference to grammatical errors in their writing appears unwarranted (mostly because grammatical errors do not demonstrate strong relationships with essay quality). The same cannot be said for spelling and punctuation, which yield stronger relationships with judgments of writing quality. Thus, future versions of the W-Pal system will likely need to be sensitive to students' spelling and punctuation errors. However, we realize that the expectations of the scoring rubric used in this study may differ from the expectations found in an actual classroom and that the rubric itself may help in determining the importance of grammar and mechanics for the raters. The findings also indicate that human ratings of grammar and mechanics go beyond overt grammar, punctuation, and spelling errors as found in the text. A better understanding of what textual elements humans attend to when assessing grammar and mechanics would assist in more accurately identifying errors, which would be helpful in developing instructional techniques more strongly grounded in teacher cognition. Overall, the findings from this study provide important implications for system development and design that are based on real learning in practice. The findings also promote a number of future research areas.

6. ACKNOWLEDGMENTS

This research was supported in part by the Institute for Education Sciences (IES R305A080589 and IES R305G20018-02). Ideas expressed in this material are those of the authors and do not necessarily reflect the views of the IES. We also thank Rod Roscoe, Tanner Jackson, Erica Snow, and Jianmin Dai for their help with the data collection and analysis and developing the ideas found in this paper. We lastly thank the readers that found our misspellings of misspell and will perpetually wonder if they judged the quality of this paper differently as a result.

7. REFERENCES

[1] Crossley, S. A. & McNamara, D. S. 2010. Cohesion, coherence, and expert evaluations of writing proficiency. In S. Ohlsson & R. Catrambone (Eds.), *Proceedings of the 32nd Annual Conference of the Cognitive Science Society* (pp. 984-989). Austin, TX: Cognitive Science Society.

[2] Crossley, S. A., & McNamara, D. S. 2011. Text coherence and judgments of essay quality: Models of quality and coherence. In L. Carlson, C. Hoelscher, & T. F. Shipley (Eds.), *Proceedings of the 29th Annual Conference of the Cognitive Science Society*. (pp. 1236-1241). Austin, TX: Cognitive Science Society.

[3] Cutler, L., & Graham, S. 2008. Primary grade writing instruction: A national survey. Journal of Educational Psychology, 100, 907 – 919.

[4] Dagneaux, E., S. Denness, S. Granger & Meunier, F. 1996. *Error Tagging Manual*, Version 1.1. Center for English Corpus Linguistics. Louvain-la-Neuve: Université Catholique de Louvain.

[5] Graham, S. 1983. Effective spelling instruction. *Elementary School Journal, 83* (5), 560-567.

[6] Graham, S., & Perin, D. 2007. A meta-analysis of writing instruction for adolescent students. *Journal of Educational Psychology, 99,* 445-476.

[7] McNamara, D., Raine, R., Roscoe, R., Crossley, S., Jackson, G., Dai, J., Cai, Z., Renner, A., Brandon, R., Weston, J., Dempsey, K., Carney, D., Sullivan, S., Kim, L., Rus, V., Floyd, R., McCarthy, P., and Graesser, A. 2012. The Writing-Pal: Natural language algorithms to support intelligent tutoring on writing strategies. In P. McCarthy & C. Boonthum-Denecke (Eds.), *Applied natural language processing and content analysis: Identification, investigation, and resolution* (pp. 298-311). Hershey, P.A.: IGI Global.

[8] Morris, D., Blanton, L., Blanton, W., & Perney, J. (1995). Spelling instruction and achievement in six classrooms. *Elementary School Journal, 96,* 145–162.

[9] Roscoe, R. & McNamara, D. in press. Writing Pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*.

[10] Roscoe, R. D., Varner, L. K., Weston, J. L., Crossley, S. A., & McNamara, D. S. in press. The Writing Pal intelligent tutoring system: Usability testing and development. *Computers and Composition.*

[11] Schoephoerster, H. 1962. Research into variations of the test-study plan of teaching spelling. *Elementary English, 39*, 460-462.

The Long and Winding Road: Investigating the Differential Writing Patterns of High and Low Skilled Writers

Laura K. Allen
Tempe, AZ, USA
Arizona State University
LauraKAllen@asu.edu

Erica L. Snow
Tempe, AZ, USA
Arizona State University
Erica.L.Snow@asu.edu

Danielle S. McNamara
Tempe, AZ, USA
Arizona State University
Danielle.McNamara@asu.edu

ABSTRACT

We investigate how writing proficiency relates to the flexible use of cohesion. Forty-five students wrote 16 essays across 8 sessions. Natural language processing techniques were used to calculate the cohesion of each essay. Random walk and Euclidian distance measures were then used to visualize and classify students' flexibility in cohesion across the essays. Results revealed that students who were more flexible in their cohesion also had greater literacy skills and prior knowledge. Further, cohesive flexibility was most strongly related to the *unity* of the pretest essays.

Keywords

Intelligent Tutoring Systems, dynamical analysis, writing, flexibility, cohesion, automated essay scoring

1. INTRODUCTION

Students' ability to effectively communicate via writing has been shown to be a critical skill for academic and professional achievement. Standardized tests, for instance, typically require students to complete a single assignment that is designed to tap into their proficiency at writing. This assessment has a profound impact on college acceptance and other opportunities, such as scholarships, honors organizations, and assistantships [1].

Unfortunately, teachers do not have the time to provide thorough feedback on every essay a student generates. In response to these needs, researchers have developed adaptive computerized systems designed to assess the quality of essays [2]. Automated essay scoring (AES) systems employ natural language processing (NLP) and statistical methods to evaluate the structure, content, and holistic quality of written text [2-3]. Although the validity of these scores has been questioned [4], AES systems tend to calculate automated scores that are comparable to human scores [5].

AES systems have been recently integrated into learning environments, such as automated writing evaluation (AWE) systems [6] and intelligent tutoring systems (ITSs) [7]. These environments emphasize the provision of instruction and formative feedback based on the quality and specific characteristics of students' writing. This has presented a number of problems regarding the ability of the algorithms to provide specific and formative feedback that is beneficial to students [8].

The above categories of systems (AES, AWE, and ITSs) tend to rely on text-level features of *individual* essays to assess writing ability. Although essay scores are generally comparable to those provided by humans, they rarely incorporate information about the students themselves (e.g., their skills, affective states, etc.) into the system feedback or scoring algorithms. Additionally, the systems place little to no focus on the writing style of individual students. In other words, they do not take into consideration the possibility that high-quality essays may exhibit different textual properties across multiple writers.

Researchers have identified a number of linguistic features that are associated with writing quality [9-13]. Through the use of NLP tools, these indices can be automatically calculated and combined to develop algorithms for essay scoring. Recently, Coh-Metrix was used to examine which linguistic features were capable of discriminating between high- and low-quality essays [11]. The results revealed that high-quality essays included more diverse and novel word choices and more complex syntax. Interestingly, no indices of cohesion were related to essay scores. Crossley and colleagues (2011) conducted an analysis using similar indices to predict essay scores. In contrast to the previous results, this study found that essay quality was *positively* associated with cohesion. These mixed findings indicate that writing proficiency is a more complex and dynamic construct than previously assumed. Specifically, this skill may not be adequately captured by a single writing sample, as linguistic properties associated with essay quality vary across multiple contexts, such as writer populations, time constraints, and prompts [9,14-15].

We hypothesize that writing proficiency is associated with a *flexible* use of linguistic properties, rather than a fixed set of features. For example, certain prompts and contexts may require different levels of cohesion to effectively convey the main idea. In this case, strong writers may have the ability to assess the context of their writing task and flexibly employ different cohesive devices that match the evidence and arguments presented in that specific essay, whereas less skilled writers might not have developed the strategies necessary to vary their style across different contexts. Researchers have cited flexibility as a characteristic of strong writers [2]. However, few studies (if any) have explicitly measured writing flexibility and examined its relation to writing skills and other individual differences. We address this gap by investigating how writing proficiency relates to students' flexible use of cohesion across various prompts, and examine how individual differences relate to this flexibility.

2. METHODS

The data was collected as part of a larger study, which compared students' use of a writing strategy ITS to an AWE component of the system. We focus on the participants who engaged with the

AWE component of the system (n = 45). Students completed a 10-session experiment. During the first session, students completed a pretest. Training occurred during the following eight sessions. Throughout each training session, students wrote two essays, each on a different prompt topic. Thus, 16 training essays were collected for each student. During session 10, students completed a posttest, which was similar to the pretest.

2.1 Measures

Students' writing proficiency was assessed at pretest and posttest through the use of timed (25-minute) and counterbalanced prompt-based essays. All essays were assessed on a scale of 1-6 by two expert raters. The holistic grading rubric was based on a standardized rubric typically used for the assessment of Scholastic Achievement Test (SAT) essays. The rubric contained subscale scores, which assessed the quality of various sections of the essay. These subscales related to the following aspects of the essay: effective lead, clear purpose, clear plan, topic sentences, paragraph transitions, organization, unity, perspective, conviction, and grammar, syntax and mechanics.

Reading comprehension ability and vocabulary knowledge were assessed using the Gates-MacGinitie reading skill test. Students' prior knowledge was assessed using a measure of prior knowledge that assessed knowledge of science, literature, and history.

Coh-Metrix was used to assess the cohesion of the students' 16 essays. Coh-Metrix [16] is a computational text analysis tool that was developed, in part, to provide stronger measures of text difficulty. This tool includes *Easability Components,* which were developed to account for the multiple dimensions of text difficulty. Referential cohesion is one of the Easability Components and reflects the degree to which words and ideas overlap across a text. For each of the 16 essays, a *referential cohesion percentile score* was computed on a scale from 0 to 100.

3. QUANTITATIVE METHODS

To assess the flexibility of students' use of cohesion, we employed NLP and dynamical methodologies. The unique combination of these methodologies affords researchers a new assessment technique that can visualize and capture the degree to which students exhibit a controlled or flexible writing style.

Table 1. Referential cohesion classification and vector assignment

Essay Cohesion Level	Axis Direction Assignment
Less than 25% Referential Cohesion	-1 on X-axis (move left)
Between 25% and 50% Referential Cohesion	+1 on Y-axis (move up)
Between 50% and 75% Referential Cohesion	+1 on X-Axis (move right)
Greater than 75% Referential Cohesion	-1 on Y-axis (move down)

Random walks are mathematical analyses that provide a spatial representation of patterns that form in categorical data across time [17]. Random walks were used to visualize patterns in students' use of cohesive devices across their 16 training essays. Each essay was classified as one of four orthogonal Cohesion Level groups (see Table 1) using the Coh-Metrix *referential cohesion percentile score* (ranging from 0-100). These orthogonal categories were then assigned to individual vectors along a scatter plot. For instance, an essay that received a referential cohesion score below 25 would be assigned to the vector (-1,0), along the left side of the X-axis. Each student's random walk began at the origin. For each essay that the student wrote, the walk would move in the direction that was consistent with its assigned vector. The resulting walk depicts each student's use of cohesion across the training essays.

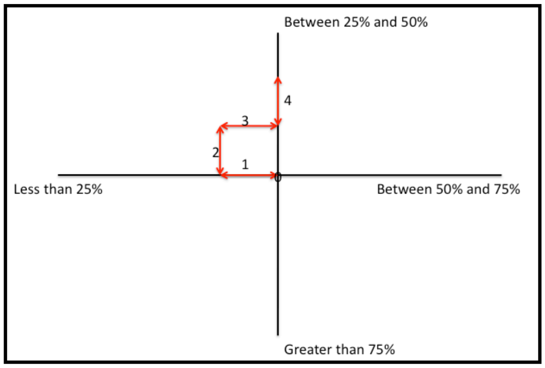


Figure 1. Example Random Walk

Figure 1 illustrates what a random walk might look like for a student who wrote 4 essays. All walks begin by placing a dot at the origin. In this example, the first essay written was low in referential cohesion (score < 25); thus, the dot moved one step left along the X-axis (#1). The next essay received a referential cohesion score between 25-50; thus, the dot moved up along the Y-axis (#2). The third essay had a referential cohesion score that ranged from 50-75, so the dot moved one step right along the X-axis (#3). The last essay received a cohesion score that ranged from 25-50; so, the dot moved one step up (#4). Using these rules, unique random walks were generated for each of the students.

To quantify the information in the random walk visualizations, distance time series were calculated for each student using Euclidian distance. Here, *y* represents the particle's position on the Y-axis, *x* represents the particle's position on the X-axis, and *i* represents the *i*th step in each student's walk.

$$\text{Distance} = \sqrt{(y_i - y_0)^2 + (x_i - x_0)^2} \qquad (1)$$

A measure of Euclidian distance was calculated for each step in a student's walk. This produced a distance time series, which reflected the degree to which students were flexible in their use of cohesion. For example, if a student used the same degree of cohesion throughout all 16 essays, that student would move far away from the origin, resulting in a high Euclidian distance score. Conversely, if a student varied a great deal in the use of cohesion, the resulting Euclidian distance for their walk would be lower, as their changes would cause them to remain close to the origin.

4. RESULTS

We calculated the average Euclidian distance of each student's walk (their *cohesion distance score*). Students varied considerably in their flexibility, ranging from a minimum cohesion distance score of 1.42 to a maximum score of 8.50 (*M* = 5.04, *SD* = 1.88). In Figure 2, each student's cohesion distance score is plotted to visualize the variation in the degree to which walks traveled.

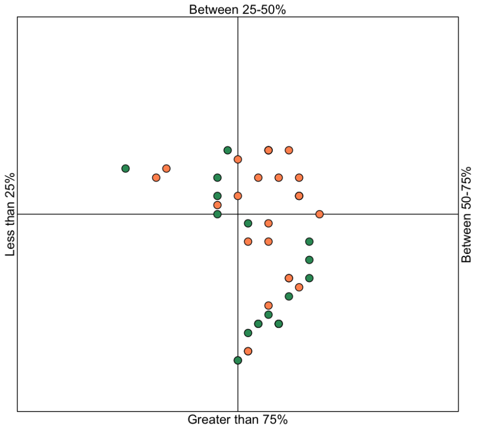


Figure 2. Visualization of Low-Skilled and High-Skilled Students' Random Walks

We next examined the degree to which cohesive flexibility varied according to students' writing proficiency. A median split was calculated on students' pretest essay scores to produce two groups: *low writing ability* and *high writing ability*. A between-subjects ANOVA revealed that high writing ability students had significantly lower cohesion distance scores (M = 4.49, SD = 1.30) compared to low writing ability students (M = 5.80, SD = 2.10), F (1, 42) = 6.28, p = .016. Figure 2 provides a visualization of these differences, with low writing ability students represented as green dots and high writing ability students represented by pink dots. As revealed in this visualization, low writing ability students (green dots) moved further from the origin than high writing ability students (pink dots) who clustered closer to the origin.

4.1 Essay Components

The correlation between cohesion distance scores and pretest holistic essay scores was marginally significant (r = -.30, p = .052), suggesting that students who were more flexible were more proficient writers. Additionally, distance scores were related to a number of the subscale scores on the writing rubric (see Table 2).

Table 2. Correlations between Cohesion Distance Scores and Rubric Subscales

Rubric Subscale	r
Lead	.02
Purpose	-.29 (M)
Plan	-.29 (M)
Use of Topic Sentences	-.27 (M)
Transitions	-.20
Organization	-.26 (M)
Unity	-.42**
Perspective	-.35*
Persuasion	-.40**
Accuracy	-.29 (M)
(M) = Marginal Significance; * = p < .05; ** = p < .01	

To determine which rubric subscale scores were most predictive of writing flexibility, we conducted a stepwise regression analysis with the significantly correlated subscale variables as predictors of cohesion distance scores. One variable was retained in the final model and predicted 17% of the variance in distance scores [F (1, 42) = 8.85, p = .005; R^2 = .17]: Unity [B = -.417, t(1, 42) = -2.98, p = .005]. Overall, these results suggested that students who produced more *coherent* and unified ideas were the students who exhibited greater flexibility in their use of cohesion, or cohesive cues.

4.2 Individual Differences

Students' cohesion distance scores were significantly (or marginally significantly) related to a number of pretest measures (see Table 3).

Table 3. Correlations between Cohesion Distance Scores and Individual Difference Measures

Individual Difference Measure	r
Reading Comprehension	-.44**
Vocabulary Knowledge	-.19
Prior Knowledge (Overall)	-.31*
Science Prior Knowledge	-.50**
History Prior Knowledge	-.11
Literature Prior Knowledge	.16
(M) = Marginal Significance; p < .05*; p < .01**	

To examine which of the individual difference measures were the most predictive of cohesive flexibility, we conducted a stepwise regression analysis including the significantly correlated variables as predictors of cohesion distance scores. One variable was retained and predicted 25% of the variance in cohesion distance scores [F (1, 43) = 14.59, p < .001; R^2 = .25]: Science Prior Knowledge [B = -.50, t(1, 42) = -3.82, p < .001]. Students who entered the writing task with greater knowledge about the world may have had an easier time adapting their writing style, as they could utilize various facts to develop their arguments.

5. DISCUSSION

One important consideration when assessing writing proficiency is the flexibility that students exert in their writing style across time. Although individual essay scores can provide valuable information about writing skills, they fail to consider the context of the writing assignments and consequently are not able to fully capture the construct of writing proficiency.

We were able to capture cohesive flexibility through the use of two novel techniques: random walks and Euclidian distances. Random walk analyses allowed us to visualize students' rigid or flexible use of cohesion across the essay assignments; additionally, it allowed us to visualize the differential patterns exhibited by high- and low-ability writers. Euclidian distance scores were then used to calculate *cohesion distance scores*. These scores confirmed the results of the random walk visualizations. In particular, they revealed that that students varied considerably in their cohesive flexibility, with low-ability students showing more consistency in their use of cohesion than high-ability students.

The results support our hypotheses and provide evidence for assumptions that have been only anecdotally raised in the writing literature [2]. Namely, they suggest that students who are more flexible in their writing style are also better writers, and vice versa. Additionally, these students outperform less flexible students on measures of literacy skills and prior knowledge. The results also suggest that cohesive flexibility was most strongly related to the *unity* (i.e., the coherence) of the pretest essay. Thus, coherence was not directly related to the presence or absence of cohesive features. Rather, students who produced more coherent essays were more flexible in their use of cohesive devices. Taken together, these results indicate that the link between textual features and writing quality may be inconsistent from assessment to assessment. Thus, more sophisticated writing assessments are needed to capture students' proficiency within various contexts.

This study extends previous work suggesting that the link between textual features and essay scores can vary across a number of contexts [9,14-15] by considering students' performance across time. Although these analyses make strong progress towards developing our understanding of writing flexibility, a number of questions remain to be answered. For instance, how does flexibility of other linguistic features (e.g., narrativity) relate to writing proficiency? Can students be trained to exhibit greater flexibility in their writing? Such analyses will help shed light on the role that flexibility plays in the development of writing proficiency. Overall, this study provides a critical insight into the complexity of automated writing evaluation and provides a novel method for providing essay scores and feedback that are more sensitive to the surrounding context of the writing assessment.

6. ACKNOWLEDGEMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080589 to Arizona State University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

7. REFERENCES

[1] Graham, S., and Perin, D. 2007. A meta-analysis of writing instruction for adolescent students. *Journal of Educational Psychology*, 99, (2007), 445-476.

[2] Dikli, S. 2006. An overview of automated scoring of essays. *Journal of Technology, Learning, and Assessment*, 5, (2006).

[3] Shermis, M., and Burstein, J. (Eds.). 2003. *Automated essay scoring: A cross-disciplinary perspective*. Erlbaum, Mahwah, NJ.

[4] Wang, J., and Brown, M. S. 2007. Automated essay scoring versus human scoring: A comparative study. *Journal of Technology, Learning, and Assessment, 6,* (2007), 46.

[5] Warschauer, M., and Ware, P. 2006. Automated writing evaluation: defining the classroom research agenda. *Language Teaching Research, 10*, (2006), 1-24.

[6] Crossley, S. A., Varner, L. K., Roscoe, R. D., and McNamara, D. S. 2013. Using automated indices of cohesion to evaluate an intelligent tutoring system and an automated writing evaluation system. In K. Yacef et al (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED).* Springer, Heidelberg, Berlin, 269-278.

[7] Roscoe, R. D., Snow, E. L., Varner, L. K., and McNamara, D. S. in press. Automated detection of essay revising patterns: Application for intelligent feedback in a writing tutor. *Technology, Instruction, Cognition, and Learning.*

[8] Roscoe, R. D., Varner, L. K., Cai, Z., Weston, J. L., Crossley, S. A., and McNamara, D. S. 2011. Internal usability testing of automated essay feedback in an intelligent writing tutor. In R. C. Murray & P. M. McCarthy (Eds.), *Proceedings of the 24th International Florida Artificial Intelligence Research Society (FLAIRS) Conference*. AAAI Press, Menlo Park, CA, 543-548.

[9] Crossley, S. A., Allen, L. K., & McNamara, D. S. in press. A multidimensional analysis of essay writing: What linguistic features tell us about situational parameters and the effects of language functions on judgments of quality. In *Multi-dimensional analysis, 25 years on: A tribute to Douglas Biber.*

[10] Witte, S. P., and Faigley, L. 1981. Coherence, cohesion and writing quality. *College Composition and Communication, 22,* (1981), 189-204.

[11] McNamara, D. S., Crossley, S. A., and McCarthy, P. M. 2010. The linguistic features of quality writing. *Written Communication, 27*, (2010), 57-86.

[12] Varner, L. K., Roscoe, R. D., and McNamara, D. S. 2013. Evaluative misalignment of 10th-grade student and teacher criteria for essay quality: An automated textual analysis. *Journal of Writing Research, 5,* (2013), 35-59.

[13] Crossley, S. A., Roscoe, R. D., McNamara, D. S., and Graesser, A. 2011. Predicting human scores of essay quality using computational indices of linguistic and textual features. In G. Biswas, S. Bull, J. Kay, and A. Mitrovic (Eds.), *Proceedings of the 15th International Conference on Artificial Intelligence in Education*. Springer, New York, 438-440.

[14] Crossley, S., Weston, J., Sullivan, S., and McNamara, D. 2011. The development of writing proficiency as a function of grade level: a linguistic analysis. *Written Communication. 28*, (2011), 282-311.

[15] Crossley, S. A., Varner, L. K., and McNamara, D. S. 2013. Cohesion-based prompt effects in argumentative writing. In C. Boonthum-Denecke & G. M. Youngblood (Eds.), *Proceedings of the 26th International Florida Artificial Intelligence Research Society (FLAIRS) Conference*. The AAAI Press, Menlo Park, CA, 202-207.

[16] McNamara, D. S., Graesser, A. C., McCarthy, P., and Cai, Z. 2014. Automated evaluation of text and discourse with Coh-Metrix. Cambridge: Cambridge University Press.

[17] Snow, E. L., Likens, A., Jackson, G. T., and McNamara, D. S. 2013. Students' walk through tutoring: Using a random walk analysis to profile students. In S. K. D'Mello, R. A. Calvo, and A. Olney (Eds.), Proceedings of the 6th International Conference on Educational Data Mining, Springer Berlin Heidelberg, 276-279.

The refinement of a Q-matrix: Assessing methods to validate tasks to skills mapping

Michel C. Desmarais
Polytechnique Montreal
michel.desmarais@polymtl.ca

Behzad Beheshti
Polytechnique Montreal
behzad.beheshti@polymtl.ca

Peng Xu
Polytechnique Montreal
peng.xu@polymtl.ca

ABSTRACT

The objective of specifying which skills are required in a given task is fundamental for the accurate assessment of a student's knowledge and for personalizing tutor interaction towards more relevant and effective assessment and learning. We compare three data driven techniques for the validation of skills-to-tasks mappings. All methods start from a given mapping, typically obtained from domain experts, and use optimization techniques to suggest a refined version of the skills-to-task mapping. To validate the different techniques, we inject perturbations in the Q-matrix and verify whether the original Q-matrix can be recovered. Tests are run over both simulated and real data. The analysis of the Q-matrix refinements of each technique over ten data sets shows that, in general, around 1/2 to 2/3 of the perturbations can be restored to their original values, but a number of potentially wrong perturbations are also introduced. The number of correctly restored and falsely switched values vary across the three techniques and between synthetic and real data. For 1 to 10 perturbations injected, simulated data recovery rate is around 2/3, and invalid alterations introduced vary around 2 to 3. For real data, the two best techniques generally recover about half the perturbations injected, but introduce between 5 and 7 alterations inconsistent with the original, expert defined Q-matrix, although some of them may be real improvements.

Keywords

Student model, Skills modeling, Psychometrics, Q-matrix, Matrix Factorization, Alternate Least-Squares, DINA

1. INTRODUCTION

Detailed assessment of skills rely on a fine grained mapping of tasks to skills. Student success and failures over these tasks provide evidence of which skills are mastered. Many intelligent tutors use such information to tailor their behaviour (for eg. [9]).

However, defining the mapping of tasks to skills is non trivial and error prone. The validation of such mapping from student test results has been the focus of recent developments in the field of psychometrics and educational data mining in recent years [3, 1, 11, 2, 6]. The ever growing abundance of student assessment traces from e-learning environments further enhances our capacity to validate expert defined mappings through data mining techniques.

This paper compares three families of techniques to refine a given mapping of skills to tasks, which we will refer to as *items*. All methods compared start with a given skill to item mapping, and typically suggest a few changes. We define a methodology to validate whether the proposed changes are appropriate. This validation rests on a number of experiments with artificial and real data to compare the quality of the changes recommended by each technique. The background work of item to skills mapping is first reviewed, followed by the description of the methodology and results of the experiments.

2. Q-MATRICES, THEIR INTERPRETATION AND VALIDATION

A mapping of item to skills is termed a Q-matrix [14]. If all specified skills are required to succeed the item, the Q-matrix is labelled **conjunctive**. If a any of the required skill is sufficient to the item's success, then it is labelled **disjunctive**. The conjunctive/disjunctive distinction is also referred to as AND/OR gates. A well known model, DINA for "Deterministic Input Noisy AND", corresponds to the conjunctive version. A variant of DINA, the DINO model (Deterministic Input Noisy OR) corresponds to a disjunctive Q-matrix [8].

Two techniques for Q-matrix validation surveyed here rely on the DINA model. A third one relies on a matrix factorization technique called ALS (Alternative Least Squares). We refer to them as (1) de la Torre (2008), (2) Chiu (2013), and (3) ALS:

(1) **de la Torre (2008)**. The method defined by de la Torre [3] searches for a Q-matrix that maximizes the difference in the probabilities of a correct response to an item between examinees who possess all the skills required for a correct response to that item and examinees who do not.

(2) **Chiu (2013)**. Chiu defines a method that minimizes the residual sum of square (RSS) between the real responses and the ideal responses that follow from a given Q-matrix [2]. The algorithm adjusts the Q-matrix by choosing the item with the worst RSS over to the data, and replaces it with the one has the lowest RSS, and iterates until convergence.

(3) **Alternate Least-Square Factorization (ALS)**. The (ALS) method is defined in [6]. Contrary to the other two methods, it does not rely on the DINA model. Instead, it decomposes the results matrix $\mathbf{R}_{m\times n}$ of m items by n students as the inner product two smaller matrices: $\mathbf{R} = \mathbf{Q}\,\mathbf{S}$,

where **R** is the results matrix, **Q** is the m items by k skills Q-matrix, and **S** is the mastery matrix of k skills by n students. The factorization consists of alternating between estimates of **S** and **Q** until convergence.

3. METHODOLOGY AND DATA SETS

The two first methods, de la Torre (2008) [3] and Chiu (2013) [2], have been shown to perform well on artificial data. On real data, their performance is more blurry. The ALS factorization method [6] has only been tested on one real data set. But the methodologies used to validate all three techniques in each respective study vary considerably and do not allow for a proper comparison.

To validate and compare the effectiveness of each technique for refining a given Q-matrix, we follow a methodology based on recovering the Q-matrix from a number perturbations: the binary value of a number of cells of the Q-matrix is inverted, and this "corrupted" matrix is given as input to each technique. If the technique recovers the original value of each altered cell, then we consider that it successfully "refined" the Q-matrix. This approach is similar to the studies mentioned [3, 2, 6].

Ten levels of perturbations are defined, from 1 to 10. For each level, we conduct up to 30 experiments that consists in choosing Q-matrix cells to be altered. If the Q-matrix contains 30 or less cells, all of them are altered in turn. If it is larger, combinations of cells are chosen at random. We refer to this procedure as a single perturbation run. The runs are repeated for each of the 10 levels of perturbation, and over the different data sets.

The measures of performance are the number of *true positives* and *false positives*:

- **Mean true positives**: a *true positive* corresponds to an alteration that was injected in the input, and was correctly switched back to its original value by the method. The measure reported is the number of correctly recovered alterations averaged over the 8 runs and by level of perturbation.
- **Mean false positive ratio**: a *false positive* corresponds to a changed Q-matrix entry returned by the method, but that was not injected in the input. Hereto, averages by perturbation runs are reported.

For real data, the definition of true/false positives rests on the assumption that the original matrix is better than the corrupted one, which is not necessarily the case with an expert generated Q-matrix. The expert may be wrong. However, we have no other means to inform us of the "real" Q-matrix and it is reasonable to assume that most of the cells in the Q-matrix are correct. Of course, for synthetic data, this assumption is correct as the Q-matrix is at the source of the generation of the data.

A total of 10 data sets are used for the validation. They are freely available from two R packages: CDM (`http://cran.r-project.org/web/packages/CDM/index.html`) [12] and Chiu (2013) (`http://cran.r-project.org/web/packages/NPCD/NPCD.pdf`). Table 1 contains a short description of each data set. Note that for the last five data sets, the source data is the same, but different Q-matrices are defined over them and subsets of items are used in the last four: the fraction data set data is used to create four variations through subsets of questions and alternative Q-matrices (Fraction 1, Fraction 2/1, Fraction 2/2, and Fraction 2/3). The artificial data sets are generated from the well known DINA and DINO models.

For obtaining the results of the de la Torre (2008) method, we used the R implementation found in the CDM package [12]. A DINA model and parameter estimation is first built with the default arguments to the `din` function, and fed to the `din.validate.qmatrix` function to obtain a refined version of the Q-matrix. For the results of the Chiu (2013) method, the R NPCD packaged is used (function `Qrefine`).

4. RESULTS

The three methods are evaluated over the 10 data sets and for 8 runs. Each run is conducted over a set of 30 different random combinations of perturbations, from 1 up to 10 perturbations. For the 1-perturbation condition, the total number of possible combinations is the size of the Q-matrix itself.

Figures 1 and 2 show the results broken down by real and synthetic data sets respectively, as space does not allow to report individual data set results.

Performance of each method is reported as a function of the number of perturbations. Recoveries are labelled "True Positives" (TP) whereas changes introduced by a method, but which do not correspond to perturbations introduced, are labelled "False Positives" (FP). The two graphs of figure 1 show the averages of the 6 real data sets, whereas the graphs of 2 show the averages for the 4 synthetic data sets. The "Total" line is shown to visually indicate the maximum that can be reached by a TP curve.

The ALS method shows the greatest ability to recover alterations, but at the cost of a higher rate of FP: changes that do not correspond to perturbations. It is followed closely by the Chiu (2013) method. The de la Torre (2008) method has a very low rate of recovery (TP) that makes it impractical. In general, the ALS and Chiu (2013) methods recover about 2/3 of the perturbations for synthetic data, and this rate falls to 1/2 for real data with ALS, and about 1/3 for Chiu (2013). For real data, the number of FP is around 5 for Chiu (2013) and around 6 for ALS, whereas it is respectively 2 and 3 for synthetic data. The relative performance of Chiu (2013) with respect to ALS is better for synthetic data and this might be explained by the fact that the data generation process is directly based on the DINA model.

A common pattern across methods is the relatively stable number of FP as a function of the number of perturbations. ALS does show an increase of close to 1 FP between 1 to 10 perturbations, whereas the increase for the Chiu (2013) and de la Torre (2008) methods is closer to 1/2 for real data, and even less for synthetic data (in fact it decreases for de la Torre (2008)). As a result, the rate of TP over FP increases with the number of perturbations.

Table 1: Data sets

Name	Number of			Description
	Skills	Items	Cases	
Sim. DINA	3	9	400	Artificial data available from the (`sim.dina`) data set of the CDM package.
Sim. DINO	3	9	400	Same parameters as No. 1 but using the DINO model (`sim.dino` data set).
Sim. CDM DINA	3	12	4000	Artificial data generated through the CDM function `sim.din`.
Sim. DCM	3	7	10000	Artificial data from chapter 9 of the book *Diagnostic Measurement* [13]
ECPE	3	28	2922	Dataset from [15] in [4]
Fraction	8	20	536	Tatsuoka's fraction algebra problems [14] (see table 1 in [5] for a description of the problems and of the skills).
Fraction 1	5	15	536	15 questions subset of Fraction with Q-matrix defined in [4].
Fraction 2/1	3	11	536	11 questions subset of Fraction with Q-matrix from [7].
Fraction 2/2	5	11	536	11 questions subset of Fraction with Q-matrix from [4].
Fraction 2/3	3	11	536	3 skills version of Fraction 1.

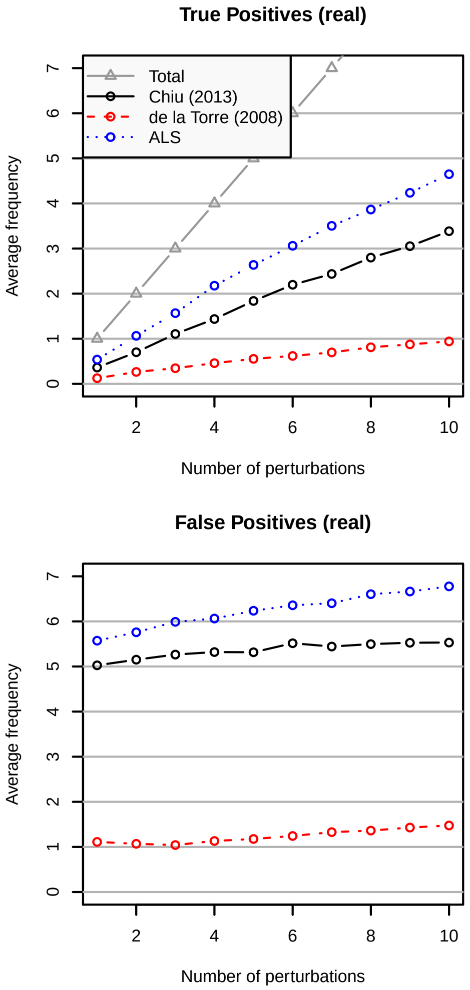


Figure 1: Average recovery rate by number of perturbations (real data)

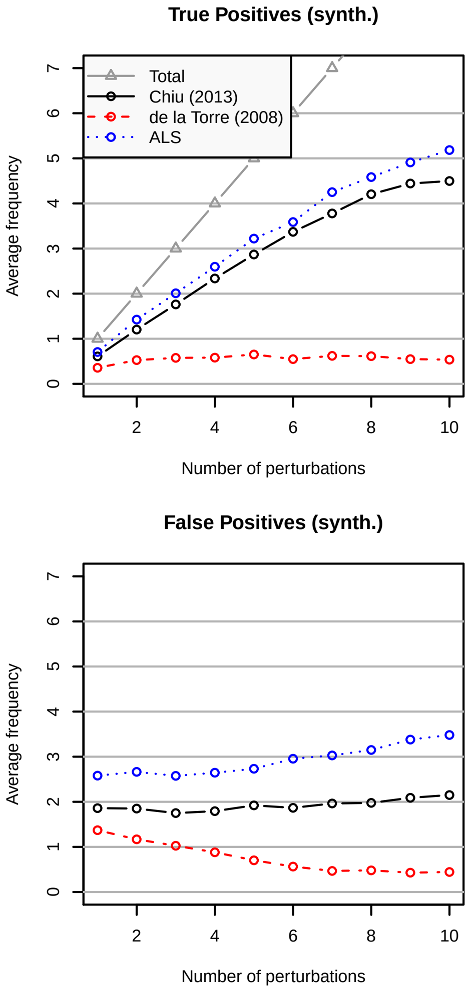


Figure 2: Average recovery rate by number of perturbations (synthetic data)

5. DISCUSSION

The contribution of this work is to provide performance assessments and compare existing methods of Q-matrix refinements based on a methodology and on metrics that allow meaningful comparisons. Previous work was limited to showing their ability to make Q-matrix refinements on an individual basis and in a restricted context.

The experiments conducted confirm that two methods, ALS and Chiu (2013) can recover the original Q-matrix from an altered one, as shown in previous work [2, 6], but the performance of the de la Torre (2008) method is considerably lower than the other two. The comparison of their performance over a number of data sets, and based on a common measure of performance, reveals wide differences across data sets (not reported here). As expected, all methods fare better on synthetic data sets, for which a close to perfect performance is reached with large samples. For real and synthetic data sets alike, the ALS and Chiu (2013) methods overall performances are comparable, but the advantage is spread between the two across the different data sets.

Can the methods be useful for refining Q-matrices in practice? Some issues clearly arise in the results. One issue is the size of the data sets required. For example, the Sim. DINA set has 400 cases and yet the best method only finds a single perturbation 1 time over 2. This result suggest that small samples of 100 cases or less are likely to be too small for being useful. In the days of big data from web deployment, for example, this is not such a major issue, but it does rule out some context of validation of a Q-matrix.

Another potential issue is that the results generally show more False Negatives than False Positives with real data. Note that, for real data, we cannot assume that all False Positives are wrong corrections. Some of them may represent potential improvements. Empirical evidence will be required to verify whether the suggested corrections do lead to real improvements when experts are presented with these corrections. Further work will also be required to validate if we can use the cross-evidence to filter out weak suggestions. For example, the recurrence of the same False Positives across perturbations and across techniques may yield stronger support to a suggestion.

Future work should also extend the comparison to more techniques such as [10, 11]. Finally, we stress the need for open access to the data and the code used in such studies. This particular study was highly facilitated by the CDM [12] and NPCD packages which provided both the code and the data.

Acknowledgements: This research was funded under the NSERC Discovery grant of the first author.

6. REFERENCES

[1] T. Barnes. Novel derivation and application of skill matrices: The Q-matrix method. *Handbook on Educational Data Mining*, 2010.

[2] C.-Y. Chiu. Statistical refinement of the Q-matrix in cognitive diagnosis. *Applied Psychological Measurement*, 2013.

[3] J. De La Torre. An empirically based method of Q-Matrix validation for the DINA model: Development and applications. *Journal of educational measurement*, 45(4):343–362, 2008.

[4] J. de la Torre. DINA model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1):115–130, 2009.

[5] L. T. DeCarlo. On the analysis of fraction subtraction data: The DINA model, classification, latent class sizes, and the Q-Matrix. *Applied Psychological Measurement*, 35:8–26, 2011.

[6] M. C. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based Q-Matrices. In *6th International Conference, AIED 2013, Memphis, TN, USA*, pages 441–450, 2013.

[7] R. A. Henson, J. L. Templin, and J. T. Willse. Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74(2):191–210, 2009.

[8] B. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.

[9] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8:30–43, 1997.

[10] J. Liu, G. Xu, and Z. Ying. Data-driven learning of Q-Matrix. *Applied Psychological Measurement*, 36(7):548–564, 2012.

[11] N. Loye, F. Caron, J. Pineault, M. Tessier-Baillargeaon, C. Burney-Vincent, and M. Gagnon. La validité du diagnostic issu d'un mariage entre didactique et mesure sur un test existant. *Des mécanismes pour assurer la validité de l'interprétation de la mesure en éducation*, 2:11–30, 2011.

[12] A. Robitzsch, T. Kiefer, A. George, A. Uenlue, and M. Robitzsch. Package CDM. 2012.

[13] A. A. Rupp, J. Templin, and R. A. Henson. *Diagnostic measurement: Theory, methods, and applications*. Guilford Press, 2010.

[14] K. Tatsuoka, U. of Illinois at Urbana-Champaign. Computer-based Education Research Laboratory, and N. I. of Education (US). *Analysis of errors in fraction addition and subtraction problems*. Computer-based Education Research Laboratory, University of Illinois, 1984.

[15] J. Templin and L. Hoffman. Obtaining diagnostic classification model estimates using mplus. *Educational Measurement: Issues and Practice*, 32(2):37–50, 2013.

Tracing Knowledge and Engagement in Parallel in an Intelligent Tutoring System

Sarah E Schultz
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA 01609
seschultz@wpi.edu

Ivon Arroyo
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA 01609
iarroyo@wpi.edu

ABSTRACT

Two of the major goals in Educational Data Mining are determining students' state of knowledge and determining whether students are affectively engaged with the task and in positive affective states. These two problems are usually examined separately and multiple methods have been proposed to solve each of them. However, little work has been done on tracing both of these states in parallel and the combined effect on a student's performance. In this work, we propose a model for tracing student engagement in parallel with knowledge as the student uses an Intelligent Tutoring System. We then compare this model to existing methods of tracing student knowledge and engagement.

Keywords

Knowledge tracing, engagement, performance, behavior, affect detection

1. INTRODUCTION

Intelligent Tutoring Systems are meant to adapt to a students' needs in order to better teach the student. In order to do this, they must have an estimation of student knowledge as the student progresses through the tutoring session. Systems might use their estimations of a student's mastery of the subject to decide whether to change the difficulty of problems given or progress to a new unit. These models may also be used by teachers and researchers to estimate students' mastery of skills or knowledge units. In the field of Educational Data Mining, the standard way to model and trace student knowledge is via knowledge tracing [1]. However, students often become disengaged as they use the software, as a result of boredom of frustration, confounding models which rely solely on performance data on individual questions to estimate knowledge, making it appear as though a student is forgetting.

The ability to detect affect is useful for Intelligent Tutors as it allows for the possibility for the tutor to intervene when a negative affective state is detected and help the student become engaged and motivated to learn. Some systems make use of sensor data to determine affect [7], but this is often impractical in a real-life learning scenario. Some researchers attempt to create sensor-less affect detectors using human coders who will observe students' apparent affective state during a session and then match these observations to behaviors that occur within the system at the same time in order to create a model, such as BROMP [11]. This is time-intensive, requiring a certain number of observations and highly trained coders.

While research has been done on tracing affective engagement without sensors or coders [3], little research has been done in modeling both knowledge and affect in parallel, attempting to account for these biases in knowledge estimation. In particular, a student's performance cannot be assumed to depend solely upon his or her knowledge of a skill, as how he or she is feeling will likely impact performance, as well. This is an area that is ripe for exploration.

Given a set of behaviors regarding correctness, timing and help seeking, some behaviors may be attributed to affective states, and some of them may be attributed to cognitive states [6, 7]. A Bayesian Hidden Markov Model (HMM) that attempts to trace knowledge and affect in parallel within the same model could potentially be able to discern between low affect and low knowledge, given a set of student correctness, timing and help seeking behaviors.

2. PREVIOUS WORK

The models explored in this work were inspired by previous successful Bayesian networks modeling students' knowledge and affect. The first of these is Knowledge Tracing, which has become a standard [1]. The second is the HMM-IRT model by Johns and Woolf [4], which took first steps towards modeling affect and knowledge in parallel.

2.1 Bayesian Knowledge Tracing

Corbett and Anderson's Bayesian Knowledge Tracing (BKT) [1] (Figure 1) is a hidden Markov model with two nodes at every time-step: the current (latent) knowledge state of the student and his or her performance on the current question (observed). Based on a student's correctness at answering questions at each time-step, the model estimates the probability that the student knows the current skill and then predicts the probability that the student will correctly answer the next question. The parameters for this model are $P(L_0)$, the probability that a student already knows the skill; P(T), the probability of learning the skill from one time-step to the next; P(G), the probability that a student who does not know the skill correctly guesses; and P(S), the probability that a student who does know the skill slips and gets the answer incorrect.

Traditionally, the KT model does not allow for forgetting (or unlearning) and this parameter is set to zero; this is in some way a quick fix, as when the model allows for forgetting, it is very sensitive to students "gaming the system" [9]. Consequently, estimates of knowledge mastery could quickly decline when students start behaving in these ways, such as hint abusing or quick guessing, and appear as if students are unlearning.

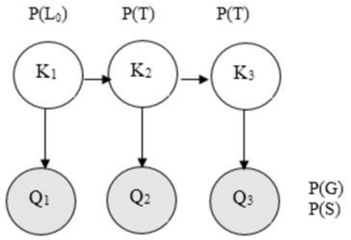


Figure 1- Bayesian Knowledge Tracing

2.2 HMM-IRT

Johns and Woolf [4] proposed another model, called the Hidden Markov Model-Item Response Theory (HMM-IRT) model. In this model, rather than using BKT, they use a hidden Markov model for tracing affect (what we call affective engagement in this paper), but pair it with a model for predicting student knowledge that relies on Item Response Theory for the estimation of conditional probabilities between specific question items and knowledge. Unlike BKT, this model estimates a single knowledge node. The HMM-IRT model allows the estimation of students' engagement at various time-steps (and relies on parameters of transitioning between affect/engagement states), but assumes a single mastery node, without learning or forgetting parameters.

The result of that research was that adding the affect/engagement component (top part of Figure 2) to the knowledge estimation model (bottom part of Figure 2) allowed for less of a decline in knowledge estimations after each question, which was apparently due to gaming behaviors and not due to unknowing.

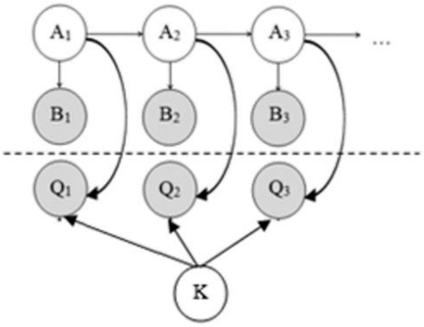


Figure 2- HMM-IRT Model

3. THE KAT MODEL

The Knowledge and Affect Tracing (KAT) model, shown in Figure 3, combines Knowledge Tracing with the HMM Affect Tracing portion of the HMM-IRT model, creating a model which allows for change in both students' knowledge and affective states. Both of these states influence question correctness.

The most important contributions, in our perspective, of both the HMM-IRT model and the KAT model, are the inclusion of transition probabilities between engaged states, in particular the probability of *becoming disengaged* in the next time step given that the student was previously engaged, and the probability of *becoming re-engaged* given that a student was previously disengaged. Knowing estimates of these probabilities for any learning system or for specific knowledge components should be very valuable to understand the impact of a learning system, or interventions. Similarly, it is valuable to know when estimates of engagement are low for personalization purposes, and knowing whether a student is likely game in the next problem or not.

The main drawback of the HMM-IRT model was that it did not include probabilities of acquisition or retention, but instead modeled students' knowledge as something stable and trait-like. Adding knowledge tracing to this model should enable researchers and systems to better predict both performance and behavior (gaming or not gaming) at the next step.

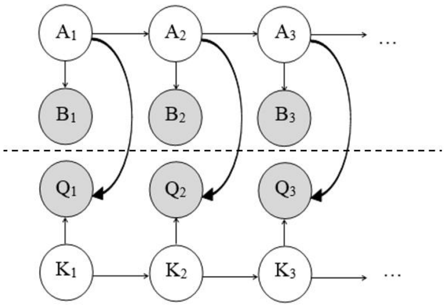


Figure 3- The KAT Model

The behaviors examined were the same as those used by Johns and Woolf [4]. These are quick guess (the student makes an attempt in less than four seconds), bottom out hint (the student uses all available hints), and normal (any other behavior). One additional behavior, called "many attempts", was also added for this work. This was defined as a student making more than three attempts at answering a problem. As multiple choice problems typically include only five possible answers, a student making more than three attempts has likely simply clicked on most choices. Baker, et al. have also shown relatively few attempts to be a good predictor of engaged concentration [8]. In preliminary tests of the KAT model, including "many attempts" as a possible behavior led to better fit than using only three behaviors in both datasets. The three behaviors not classified as normal are grouped as "gaming" behaviors in order to allow the models to predict whether a student will game at each opportunity. Although gaming is traditionally thought of as disengaged behavior, students could act in a way that is defined here as a gaming behavior even when they are engaged.

The new conditional probability tables of the observed nodes of the KAT model are shown in Tables 1 and 2. Knowing the skill (K), being affectively engaged (A), answering a question correctly (Q), and behaving normally (B) (i.e., not gaming) are indicated by "true" in their respective columns. The last column gives a name to these new probabilities to be estimated, which consist of guessing or slipping while being in a state of affective engagement or disengagement at the same time.

Table 1- CPT for Performance (Q) Nodes of KAT Model

Known **(Latent)**	**Engaged** **(Latent)**	**Correct** **(Observed)**	**Probability**
False	False	False	1-guess_not_eng
True	False	False	slip_not_eng
False	True	False	1-guess_engaged
True	True	False	slip_engaged
False	False	True	guess_not_eng
True	False	True	1-slip_not_eng
False	True	True	guess_engaged
True	True	True	1-slip_engaged

The probabilities associated to the Gaming Behavior nodes (B) are shown in table 2, and depend on affective engagement. These probabilities distinguish whether a student has gamed in a situation when he/she was actually truly engaged (some sort of an ‘affective slip’) corresponding to ‘game_engaged’ and its counterpart, where the student was actually affectively disengaged but apparently behaved normally this time (1-game_not_eng).

Table 2- CPT for Gaming Behavior Nodes (B) of KAT Model

Engaged (Latent)	**Non-Gaming-Behavior (Observed)**	**Probability**
False	False	game_not_eng
True	False	game_engaged
False	True	1-game_not_eng
True	True	1-game_engaged

San Pedro et al. showed that student knowledge of a skill is related to affect (for example, students who know a skill well are more likely to be engaged) [7], so a variation on the KAT model was created to take this into account. This model, KAT2, includes the link between knowledge and affect.

4. DATASETS

The data was gathered from student logs of two mathematics tutoring systems, ASSISTments [2] and Wayang Outpost [7], for middle and high school students. All problems in Wayang are multiple choice, while problems in ASSISTments generally, though not always, require students to type in their answer, instead.

The ASSISTments data used here is from the 2009-2010 school year. This data comes from a special type of problem in ASSISTments called “skill builders.” In skill builders, students practice a specific skill until they get three problems correct in a row, in which case the skill is considered “mastered,” or they reach a preset daily limit and are told to return later. The Wayang data set comes from the spring of 2009 and includes two hundred ninety five students in grades 7 through 10 from two rural-area schools in Massachusetts.

Five knowledge components were chosen from ASSISTments and four from Wayang to test the models as they are all limited to examining each knowledge component separately. Table 4 shows the breakdown of the data used by knowledge component.

Table 4- Knowledge Components Examined

Knowledge Component	**System**	**Number Students**	**Total Number Opps**	**% Gaming**
Box and Whisker	ASSISTments	505	2020	13
Circle Graph	ASSISTments	616	2487	30
Table	ASSISTments	713	2894	4
Pythagorean Theorem	ASSISTments	283	1290	10
Equations	ASSISTments	408	1598	35
Perimeter	Wayang	285	1422	15
Area	Wayang	279	1385	17
Angles	Wayang	274	1355	16
Triangles	Wayang	260	1267	20

5. METHODS

All models were built using Murphy’s Bayes Net toolbox for MATLAB [5]. A student-level five-fold cross validation was run on all models, keeping folds consistent across models. Parameters were learned for the training data using expectation maximization and then tested on the test data. This was done five times for each knowledge component, where each time a different fold served as the test data while the other four served as training data. For all models, predictions of performance at the next step were compared with actual performance in order to calculate mean absolute error (MAE) and root mean squared error (RMSE). Additionally, for KAT and HMM-IRT, predictions of behavior were compared to actual behaviors. As struggling students will see more questions assessing the same knowledge component in both ASSISTments skill builders and Wayang Outpost, only the first five opportunities within each knowledge component are examined to avoid over-fitting to such students. Since these five opportunities are likely to be within one session, not allowing time for students to forget material, forgetting is still assumed to be zero. All data and code used can be found at the first author’s webpage [10].

6. RESULTS

As both error metrics calculated, MAE and RMSE, resulted in patterns that were not significantly different, only RMSE is reported here.

Tables 5 and 6 show each model’s predictive performance on the ASSISTments data and Tables 7 and 8 show how well the models did on the Wayang data. Tables 5 illustrates the RMSE for each model’s prediction of students’ performance, while 6 shows the error of prediction for students’ behavior. These tables show the mean average of RMSEs across folds for each skill.

Table 5 – RMSE for Performance (Q)

Skill	**KT**	**HMMIRT**	**KAT**	**KAT2**
Box and Whisker	**0.426**	0.495	0.468	0.493
Circle Graph	**0.434**	0.524	0.507	0.512
Table	**0.467**	0.498	0.483	0.495
Pythagorean Theorem	**0.480**	0.498	0.484	0.503
Perimeter	**0.471**	0.476	0.476	0.476
Area	**0.455**	0.476	0.460	0.459
Angles	**0.454**	0.466	0.466	0.465
Triangles	**0.483**	0.487	0.4866	0.485

Table 6 – RMSE for Gaming Behavior (B)

Skill	**HMMIRT**	**KAT**	**KAT2**
Box and Whisker	0.350	0.326	**0.325**
Circle Graph	0.196	**0.178**	0.179
Table	0.462	**0.422**	0.433
Pythagorean Theorem	0.303	**0.295**	**0.295**
Perimeter	0.357	0.357	**0.356**
Area	0.377	0.362	**0.361**
Angles	0.377	0.359	**0.357**
Triangles	0.400	0.394	**0.392**

These tables show that BKT is the best predictor of student performance-- the correctness at answering future questions. The two KAT models also generally outperform HMM-IRT at predicting performance. The original KAT model was significantly better at predicting performance than the KAT2 model on the ASSISTments data (ttest p<0.05), except on the skill

"Table" (p=0.09), and although the KAT2 model performed slightly better on the Wayang data, this difference was not significant (p>0.1). Both KAT models are also significantly better at predicting behavior than the HMM-IRT model, except on the Wayang topic "Perimeter," where KAT2 is marginally better than HMM-IRT and KAT is marginally worse. The two KAT models were not significantly different with respect to predicting behavior, except for on the ASSISTments skill "Table," on which the original KAT model performed better.

7. DISCUSSION

While traditional BKT appears to be the best model for predicting student future correctness performance at math questions, KAT seems to be best at predicting performance and gaming behaviors simultaneously. KAT better predicts performance than HMM-IRT in eight of nine knowledge components tested and gaming behavior in all nine knowledge components, including six where there was a significant difference between KAT's predictions and HMM-IRT's. As KAT was significantly better than KAT2 at predicting student performance at math questions in one system, the KAT model appears to be a better choice for modeling students than the KAT2 variation.

The fact that KAT, which allows for student learning, was better able to predict performance means that it is quite likely that students are, in fact, learning while using these systems, so that the probability of acquisition and retention matter at the moment of predicting knowledge and performance in the next time slice. Assuming that a student's knowledge state does not change during the session, as in HMMIRT, leads to a poorer model fit.

It is interesting that KAT was also better at predicting behavior than HMMIRT, as both models use the same CPT for these nodes. When both affective transitions and learning are allowed, a change in performance can be attributed to either, or both, perhaps allowing a more accurate model of engagement, and therefore better predictions of gaming behavior.

8. CONTRIBUTIONS AND FUTURE WORK

This work introduced a new model, KAT, for tracing students' knowledge and engagement in parallel while using an ITS. While the traditional KT alone was slightly better at predicting performance than any of the other models, KAT was better at predicting student performance and behavior than the previously existing HMM-IRT model. A variation, the KAT2 model, was also explored and shown to be slightly weaker than the original KAT model.

While this work included the original form of the KAT model and one variation, many other variations could be valid. For example, research involving sensors and self-reports of affect has shown that performance on one question influences a student's affect at the next time-step [7]. This could be added to the KAT model to create another variation.

Future versions of the KAT model should also allow for more affective states, rather than measuring only engagement. For this study, it was useful to keep all variables binary in order to determine which model was best able to predict performance and behavior based on knowledge and engagement, but the KAT model is meant to be a model of knowledge and *affect* tracing. It is possible that allowing for more specific affective states could allow for better prediction of gaming. Perhaps being bored is more likely to lead to these behaviors than being frustrated, although both could fall under the category of "disengaged." Additionally, allowing for forgetting would be an interesting avenue to explore in the future, looking at the knowledge predictions. It is possible KT will predict students are forgetting whereas knowledge estimations will not change in models allowing for gaming.

ACKNOWLEDGEMENTS

We would like to thank the U.S. Department of Education, #P200A120238, Fellowships in Computer Science to Support the Learning Sciences & Security, Heffernan (PI) and the National Science Foundation, #1324385, Cyberlearning DIP, Impact of Adaptive Interventions on Student Affect, Performance, and Learning, Arroyo, Woolf and Burleson. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.We are also grateful for the feedback and advice of Dr. Neil Heffernan, Dr. Joseph Beck, Trenton Tabor, Michael Wixon, and the entire EDM research group at Worcester Polytechnic Institute. We would also like to thank ASSISTments for making their dataset used within publicly available.

REFERENCES

[1] Corbett, A.T., Anderson, J.R., "Knowledge tracing: Modeling the acquisition of procedural knowledge." *User Modeling and User-Adapted Interaction*, 1995, 4, p.253-278.

[2] Mendicino, M., Razzaq, L., & Heffernan, N. T. (2009). A Comparison of Traditional Homework to Computer-Supported Homework. Journal of Research on Technology in Education, 41(3).

[3] Beck, J.E. "Engagement tracing: using response times to model student disengagement." *Proceedings of AIED conference*, 2005. p. 88-95. IOS Press

[4] Johns, J. and Woolf, B.P. "A Dynamic Mixture Model to Detect Student Motivation and Proficiency." *Proceedings of AAAI Conference*, 2006, 1, p. 163-168.

[5] Murphy, K. "The Bayes Net Toolbox for MATLAB", *Computing Science and Statistics*, 2002.

[6] San Pedro, M.O.Z., Baker, R.S.J.d., Gowda, S.M., Heffernan, N.T. "Towards an Understanding of Affect and Knowledge from Student Interaction with an Intelligent Tutoring System." In *Proceedings of the 16th International Conference on Artificial Intelligence and Education*. Memphis, TN, USA, July 9-13, 2013.

[7] Arroyo, I., Cooper, D. G., Burleson, W., Woolf, B. P., Muldner, K., and Christopherson, R. "Emotion Sensors Go To School." In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*. Brighton, UK, July 6-10, 2009.

[8] Baker et al. "Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra." In *Proceedings of the 5th International Conference on Educational Data Mining*. Chania, Greece, June 19-21, 2012.

[9] Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". In *Proceedings of ACM CHI 2004: Computer-Human Interaction*, 383-390.

[10] Schultz, S. Webpage: users.wpi.edu/~seschultz

[11] Ocumpaugh, J., Baker, R.S.J.d., Rodrigo, M.M.T. (2012) "Baker-Rodrigo Observation Method Protocol (BROMP)" *1.0. Training Manual version 1.0. Technical Report*. New York, NY: EdLab. Manila, Philippines: Ateneo Laboratory for the Learning Sciences.

Tracking Choices: Computational Analysis of Learning Trajectories

Erica L. Snow
Arizona State University
Tempe, AZ, USA
Erica.L.Snow@asu.edu

Laura K. Allen
Arizona State University
Tempe, AZ, USA
LauraKAllen@asu.edu

G. Tanner Jackson
Arizona State University
Tempe, AZ, USA
TannerJackson@asu.edu

Danielle S. McNamara
Arizona State University
Tempe, AZ, USA
Danielle.McNamara@asu.edu

ABSTRACT

This study investigates how variations in students' trajectories within the tutoring system, Writing Pal, varied as a function of individual differences and ultimately related to changes in the quality and linguistic properties of prompt-based essays. Forty-two college students interacted freely with the computerized writing tutor for approximately 4 hours. Using a novel statistical technique (random walks), we visualized students' self-paced trajectories within the Writing Pal system interface. Analyses revealed that students' self-reported perseverance was predictive of more systematic interaction patterns. Students' interaction patterns did not directly influence the quality of their writing; however, students' trajectories within the system was related to changes in the fine-grained linguistic properties of their essays. These findings demonstrate the potential for random walks to provide researchers with a wealth of information about students' interactions and subsequent learning outcomes within adaptive learning environments.

Keywords

Intelligent Tutoring Systems, random walks, natural language processing, prompt-based writing

1. INTRODUCTION

Intelligent Tutoring Systems (ITSs) are sophisticated learning environments that provide students with individualized pedagogical instruction. This customized instruction is often based on variations in students' performance and ability levels [1]. For instance, many ITSs use students' previous knowledge and current skill levels to build user models specific to each student. These models are then tweaked and corrected through students' interactions within the system [1].

This level of customization affords students unique learning trajectories that often vary as a function of individual differences [2 - 3]. Indeed, researchers have shown that numerous individual differences can influence the way in which students interact and perform within ITSs [2 - 3]. One individual difference that has not been extensively studied in the domain of ITSs, and which may be important to the way in which students approach and interact with a system, is perseverance. This characteristic may be especially important for adaptive environments that provide students with an abundant amount of tasks that are scaffolded in a specific way. Previous work has shown that a student's perseverance is related to their ability to regulate their behaviors and achieve long-term goals [4]. Thus, if a student has high perseverance, they are more likely to continue learning tasks until the work is complete.

As students' experiences and trajectories vary, researchers are afforded a unique opportunity to examine the optimality of various learning paths within adaptive environments. To examine the efficacy of different routes within an adaptive environment, methodologies are needed that capture the nuanced ways in which students interact with ITSs across time. Dynamical analysis techniques offer researchers a unique means of visualizing and characterizing students' trajectories within these complex systems. These techniques focus on the fluid and complex interactions that are often missed by traditional static measures. Previous work with dynamical analyses has shown that these techniques can capture nuanced trends in students' choices within various adaptive environments [2]. Thus, these methodologies may provide researchers with tools to capture various trajectories and their subsequent impact on learning outcomes.

1.1 Writing Pal

The Writing Pal (W-Pal) is an ITS designed to provide students with comprehensive writing strategy instruction [5]. Specifically, W-Pal focuses on providing students with various strategies for prewriting, drafting, and revising. W-Pal is broken up into eight separate modules. Each module contains animated lesson videos that are narrated by a pedagogical agent, as well as game-based practice and essay writing practice. The design of W-Pal scaffolds students through these eight modules systematically and provides a deliberate form of instruction and practice.

1.2 Current Study

ITSs adapt to individual users' needs and abilities and often provide each user with a unique experience within the system. These varying experiences afford researchers an opportunity to examine optimal vs. non-optimal learning paths. The current study uses a novel dynamical methodology (random walks) to capture and evaluate students' trajectories within the writing tutor, W-Pal. Students were given free choice to interact with the system however they chose. W-Pal remained modular with an apparent organization or sequence of lessons; however, no feature was locked within the system. Thus, students could "jump around" within the interface if they so chose. Using random walks, we investigated how variations in students' choice patterns varied as a function of individual differences in perseverance. We also examined how variations in trajectories impacted changes in the quality and properties of students' writing.

2. METHODS

2.1 Participants

The participants included 42 college students from a large university campus in the Southwest United States. The students were, on average, 19.2 years of age, with the majority of students reporting their grade level as college freshman. Of the 42 students, 57% were female, 53% were Caucasian, 14% were Asian, 9% were African-American, 14% were Hispanic, and 10% reported other nationalities.

2.2 Procedure

The participants completed 4 sessions (6 hours total) including a pretest, strategy training within W-Pal, and a posttest. During the pretest (session 1), students completed questionnaires including measures of motivation, perseverance, and expectations of technology. Within the pretest, students were also asked to compose a timed (25-minute) essay in response to an SAT-style prompt. During training (sessions 2 and 3), students spent approximately 4 hours interacting freely within the system interface. The interface of the W-Pal system was entirely unlocked during training. The modules within W-Pal were still in an instructional scaffolding format; however, students were free to interact in the system however they saw fit. Finally, at posttest, students completed a battery of questionnaires similar to those in the pretest and composed a timed (25-minute) essay in response to an SAT-style prompt. Essay prompts were counterbalanced across pretest and posttest.

2.3 Measures

2.3.1 Writing Performance

During pretest and posttest, students were asked to write a timed (25-minute) SAT-style essay in response to a prompt. The quality of each student's essay was assessed through the use of an NLP algorithm [6]. This algorithm assigns essay scores on a 1 to 6 scale ranging from "Poor" to "Great."

2.3.2 Linguistics Features

To assess the linguistic features of the pretest and posttest essays, we utilized Coh-Metrix. Coh-Metrix is a computational text analysis tool that calculates linguistic indices at lower and higher-levels of given texts. The lexical indices in Coh-Metrix include word-level information, such as lexical diversity and word frequency. Syntactic measures comprise indices related to the complexity of sentences constructions, such as the number of modifiers per noun phrase and the incidence of agentless passive constructions in a text. Cohesion measures indicate connections between ideas in a text; some relevant measures include: incidence of connectives and content word overlap (for adjacent sentences and all sentences). Finally, Latent Semantic Analysis (LSA) is used to provide information about the semantic similarity of texts.

2.3.3 Perseverance

Students' perseverance was measured using the Duckworth et al. (2007) Grit scale [4]. This measure comprises 8 short questions designed to capture students' willingness to persist at tasks and persevere in the face of failure.

2.3.4 System Interaction Choices

Within the W-Pal system, students can chose to interact with a variety of features that fall into one of three categories. Each of these categories represents a different type of functionality within W-Pal; these functionalities are *lesson videos*, *game-based practice*, and *essay practice*.

2.5 Data Processing

Students' data logs from their interactions with W-Pal were used to trace and categorize every interaction into one of the three previously mentioned category types: lesson videos, educational games, and essay practice. Tracking students' choices with these three distinct features affords the opportunity to investigate patterns in students' choices during their time within the system. This is especially important given that the system interface was completely unlocked during training. Thus, this categorization provides a stealth means of assessing students' behaviors and corresponding trajectories when they are free to claim agency over their experience.

3. QUANTITATIVE METHOD

To examine variations in students' behavior patterns within W-Pal, random walks were conducted. This analytical tool provides a means to visualize students' trajectories within the system. The following section provides a brief description and explanation of random walks.

It can be difficult to visualize fine-grained patterns that emerge within categorical data. One mathematical tool that can provide researchers with a spatial representation of such patterns is a random walk [2]. Random walks were used within the current study to visualize and capture the fluctuations within students' interaction patterns in W-Pal. These patterns emerged through the sequential order of students' interactions with the three feature categories (i.e., lesson videos, educational games, and essay practice). To create a spatial representation of students' trajectories within the system, each category is given an arbitrary assignment along an orthogonal vector in an X, Y scatter plot. These assignments were as follows: educational games (0,1), lesson videos (1,0), and essay practice (0,-1). These vectors are random and not associated with any qualitative value; instead, they simply provide an orthogonal grid where we can view patterns of system interactions. Random walks have previously been used to trace students' interaction patterns within the game-based ITS, iSTART-ME [2].

To generate a visualization of students' time in the system, we created individual walks for each student by placing an imaginary particle at the origin (0,0). Then, using log data we moved the particle in a manner consistent with the vector assignment, which effectively assigns a movement to students' interaction choices within the system. The resulting walk is a combination of students' "movements" and thus gives us a fine-grained look at each student's trajectory within the W-Pal system.

To illustrate what a random walk might look like for a student within the W-Pal system, see Figure 1. The starting point for all students' walks is (0,0) where the horizontal and vertical axes intersect. In the example provided in Figure 1, the student's first interaction was a lesson video; so, the particle moves one unit to the right along the X-axis (see # 1 in Figure 1). The student's second interaction was with an educational game; thus, the particle moved one unit up along the Y-axis (see # 2 in Figure 1). The student's third interaction was another lesson video, which again moves the particle one unit right along the X-axis (see # 3 in Figure 1). The student's fourth and final interaction choice was essay practice, which moved the particle one unit down along the Y-axis (see # 4 in Figure 1). These simple rules allowed us to generate unique random walks for each of the 42 students.

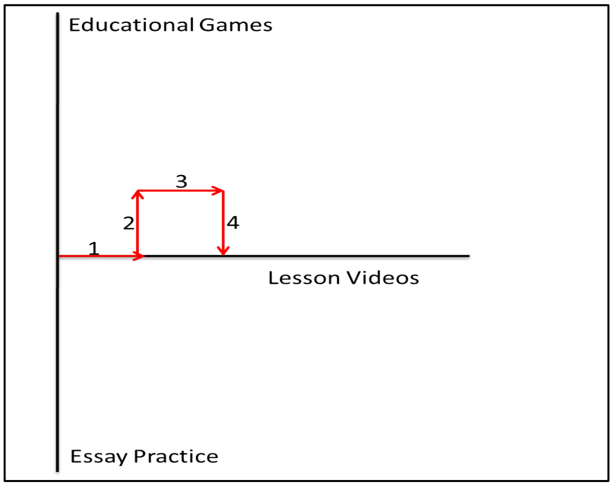


Figure 1. Example Random Walk within W-Pal

4. RESULTS

Students in the current study were free to interact with the W-Pal system in any way they saw fit. Using log data, we classified students' interactions into one of three possible categories (i.e., lesson videos, educational games, and essay practice). To examine how students interacted with the system, we calculated the total frequency of students' interactions with each of these three categories. On average, students made 19 interaction choices and spent the majority of the time watching lesson videos (73%) and playing games (26%). Only two students chose to interact with the essay practice; however, neither wrote more than two sentences before choosing to interact with a different feature (i.e., lesson video or educational game). As a result of these frequency analyses, we condensed our random walks to only include the X and Y coordinates associated with the lesson videos and educational games.

To examine students' patterns of interactions within the W-Pal system, log data were used to create a unique random walk for each student. These walks construct a visual representation of each student's unique interaction trajectory. We then calculated a slope for each student using the x and y coordinates embedded within their unique random walk (*M*=.23, *SD*=.14, Range= .00 - .46). Students' interaction trajectories (i.e., their slopes) inform us about the way in which students engaged in the system when everything was unlocked and they could "jump around" from module to module. These slopes serve as a coarse measure of each student's unique trajectory within the W-Pal system. Although slope analysis can obscure some of the variability in each student's unique walk, this metric provides valuable insight into the development of students' trajectories across time.

W-Pal was originally designed to be modular in nature thereby scaffolding students through systematic strategy instruction. Thus, this ordered design creates its own unique system trajectory. Using only lesson videos and educational games, we calculated a random walk for the system that represented the designed instructional scaffolding. We then computed a slope analysis to obtain the trajectory of the random walk that would be generated if students went through the system as designed (i.e., no skipping around). Thus, through the use of random walks we were able to look at differences between designed scaffolding trajectories (i.e., how researchers intended the system to be used) and students' trajectories (i.e., the way students' chose to use with the system).

Results from the slope analysis revealed that the system trajectory had a slope of .52. Interestingly, the highest slope value for any student was .46; thus, no one student went through the system exactly as it had been designed, although many students came close, thus skipping around the interface very rarely. In the current study, we hypothesized that students' self-report ratings of perseverance would be related to the way in which they approached the system. Utilizing these slopes, we examined the relation between slope magnitude and individual differences in perseverance. A correlation analysis revealed that the magnitude of walk slopes was positively related to students' self-reported perseverance (*r*=.332, *p*=.032). Thus, students who reported a higher likelihood to persevere (i.e., high Grit) demonstrated a more vertical trajectory, which more closely matched the system trajectory. A median split was calculated on students' pretest self-reports of perseverance (i.e., grit) to produce a visualization of the differences in system trajectories based on students' self-reported perseverance (Figure 2). This split produced two groups: *low grit* and *high grit* students, with low grit students represented as the red slopes and high grit students represented by green slopes. Within Figure 2, the black slope represents the system trajectory. This visualization supports the correlational results by revealing that high grit students (green slope) were much more likely to interact in a pattern similar to the designed instructional scaffolding (black slope).

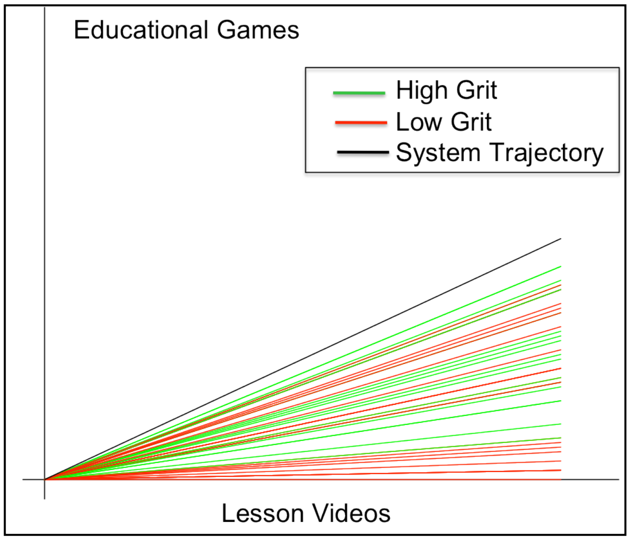


Figure 2. High and Low Grit Trajectory Comparison

To assess the relation between students' system trajectories and essay quality, Pearson correlations were conducted using students' random walk slope and their pretest, posttest, and total gain essay scores (i.e., posttest – pretest). Results from this analysis indicated that students' trajectories within the system showed a marginal negative relation to the quality of their pretest essays (r=-.291, p=.061). However, there was no significant relation between students trajectories in the system and the quality of their posttest essays (r=-.072, p=.649) or their holistic gain scores in essay quality (r=.188, p=.234).

Although there was no relation between students' trajectories and essay quality, we hypothesized that variations in students' trajectories within the W-Pal system might be related to changes in their essays at a more fine-grained size. We utilized Coh-Metrix to analyze the linguistic features of students' pretest and posttest essays. We then calculated a change score in the linguistic features (i.e., posttest – pretest) that indicated the extent to which linguistic features within the essays changed across the two assessments. Four Coh-Metrix change variables were significantly correlated to students' trajectories within the system. These variables were incidence of pronouns (r=-.381, p<.05), paragraph length (r=.331, p<.05), LSA paragraph to paragraph (r=.312, p<.05), and content word overlap (r=.371, p<.05).

To examine these relations further, a stepwise regression analysis was calculated to predict students' system trajectories from changes in the four Coh-Metrix variables that exhibited significant correlations with students' interaction trajectories. Two linguistic variable change scores were retained in the final model and combined to predict 27% of the variance in students' trajectories [F(1,39)=6.69, p=.014; R^2=.27]: noun incidence change [B=-.359, t(1,39) = -2.61, p=.012] and content word overlap change [B=.353, t(1,39)=2.57, p=.014]. Overall, this analysis indicated that when students' system trajectories resembled the actual system design, they were more likely to increase their local cohesion and substantive content of their essays.

5. DISCUSSION

ITSs are designed to provide customized instruction to students based on their individual needs and abilities [1]. This individualized instruction can often lead to students experiencing different learning trajectories within a given system. The emergence of these various learning trajectories has led researchers to begin to examine ways to investigate optimal versus non-optimal learning paths. One way to examine optimality within an adaptive environment is to examine how learning gains vary as a function of the trajectories students take within a system.

The current study made use of a novel methodology by employing random walks to capture and visualize students' unique interaction patterns within W-Pal. Random walk analyses revealed that students who interacted with the system in a more systematic way (i.e., closer to the designed instructional scaffolding) were also students who had higher perseverance scores. This indicates that students who jumped around the system more and did not follow the intended instructional scaffolding were individuals who reported having less perseverance. These results fall in line with previous work that has shown that individual differences influence learners' trajectories within adaptive environments [2 - 3].

Results from this study also revealed how learners' trajectories in the system influenced the quality and linguistic features of their writing. Overall, writing quality was not related to students' in-system trajectories. Interestingly, some changes in linguistics were related to students' trajectories within the system. Most notably, as students engaged in a trajectory that more closely resembled the designed system instructional scaffolding, their essays became more cohesive. Thus, these students' essays at posttest increased in the connecting of ideas.

Overall, the analyses presented here provide some promise that random walks are valuable data analytic and visualization tools that can shed light upon various behavioral trends exhibited by students. Indeed, through the use of dynamic methodologies, researchers may be able to better trace and ultimately recognize optimal versus non-optimal learning trajectories. These techniques also afford researchers the opportunity to investigate the efficiency of their system design.

5. ACKNOWLEDGEMENTS

This research was supported by the Institute for Educational Sciences (IES R305A080589). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the IES.

6. REFERENCES

[1] Vanlehn, K. 2006. The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16 (2006), 227-265.

[2] Snow, E. L., Likens, A., Jackson, G. T., and McNamara, D. S. 2013. Students' walk through tutoring: Using a random walk analysis to profile students. In S. K. D'Mello, R. A. Calvo, and A. Olney (Eds.), *Proceedings of the 6th International Conference on Educational Data Mining*, (Memphis, Tennessee, July 6 -9, 2013), Springer Berlin Heidelberg, 276-279.

[3] Snow, E. L., Jackson, G. T., Varner, L. K., and McNamara, D. S. 2013. The impact of performance orientation on students' interactions and achievements in an ITS. In C. Boonthum-Denecke & G. M. Youngblood (Eds.), *In Proceedings of the 26th Florida Artificial Intelligence Research Society Conference* (St. Petersburg, FL, May 21 – May 25, 2013). FLAIRS-26. AAAI Press, 521-526.

[4] Duckworth, A. L., Peterson, C., Matthews, M. D., and Kelly, D. R. 2007. Grit: perseverance and passion for long-term goals. *Journal of personality and social psychology*, 92 (2007), 1087.

[5] Roscoe, R. D., Snow, E. L., Brandon, R. D., and McNamara, D. S. (2013). Educational game enjoyment, perceptions, and features in an intelligent writing tutor. In C. Boonthum-Denecke & G. M. Youngblood (Eds.), In *Proceedings of the 26th Florida Artificial Intelligence Research Society Conference* (St. Petersburg, FL, May 21 –May 25, 2013). FLAIRS-26. AAAI Press, 515-520.

[6] McNamara, D. S., Crossley, S. A., and Roscoe, R. D. 2013. Natural language processing in an intelligent writing strategy tutoring system. *Behavior Research Methods,* 45, (2013) 499-512.

Unraveling Students' Interaction Around a Tangible Interface Using Gesture Recognition

Bertrand Schneider
Stanford University
schneibe@stanford.edu

Paulo Blikstein
Stanford University
paulob@stanford.edu

ABSTRACT

In this paper, we describe techniques to use multimodal learning analytics to analyze data collected around an interactive tangible learning environment. In a previous study [4], we designed and evaluated a Tangible User Interface (TUI) where dyads of students were asked to learn about the human hearing system by *reconstructing* it. In the current study, we present the analysis of the data collected in form of their gestures, and we describe how we extracted meaningful predictors for students' learning from this datasets. We explored how Kinect^{TM} data can inform "in-situ" interactions around a tabletop (i.e. using clustering algorithms to find prototypical body positions). We discuss the implications of those results for analyzing data from rich, multimodal learning environments.

Keywords

Tangible User Interface; Data Mining; Constructionism.

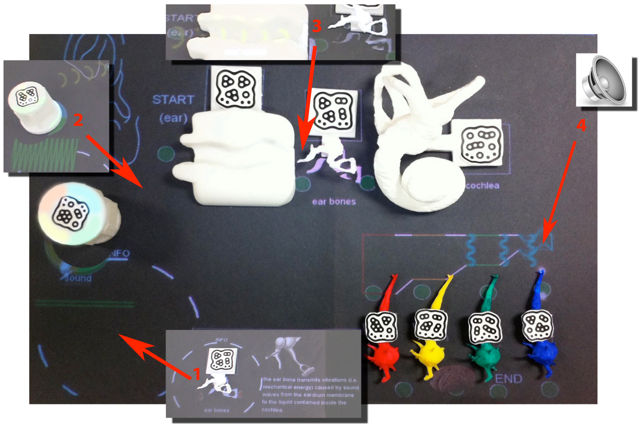
1. INTRODUCTION

Students' gestures have been extensively researched in the learning sciences: numerous studies on embodied cognition have unraveled links between learning and students' gestures. More generally, there has been a plethora of studies about people's intuitive representations of everyday situation and bodily language [3]. This line of research has provided new ways to understand the way students integrate new concepts to their everyday understanding of science phenomena.

Yet, this field of research suffers from serious methodological limitations. Most studies are qualitative by nature, or require researchers to manually annotate hours of video recordings. Now that the theoretical underpinnings of the field are established, it would be the right time to speed up discovery and data analysis, but the pace at which new results are generated is slower than desired, especially for highly granular data. The emerging field learning analytics and educational data mining, and especially the field of multimodal learning analytics [9] might provide just the right data collection and analysis tools to tackle this problem. Thus, the goal of this paper is to address this methodological gap by suggesting new ways to conduct research on students' body language, as well as providing news lenses to look at students' micro-behaviors while learning. In our study, we collected data on user's actions by using a Microsoft Kinect™. We then used data mining techniques to make sense of those two datasets.

2. THE CURRENT STUDY

In a previous study, we were interested in pursuing the work started with two other TUIs developed in our lab. In our research, we have found that TUIs can be advantageously used in a discovery-learning situation when students approach an unfamiliar topic compared to a standard "tell-and practice" instruction (i.e. using a TUI before, rather than after, a standard kind of instruction such as reading a textbook chapter or attending a lecture). In a controlled experiment [6], we showed that students who first used a TUI and then read a textbook chapter outperformed students who completed the same activities but in the reverse order (text followed by TUI). To show that being physically engaged doesn't fully explain our results, we designed the following experiment, where students were asked to discover how the human hearing system works (N=38). Pairs of students worked on a tangible interface called EarExplorer (Fig. 1) where they learned about the human hearing system by reconstructing it. In one condition, students rebuilt the hearing system by trial and error, using resources provided by the system. In a second condition, they used the same setup except that a video of teacher demonstrated how to rebuild the hearing system and explained the function of each organ as students progressed through the activity. Students in both conditions then read a textbook chapter explaining sound transduction in a more formal way. We found that students in the first group achieved a higher learning gain as measured by the pre and post-test. A MANOVA showed that participants in the "discover" group learned significantly more after the first activity: $F(1,35) = 22.11$, $p < 0.001$ and after the second activity $F(1,35) = 16.15$, $p < 0.001$ compared to the participants in the "listen" condition.



the ear bone transmits vibrations (i.e. mechanical energy) caused by sound waves from the eardrum membrane to the liquid contained inside the cochlea.

Figure 1: The EarExplorer Interface. Students use the infobox (1) to learn about the different organs; they then generate sounds at different frequencies with a speaker (2); sound waves travel from the emitter through the ear canal to the ear bones (3); finally, the sound reached the basilar membrane inside the cochlea, activates a specific neuron and replayed the sound if the configuration is correct (4).

Those results suggest that hands-on activities alone don't fully take advantage of educational TUIs; rather, discovery should be an integral part of designing interactive hands-on activities. Given those findings, our goal is to take a new look at this dataset by applying data mining and multimodal learning analytics techniques to students' body language: we will analyze the logs generated by a Kinect™ sensor that recorded students' actions.

2.1 Kinect data

As described earlier, a Kinect sensor captured the body movements of both students during the learning activity (30 data points per second). For the sake of simplicity, we only analyzed the students' body language during the hands-on activity (i.e. when reconstructing the hearing system, as opposed to reading the text which is the second activity). This step lasted 15 minutes, which gives us 15 * 60 * 30 = 27000 data points for each participant. Since 38 students took part in the study and two were removed for missing data, we have approximately 1 million data points from the Kinect sensor. This is a relatively large dataset that needs to be drastically simplified in order to make sense of it.

2.2 Movements

The most straightforward (and admittedly naïve) approach to analyzing the Kinect dataset is to compute the amount of movement generated by each participant. The hypothesis is that more engaged students move their body more than less engaged ones, and that physical movement is a proxy for general engagement; this general engagement in turn is related to learning gains. There are two ways to compute this metric: the first one is by calculating the Euclidean distance between each joint and averaging the result over time. This approach is not ideal, because it does not take into account the natural variations in students limbs' lengths. An arguably better way to compute movements is to look at variations in angles between joints in body positions. We tried both approaches and sliced the data over time to get a measure for each minute. We also computed an overall score, as well as a score for each joint. We did not find any significant correlation between the measures described in this paragraph and learning gains. For instance the amount of movement computed with joint angles produced the following correlation: r(34) = 0.079, p = 0.648. On the one hand, this result is somewhat surprising: we would expect at least some of those measures to be associated with higher engagement and thus more learning. On the other hand, a movement of the hand, for instance, can mean a range of different things (e.g. a sign of boredom, interest, a deictic gesture, and so on), so ultimately the results make sense. Many simple gestures are ambiguous by nature, and in our particular case we did not have enough information to correctly contextualize them. In the next sections, we look at more refined measures of students' activity, such as bimanual coordination, body synchronization and postures.

2.3 Bimanual Coordination

In a related study, Worsley & Blikstein [9] have shown that bimanual coordination was predictive of participants' expertise in solving an engineering problem. Based on these results, we decided to compute a similar metric. More specifically, the idea is to compute and compare the amount of movement generated by each hand. Figure 2 shows two examples generated by this approach: on the right, the student barely used his left arm and the student on the left is used both arms during the entire activity. However, to make sense of this metric, we need to introduce additional results that we found on the initial dataset.

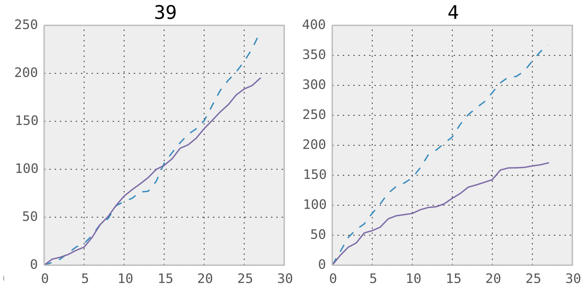


Figure 2: hand coordination of two students during the activity. Participant #39 is bimanual and uses both hands. Participant #4 uses predominantly his right hand (blue line).

Previous research has shown that each student working in groups can often be categorized as either being the "driver" or the "passenger" of the interaction [7]. Several indicators can be used to categorize each dyad's members: 1) who started the discussion when the experimenter leaves, 2) who spoke most, 3) who managed turn-taking (e.g., by asking "what do you think?", "how do you understand this part of the diagram?"), and 4) who decides the next focus of attention (e.g., "so to summarize, our answers are [...]. I think we need to spend more time on X"). This measure can be considered as an aggregate estimation over the whole activity of the dyad's dynamic profile. We acknowledge that subjects are likely to shift roles during the activity. We also recognize that this categorization is more likely to be a continuum, and that in a few cases the difference between drivers and passengers may be subtle.

In our case, after making this distinction for students, we further separated them by computing a median-split on their GPA. This resulted in four categories: a student could either be a driver or a passenger with a high or low GPA (Fig. 3). Surprisingly, having a proficient driver in the group does not lead to higher learning gains: F(1,16) = 0.04, p = 0.84, Cohen's d = 0.17 (low GPA driver: mean=7.63, SD=1.84; high GPA driver: mean=7.87, SD=2.57). On the other hand, having a passenger with a high GPA *does* lead to increased learning gains: F(1,18) = 3.51, p = 0.08, Cohen's d = 1.4 (low GPA passenger: mean=6.22, SD=2.26; high GPA passenger: mean=8.36, SD=2.43).

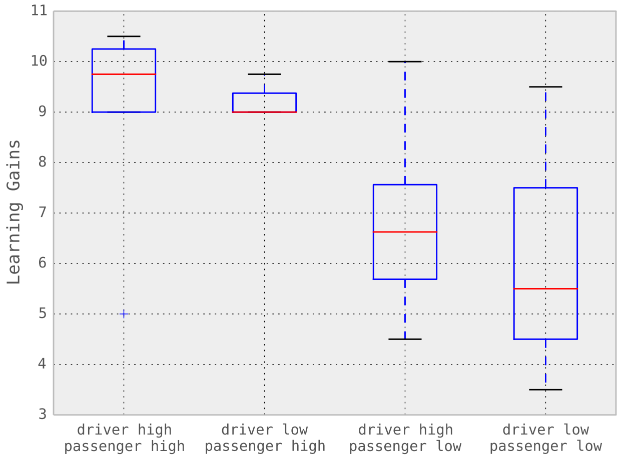


Figure 3: Boxplots of the four kinds of dyads described above: driver / passenger with high / low GPA.

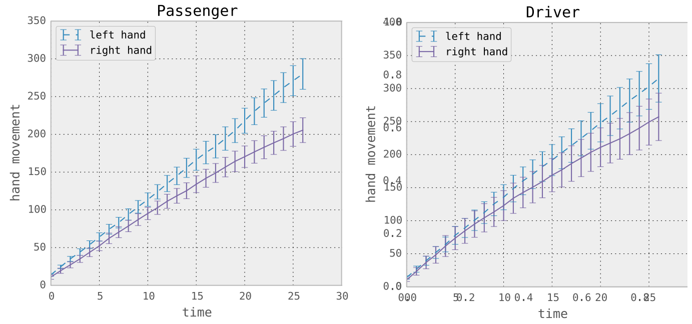


Figure 4: Bimanual coordination from Drivers and Passengers in dyads of students.

This result is not totally unexpected: proficient students who do not "take control" of the activity tended to leave more space for trial and error to their partner and suggested hints when needed. This situation resulted in increased participation and engagement from the low GPA student. In the opposite situation, the same student would stay passive and let the driver solve the problem on her/his own. The distinction between proficient / non-proficient drivers / passengers allowed us to find interesting patterns in our data. More specifically, we found that drivers tend to use both hands while the passenger uses at most one hand. Figure 4 show the aggregated evaluation over time of the hand movements of those two types of students. Using an ANOVA, we found a significant differences between the amount of movements of each hand for the passengers at the end of the activity: F(1,35) = 7.66, p = 0.01 (left hand mean=280.00, SD=86.30; right hand mean=205.55, SD=69.62). This difference was not significant for the drivers: F(1,35) = 1.24, p = 0.27 (left hand mean=315.38, SD=152.32; right hand mean=257.21, SD=152.27).

This result shows that we can potentially discriminate between drivers and passengers by looking at their hand movements. As a possible implication of this result, we can imagine future systems where machine-learning algorithms will make predictions about the "status" of each member of a dyad. Using many more features, we can imagine a learning environment where personalized scaffolding is provided depending on the groups' dynamic: proficient leaders can be encouraged to take a more passive role, while less proficient students would be provided with more scaffoldings and more opportunities to participate.

2.4 Going Beyond Movements

While the previous results provide us with interesting metrics to predict learning, they are rather unsophisticated. In this section we describe how we used clustering algorithms to automate the creation of coding schemes on body postures. Recall that most previous work in this area was conducted manually, by analyzing videos frame by frame in a highly time consuming process. If we can show that an algorithm can accomplish a similar task, it will provide researchers with an easy and efficient way to quickly analyze students' body language. Our approach was to take our entire dataset (1 million entries), and transform it into (joint) angles instead of positions in a three-dimensional Cartesian coordinate system. We then fed this matrix into a simple clustering algorithm (K-means) that provided us with prototypical body positions. As a first step, we decided to keep our analyses as simple as possible and limited the number of clusters to three. The results are shown in Figure 5. We found the three clusters to have interesting properties: the first one (top left) represents an "active" position: both arms are on the table, supposedly manipulating something or at least ready to act; the head is tilted toward the table in an attentive position. The second cluster (top right) shows a "semi-active" posture: one arm is flexed, while the other one is straight on the table, probably manipulating a tangible. The last one (bottom left) represents a "passive" posture, where both arms are crossed and the body looks relaxed. We then used those three clusters to classify each data point into one of those three clusters based on proximity to cluster centroids and counted how many times each student was in each posture. The way we interpret those three clusters seem to correlate with our previous measures: the first posture is positively associated with students' learning gains r(34) = 0.329, p < 0.05 while the third one is negatively correlated with students learning gains r(34) = -0.420, p < 0.05. Additionally, we found that the number of times students transitioned from one posture to another was also significantly correlated with their learning gains: r(34) = 0.335, p < 0.05. This suggests not only that some postures are indicative of learning, but certain sequences of postures are too.

Previous work [8] has shown that "ideal" cycles of cognition (i.e. planning, executing and evaluating an action) are usually associated with higher performances and higher learning gains. It is possible that the results of our clustering algorithm produced a similar construct: an increased number of cycles where students think for a while (posture 1 and 2) and then execute an action (posture 3) could be interpreted as something akin to an ideal cycle of cognition described by [8]. We should mention that we tried several approaches before finding the optimal way to cluster our dataset. We first tried to use joint *positions* in a three-dimensional space, as measured by the Kinect (i.e. the x,y,z coordinates of each joint of the kinect skeleton: head, neck, shoulders, elbows, arms). We found two main issues with this method: first, clusters were influenced by students' orientation toward the tangible interface (right or left side). Second, the size of their limbs interfered with the clustering algorithms: longer limbs were more likely to be clustered together.

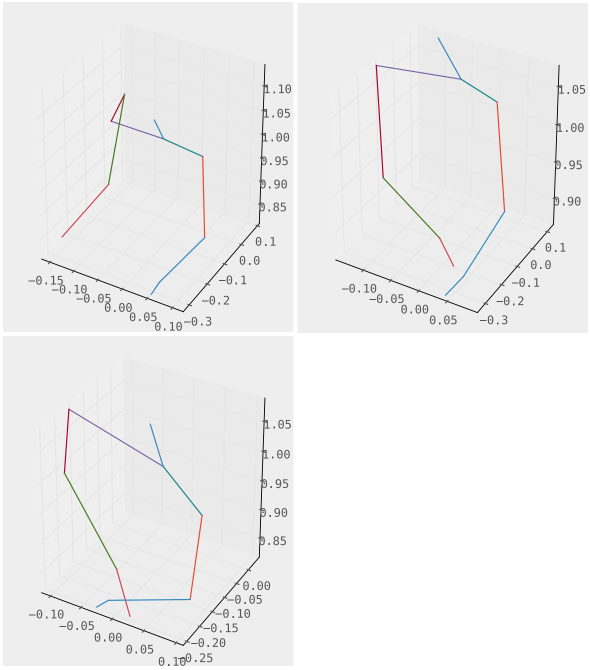


Figure 5: The results of the k-means algorithm on students' body posture (1 million data points).

2.5 Analyses at the dyad level

We describe here additional results conducted at the dyad level, i.e. when taking both bodies into consideration.

2.5.1 Body Synchronization

In a previous study [5], Schneider & Pea found that students' visual synchronization (as measured by eye-trackers) was correlated with their learning gains. That is, more moments of joint attention was beneficial to establishing a common ground, which in turn positively influenced how much students learned during an activity. Other lines of research suggest that body synchronization is associated with productive collaborations [1]. We were inspired by those results and decided to compute a metric for gestures synchronization using the Kinect data. Our approach was to first take pairs of data points (one from each student) and computes the distance between them. Distance was calculated by taking the absolute value of the difference between the joint angles of each participant. An ANOVA did not reveal any significant effect of this measure on our experimental manipulation: F(1,17) = 0.92, p = 0.35, Cohen's d = 0.14 ("discover" mean=0.46, SD=0.09; "listen" mean=0.42, SD=0.05). We also did not find a significant correlation between body synchronization and learning gains: r(16) = 0.189, p = 0.453. It suggests that even though gaze synchronization is a strong predictor for students' quality of collaboration, body synchronization does not hold the same properties in our dataset.

2.5.2 Body Distance

A last metric that we assessed was inspired by the theory of Proxemics developed by Edward T. Hall [2]. In this seminal work, he proposed to categorize the distances around a person into different zones: the intimate area (less than 15 cm to 46 cm), the personal space (46 to 122 cm), the social distance (122 to 370 cm) and the public distance (370 to 760cm or more). Interestingly, in our study students were seated at a distance that varied between the intimate and personal distance. Moving from a personal to an intimate distance is considered a violation of someone's territory if there is not implicit agreement that someone can do so. Thus, a small distance between two students can potentially characterize a productive collaboration and thus higher learning gains. Similarly, a larger distance can be an indicator of a poor collaboration. We computed the distance between students at each data point (i.e. 30 times per second) by taking the rightmost joint from the student on the left side and the leftmost joint from the student on the right side of the table; we then calculated the Euclidean distance between those two points and averaged a global score for the entire activity (27000 data points). We did not find a correlation between learning and the distance between students' bodies: r(16) = 0.377, p = 0.123. However we found that this metric was correlated with students' pre-existing knowledge on the topic taught (i.e. score on the pre-test): r(16) = -0.548, p = 0.019. While there could be multiple interpretations of this result, it suggests that students who are unfamiliar or uncomfortable with the subject matter tend to establish a larger distance with their peers and possibly be more defensive during a collaborative activity.

3. DISCUSSION

In this paper, we developed several metrics that can be used to predict students' learning around an interactive tabletop. We found that the raw amount of movement was not a relevant predictor for our purposes; however, bimanual coordination was predictive of students' leadership in a group. Additionally, clustering body position with k-means provided us with three categories: we found that "active" positions were correlated with learning gains, "passive" positions were negatively correlated with learning gains, and that transition between those states was predictive of learning. Third, we explored students' body language on a social level: contrary to common social psychology theories, we found that body synchronization was not correlated with any of our measures. Fourth, the distance between students' bodies during the activity was associated with their pre-existing knowledge on the topic taught: students with low scores on the pre-test tended to be further away from their partner compared to students who obtained a high score.

4. CONCLUSION

Our goal was to show the potential of rich datasets for advancing our understanding of students' learning trajectories. We contrast our approach with online settings where the data is drastically more limited (e.g. click-stream data). We believe that insights are more likely to be generated in these "in-situ" settings, because researchers can more easily collect relevant educational data: for instance gestures captured with a Kinect sensor, eye gaze movements collected with (mobile) eye-trackers, and arousal measures gathered using galvanic skin response sensors. There are multiple implications stemming from our results. One of them is that seemingly erratic events such as gestures can be objectively correlated with learning gains in ecologically-valid tasks -- i.e., students were using an interactive tabletop which had no constrains for gestures -- everything was allowed. The task itself was very open-ended -- there were multiple paths to success. This is a departure from research that uses data mining in very constricted and well-structured tasks.

5. REFERENCES

[1] Chartrand, T.L. and Bargh, J.A. The chameleon effect: The perception–behavior link and social interaction. *Journal of Personality and Social Psychology 76*, 6 (1999), 893–910.

[2] Hall, E.T. *The hidden dimension (1st ed.)*. Doubleday & Co, New York, NY, US, 1966.

[3] Roth, W.-M. Gestures: Their Role in Teaching and Learning. *Review of Educational Research 71*, 3 (2001), 365–392.

[4] Schneider, B., Bumbacher, E., Salehi, S., & Blikstein, P. Discovery Learning versus Traditional Instruction with a Tangible Interface. *Unpublished Manuscript.*

[5] Schneider, B. and Pea, R. Real-time mutual gaze perception enhances collaborative learning and collaboration quality. *International Journal of Computer-Supported Collaborative Learning 8*, 4 (2013), 375–397.

[6] Schneider, B., Wallace, J., Blikstein, P., and Pea, R. Preparing for Future Learning with a Tangible User Interface: The Case of Neuroscience. *IEEE Transactions on Learning Technologies 99*, 1 (5555), 1.

[7] Shaer, O., Strait, M., Valdes, C., Feng, T., Lintz, M., and Wang, H. Enhancing genomic learning through tabletop interaction. *In Proc. of the 2011 conference on Human factors in computing systems*, ACM (2011), 2817–2826.

[8] Tschan, F. Ideal Cycles of Communication (or Cognitions) in Triads, Dyads, and Individuals. *Small Group Research 33*, 6 (2002), 615 –643.

[9] Worsley, M. and Blikstein, P. Towards the Development of Multimodal Action Based Assessment. *In Proc. of LAK2013*, ACM (2013), 94–101.

Poster Papers

A Predictive Model for Video Lectures Classification

Priscylla Silva
Institute of Computing
Federal University of Alagoas
Alagoas, Brasil
pmss@ic.ufal.br

Roberth Pinheiro
Institute of Computing
Federal University of Alagoas
Alagoas, Brasil
rraraujop@gmail.com

Evandro Costa
Institute of Computing
Federal University of Alagoas
Alagoas, Brasil
evandro@ic.ufal.br

ABSTRACT

In the educational context, it is important to provide students with learning resources, such as tutorials, video lectures, and educational games to help their learning process, especially when they are not in the school and have difficulties or doubts. In these situations, recommendation systems may be used to suggest learning resources for students, avoiding, for instance, the task of making the manual process of searching and selecting resources. Most generic recommendation systems for video lectures use viewing history of the user to make recommendations of videos, which are in according of the user interests. In the educational context, other factors must be considered, the video should not only be of interest to the student, as it will be used as a learning resource for the main purpose of helping the student to learn a particular subject or clarify doubts. Hence, in this paper we evaluated three classifiers and propose a predictive model to classify video lectures according to their quality. We applied machine-learning algorithms on a set of video lectures by classified students according to some quality requirements. We conducted an experiment and preliminary results indicate good quality of the selected prediction model.

1. INTRODUCTION

The search for relevant information on the Web is a well known problem that has been addressed by several studies in the literature. Recommendation systems have been considered one important way to address this problem. In the educational context, recommendation systems are used to recommend learning resources for students, saving the students the time by manual process of searching and selecting resources. One of the most used resources by students are the video lectures. Websites like Youtube[1] and Vimeo[2] have many videos within various themes, including video lectures of various topics.

The amount of videos on the Internet is growing at an explosive rate [1], making it harder the search for good and appropriate video lectures. When students have to learn a subject or they are in doubt, they generally perform the following steps: 1. Search on a website of videos using keywords from the subject; 2. Choose one of the first videos ranked to watch; 3. If the video is not good enough for them, then they stop watching it and try another video. It may occur students selecting several bad videos followed, until they find a video that they classifies as good and watch the full video to learn what they need. This happens because many students have no way to predict the quality of video lecture selected.

Most generic recommendation systems for videos uses viewing history from previous users to suggest videos [3]. In the educational context, other factors must be considered, the video should not only be of interest to students, as it will be used as a learning resource for the main purpose of helping students to learn a particular subject or clarify doubts. In this paper we evaluated three classifiers to classify video lectures according to their quality. The classifiers Navie Bayes, SVM and C4.5 were used. The classifier with better performance compared to others was selected.

2. EXPERIMENT

The purpose of the experiment was to evaluate the predictive ability from classifiers towards video evaluation. For perform the experiment the Weka software was used [2].

To perform the experiment 120 video lectures from *YouTube* were collected in the mathematic domain. The video lectures belong to the following topics: logarithms, Cartesian plane, set theory, polynomial functions, geometric progression and matrices. A total of 15 undergraduate students volunteered to evaluate videos. Each video lecture was assessed by 5 students who had attended courses that have mathematical knowledge as a prerequisite, such as calculus and linear algebra. Students evaluated the video lectures by applying grades from 0 to 10, median grade from this evaliation was used to generate the overall assessment of video lecture. Students were instructed to review the video lectures on the following criteria: clarity, teaching method, depth in the proposed issue, audio quality and image, teacher's didactic, among others. A video lecture is composed of the following attributes: title size, description size, duration, date of publication, view count, like count, dislike count and comment count. The attributes were normalized and then discretized.

[1] http://www.youtube.com/
[2] http://vimeo.com/

All attributes have been discretized using histogram analysis. Each video has a class called "evaluation" that can take the labels: inadequate, bad, average, good, excellent.

The experiment was performed using 10-fold stratified cross-validation. This procedure divides the sample into k mutually exclusive parts (folds), for each step, $k-1$ folds are used for training and the induced hypothesis is tested on the remaining fold. In order to get statistically meaningful results, the number of iterations used was 10. In case of 10-fold cross-validation this means 100 calls of one classifier with training data and tested against test data. The current experiment performs 10 runs of 10-fold stratified cross-validation on the dataset using Navie Bayes, SVM and C4.5 scheme, this means 300 calls. The experiment consists in to confirm if the video lectures were automatically labeled correctly (in the sense of assigning a evaluation).

2.1 Evaluation Metrics

Given an algorithm A and a set of instances denominated T, assume that T is divided into k partitions. In the case of 10-fold cross-validation, $k = 10$. For each partition i, the hypothesis h_i is induced and the error denoted by $err(h_i)$, where $i = \{1, 2, ..., k\}$ is calculated. The mean, variance and standard deviation for all partitions are calculated using the following formulas: i) $mean(A) = mean(A,T) = \frac{1}{k}\sum_{i=1}^{k} err(h_i)$; ii) $var(A) = var(A,T) = \frac{1}{k}\left[\frac{1}{k-1}\sum_{i=1}^{k}(err(h_i) - mean(A,T))^2\right]$ iii) $sd(A) = sd(A,T) = \sqrt{var(A,T)}$.

When comparing two inductors in the same domain T, the standard deviation can be seen as a picture of the robustness of the algorithm: if the errors (calculated on different test sets) derived from induced hypotheses using different training sets are very different from one experiment to another, this indicates that the inductor is not robust to changes in the training set, coming from the same distribution. To compare two machine learning algorithms and decide which one is better (with confidence level of 80%), just take the general case to determine whether the difference between two algorithms (A_i and A_j) is significant or not, assuming a distribution normal. For this, the mean and standard deviation combinations are calculated according to the following equations: i) $mean(A_i - A_j) = mean(A_i) - mean(A_j)$; ii) $sd(A_i - A_j) = \sqrt{\frac{sd(A_i)^2 + sd(A_j)^2}{2}}$; iii) $ad(A_i - A_j) = \frac{mean(A_i - A_j)}{sd(A_i - A_j)}$. The absolute difference (ad) is given in standard deviations.

If $ad(A_i - A_j) > 0$ then A_j overcomes A_i and if $ad(A_i - A_j) >= 1.29$ then A_j overcomes A_i with 80% degree of confidence.

If $ad(A_i - A_j) <= 0$ then A_i overcomes A_j and if $ad(A_i - A_j) <= -1.29$ then A_i overcomes A_j with 80% degree of confidence.

3. RESULTS AND DISCUSSION

In the results of the experiment, the classifier that showed the best performance was the SVM. Table 1 shows the results of the comparative analysis of used classifiers.

Table 1: The comparative analysis of used classifiers.

A_i	Navie Bayes	SVM	C4.5
A_j	SVM	C4.5	Navie Bayes
mean	0,08	-20,09	20,09
sd	0,05	56,96	56,96
ad	1,41	-0,35	0,35

In the comparison between Navie Bayes and SVM, where A_i = Navie Bayes and A_j = SVM, we have $ad(A_i - A_j) > 0$ and $ad(A_i - A_j) > 1.29$, therefore the SVM outperforms Navie Bayes with confidence level of 80%. In the comparison between SVM and C4.5, where A_i = SVM and A_j = C4.5, we have $ad(A_i - A_j) < 0$, therefore the SVM outperforms C4.5, but does not overcome the level of confidence of 80%, because $ad(A_i - A_j) > -1.29$. In the comparison between C4.5 and Navie Bayes, where A_i = C4.5 and A_j = Navie Bayes, we have $ad(A_i - A_j) > 0$, therefore the Navie Bayes outperforms C4.5, but does not overcome the level of confidence of 80%, because $ad(A_i - A_j) < 1.29$.

Although we have achieved good results with our experiments, we verified three treats to validity of our work: i) The small number of volunteers (15) for evaluate the video lectures; ii) The limited domain and limited dataset; iii) The limited number of attributes - in our work we used only nine attibutes. We pretend to perform new experiments increasing the number of attributes, such as: analysis of the subtitles, audio and image quality, type of lesson (theoretical or problem solving), resources used in the video lecture (blackboard, slides or pen and paper), among others.

4. CONCLUSION AND FUTURE WORK

In this work we present an analysis of classifiers to predict the quality of video lectures. We conducted experiments with the classifiers: Navie Bayes, SVM and C4.5. The classifier that showed the best performance was the SVM, that was selected as a predictive model. We conducted an experiment and preliminary results indicate good quality of the SVM as prediction model. In our future work, we will conduct experiments with more users and videos. The analysis performed in this paper is part of an initial work to build a predictive model to determine the quality of video lectures. We plan to improve the prediction model with other factors such as context, viewing history, audio quality and relationships between user can be used to provide better results. In the future, the authors plan to integrate this predictive model in a recommendation system of video lectures.

5. REFERENCES

[1] S. Boll. Multitube–where web 2.0 and multimedia could meet. *IEEE MultiMedia*, 14(1):9–13, Jan. 2007.

[2] R. R. Bouckaert, E. Frank, M. A. Hall, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. WEKA–experiences with a java open-source project. *Journal of Machine Learning Research*, 11:2533–2541, 2010.

[3] X. Zhao, H. Luan, J. Cai, J. Yuan, X. Chen, and Z. Li. Personalized video recommendation based on viewing history with the study on youtube. In *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service*, ICIMCS '12, pages 161–165, New York, NY, USA, 2012. ACM.

Accepting or Rejecting Students' Self-grading in their Final Marks by using Data Mining

J. Fuentes
Department of Computer Science, University of Cordoba, Spain
javier.fuentes@uco.es

C. Romero
Department of Computer Science, University of Cordoba, Spain
cromero@uco.es

C. García-Martínez
Department of Computer Science, University of Cordoba, Spain
cgarcia@uco.es

S. Ventura
Department of Computer Science, University of Cordoba, Spain
sventura@uco.es

ABSTRACT

In this paper we propose a methodology based on data mining and self-evaluation in order to predict whether an instructor will or will not accept the students' proposed marks in a course. This is an on-going work in which we have evaluated the usage of classification techniques and cost-sensitive corrections. We have carried out several experiments using data gathered from 53 computer science university students.

Keywords

self-grading, self-evaluation, cost-sensitive classification.

1. INTRODUCTION

Assigning appropriate grades to students is an arduous and difficult process for instructors. Grades are, by their nature, somewhat subjective; every instructor uses different criteria to assign them and place a different emphasis on them. And with trends in higher education moving toward large class sizes, yet simultaneously toward more personalised and individualised instruction, self-grading may facilitate the achievement of these two objectives [3]. However, the main disadvantage of self-grading is grade inflation, that is, normally, more students, particularly among younger students, grading themselves higher than what they should get [4]. Roughly speaking, students' self-gradings are satisfactory substitutes for teacher gradings, if these two measures are comparable. If a student's grades were very different from the teacher's judgment, then the teacher should supervise and thoroughly evaluate the work, activities, and/or exams. Following this idea, in this paper we are interested in predicting what the instructor's decision is concerning the possible acceptance of the students' proposed final marks in a course. To do that, we use a methodology based on classification and self-evaluation checklists [1].

2. METHODOLOGY

The methodology that we have used in this study is as follows. During the course, students are evaluated by means of a multiple choice testing that is an effective assessment technique. Before the final exam date, all students are requested to self-grade. Students propose the mark/grade that they think they should get for the course. Then, the instructor accepts or declines the proposed mark of each student as the final mark for the course. This way, only students whose score was declined by the instructor will have to sit the final exam. Finally, we try to predict the instructor's decision of accepting or declining the score proposed by students. We have used two different (initial and new) data mining approaches (see Figure 1).

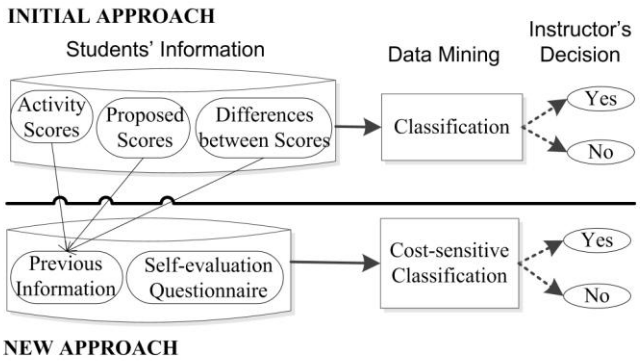


Figure 1. Approaches for predicting instructor's decision.

The initial approach uses three numerical variables: the score obtained by students in the course's activities, the proposed scores by students and the difference between these two previous scores. Then, it applies traditional classification algorithms for predicting the instructor's decision about whether to accept the proposed students' scores (YES) or not (NO). The new approach uses the three previous variables as well as a self-evaluation questionnaire as another source of information. Then, it applies cost-sensitive classification [2] that is normally used for obtaining better performances than traditional classification with unbalanced datasets. In fact, in our particular problem we are much more interested in the correct classification of NO (normally the minority class) than YES (the majority class). To do that, costs can be incorporated into the algorithm and considered during classification. In the case of two classes, costs can be put into a 2 × 2 matrix in which diagonal elements represent the two types of correct classifications and the off-diagonal elements represent the two types of errors. This matrix indicates that it is *N* times more important to correctly classify NO than YES students.

3. DATASET

We have used a dataset collected from second year university Computer Science students in 2012-13. During a traditional, face-to-face course on artificial intelligence, the instructor gave the students the option to self-grade. Out of the 86 students enrolled in the course, 53 accepted to self-grade, approximately 60%. For each one of these 53 students, we gathered the next attributes:

- **Activities score**. This is the average score obtained by students in three activities undertaken during the course. The three activities were Moodle multiple-choice tests with 10 questions, available at different moments of the course. The activities score of each student is a number between 0 and 10 points that is the average of the three activities.

- **Proposed score**. This is the final mark/score that the students believe that they should get in the course. Students themselves proposed their marks (number between 0 and 10).
- **Difference between scores**. It is the difference between the two previous scores. It is a positive or negative value (between -10 and +10) obtained automatically as the activities score minus the proposed score.
- **Self-evaluation questionnaire score**. This is the score obtained in a self-evaluation questionnaire. We have used a self-evaluation questionnaire developed at the University of Ohio (USA) [5]. It contains 50 yes/no questions for determining whether a student is a good or poor student. The students completed the questionnaire two weeks before the final exam date. The University of Ohio also provides a template with the responses of good students. Using this template we have calculated a score for each student as the number of answers equal to those of the good students.

The output attribute or class to predict in our problem is the instructor's decision. It is a binary value: YES or NO, that indicates whether the instructor accepts or declines the students' proposed scores. The instructor provided us with this value for each one of the 53 students: 37 YES (70%) and 16 NO (30%).

4. EXPERIMENTS

We have carried out several experiments in order to test our proposed methodology for predicting the instructor's decision. In these experiments we have used 35 classification algorithms provided by Weka 3.7: NaiveBayes, NaiveBayesSimple and NaiveBayesUpdateable, Logistic, RBFNetwork, SimpleLogistic, SMO, SPegasos, VotedPerceptron, MultilayerPerceptron, IB1, IBk, KStar, LWL, ConjunctiveRule, DecisionTable, DTNB, JRip, NNge, OneR, PART, Ridor, ZeroR, ADTree, BFTree, DecisionStump, FT, J48, J48graft, LADTree, LMT, NBTree, RandomForest, RandomTree, REPTree and SimpleCart. We executed all the algorithms using 10-fold cross-validation and their default parameters. Three classification performance measures were used to test the algorithms' results: Accuracy, True Positive rate (TP rate) or sensitivity, and True Negative rate (TN rate) or specificity. Figures 2 shows the obtained average values of all algoritms when using the initial and new approach with different values of cost (*N* = 1, 2, 3, 4 and 5).

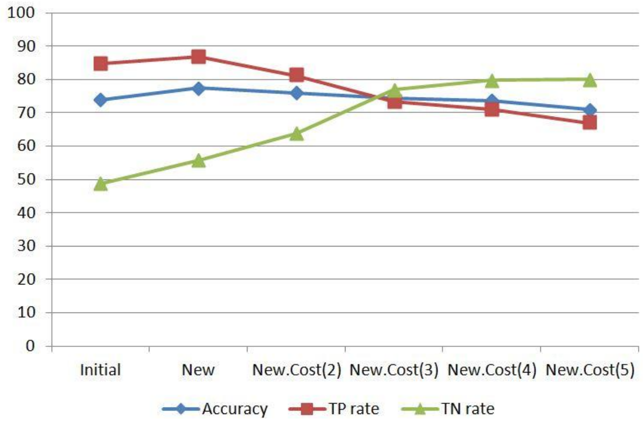


Figure 2. Traditional/Initial classification versus New/Cost-sensitive classification performance.

We can see in Figure 2 that the new approach improved the initial approach in the three evaluation measures and so, the self-evaluation questionnaire has shown to be a good source of information. However, TN rate continued at a very low values and so, we applied different costs for improving it. In Fact, Figure 2 also shows that when we increase the cost/weight of correctly classified NO students, it increases the TN rate. However, it also decreases the accuracy and TP rate. So, it is necessary to select the best *N* value in our problem in which TN rate improves without affecting accuracy and TP rate very much. For example, in our case, we can see that in Figure 2 an agreement/good solution is for *N*=3 in which the three measures cross its values.

Finally, we show an example of a model obtained with one of the classification algorithms. We have selected the output of the J48 algorithm (see Figure 3) because it obtained one of the best performances and it is also a well-known white-box classification algorithm. Using the discovered IF-THEN rules, instructor can make decision about which students accept or not their scores.

```
IF Difference-Between-Scores >= 0 THEN Decision=YES
ELSE IF Difference-Between-Scores < 0
 AND Proposed-Score <= 5 THEN Decision=YES
 ELSE IF Proposed-Score > 5
      AND Self-evaluation-Score >=5.9 THEN Decision=YES
      ELSE IF Self-evaluation-Score <5.9 THEN Decision=NO
```

Figure 3. Example of obtained decision tree.

5. CONCLUSIONS

Regarding the performance of the prediction of the instructor's decision, the results obtained show that the use of self-evaluation, and cost-sensitive classification improved the accuracy, sensitivity and specificity in our dataset. Our final objective is to use it as a time-saving scheme because only the rejected students (whose score the instructor does not accept) have to sit the final examination at the end of the course. Currently, we are carrying out more experiments with a greater number of students of different university courses. In the future, we want also work in the calibration task, which refers to how accurately individuals can predict how well they do on a task.

6. ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Science Technology project, TIN-2011-22408, and FEDER funds.

7. REFERENCES

[1] Andrade, M.R. Monitoring student performance with self-evaluation checklists: an ongoing case study. http://www.jrc.sophia.ac.jp/courses/pdf/kiyou2701.pdf

[2] Elkan, C. The foundations of cost-sensitive learning; International Joint Conf. on Artificial Intelligence, 1-6. 2001.

[3] Strong, B., Davis, M., Hawks, V. Self-grading in large general education classes. Colleague Teaching, 52(2), 52-57. 2004.

[4] Tan, K. Qualitatively different way of experiencing students self-assessment. Higher Education Research&Development. 52: 52-57. 2008.

[5] Test de Autoevaluación para Estudiantes Secundarios y Universitarios.http://www.tecnicas-de-estudio.org/general/sabes-estudiar.htm

Analysis and extraction of behaviors by students in lectures

Eiji Watanabe
Konan University
Kobe, Japan
e_wata@konan-u.ac.jp

Takashi Ozeki
Fukuyama University
Fukuyama, Japan

Takeshi Kohama
Kinki University
Kinokawa, Japan

ABSTRACT

In this paper, we discuss the influence the following behaviors on the behavior by a specific student in lectures; (i) the behavior by the lecturer, (ii) the behaviors by other students, and (iii) the behavior by oneself. First, we detect features for behaviors by lecturer and students by using image processing methods. Next, the relations among the above features are approximated by neural networks. Finally, we analyze the interaction between behaviors by lecturer and students based on the internal representations and show the synchronization between students.

Keywords

Lecture, Lecturer, Student, Behaviors, Time series model

1. INTRODUCTION

In lectures, the change of behaviors (writing on the blackboard and explaining) by the lecturer play important roles on the interests and the understanding by students [1]. Authors have already discussed the relationship between behaviors by lecturers and students by using a neural network [3]. However, since students can see behaviors by other students, we should focus on the relations among students.

In this paper, we construct time-series models for the interaction between behaviors by lecturer and students by using neural networks [2]. Concretely, we detect behaviors by using image processing methods and construct a time-series model for their behaviors. Finally, we analyze the interaction between behaviors by lecturer and students based on the internal representations in neural networks.

2. ANALYSIS OF BEHAVIORS BY STUDENTS

Students listen to the explanation by the lecturer and look at contents on the blackboard. Moreover, a student can see behaviors by neighbor students and students are influenced by behaviors by other students. Furthermore, students are listening to the lecturer and taking notes at their own paces. As shown in Figure 1, we can summarize the influences on behaviors by students as follows; (i) behaviors (explanation and writing) by lecturer, (ii) behaviors (listening and writing) by other students, (iii) behavior by oneself.

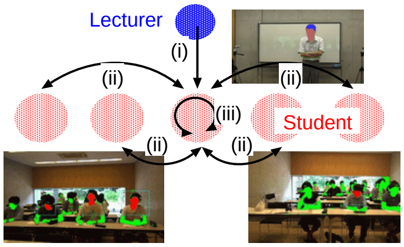


Figure 1: Influences on behaviors by students

2.1 Analysis of behaviors by students

Features for behaviors by lecturers and students can be extracted by image processing methods based on the color information [3]. Here, we define features as follows; the head position of the lecturer: $x^{L}_{\mathrm{head}}(t)$, the number of skin-colored pixels in the face region of the lecturer: $x^{L}_{\mathrm{face}}(t)$, and the number of skin-colored pixels in the face region of the p-the student: $x^{S,p}_{\mathrm{face}}(t)$.

2.1.1 Input-output relation of behaviors by students

Based on Figure 1, we approximate the number $x^{S,p}_{\mathrm{face}}(t)$ of skin-colored pixels in the face of the p-th student by the head position (horizontal) $x^{L}_{\mathrm{head}}(t)$ of the lecturer, the number $x^{L}_{\mathrm{face}}(t)$ of skin-colored pixels of the lecturer, and the number $x^{S,q}_{\mathrm{face}}(t)$ of skin-colored pixels of the q-th student as follows;

$$\begin{aligned} x^{\mathrm{S,p}}_{\mathrm{face}}(t) &= \alpha^{L}_{\mathrm{face}} f(\sum_{\ell} w^{L}_{\mathrm{face},\ell} x^{L}_{\mathrm{face}}(t-\ell)) \\ &+ \alpha^{L}_{\mathrm{head}} f(\sum_{\ell} w^{L}_{\mathrm{head},\ell} x^{L}_{\mathrm{head}}(t-\ell)) \\ &+ \alpha^{S,p}_{\mathrm{face}} f(\sum_{\ell} w^{\mathrm{S,p}}_{\mathrm{face},\ell} x^{\mathrm{S,p}}_{\mathrm{face}}(t-\ell)) \\ &+ \sum_{q\neq p} \alpha^{S,q}_{\mathrm{face}} f(\sum_{\ell} w^{\mathrm{S,q}}_{\mathrm{face},\ell} x^{\mathrm{S,q}}_{\mathrm{face}}(t-\ell)) + e(t), \quad (1) \end{aligned}$$

where $q \neq p$ and ℓ denotes the time delay. We assume that $e(t)$ is a sequence of Gaussian noise. Here, $\{\alpha\}$ denote weights for features by the lecturer and students and $\{w\}$

denote weights for the time-delay of features. Moreover, $f(\cdot)$ denotes a sigmoid function.

2.1.2 *Learning of the input-output relation by using a neural network model*

For the approximation of Eq. (1), we use a neural network model as shown in Figure 2. Obviously, the connection between input and hidden units is sparse and such a connection intends to the clarification of the role of weights α for each feature. Moreover, weights $\{w\}$ between input and hidden layers play the role of the clarification of the time-correlation among features. The object for the learning for a neural network model in Figure 2 is to minimize E.

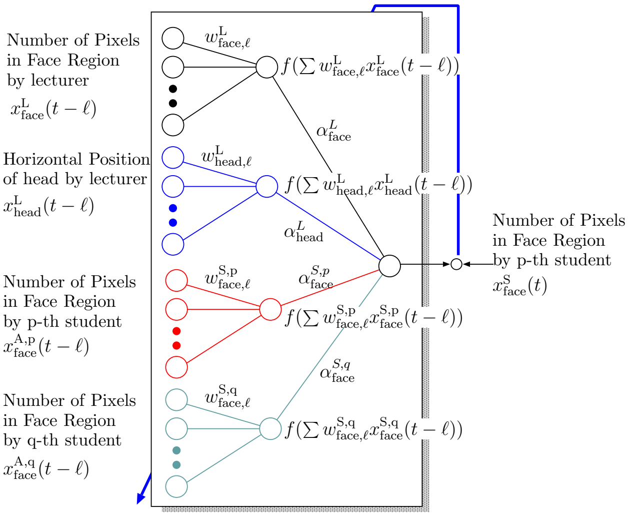

Number of Pixels
in Face Region
by lecturer
$x^{\mathrm{L}}_{\mathrm{face}}(t-\ell)$
$w^{\mathrm{L}}_{\mathrm{face},\ell}$
$f(\sum w^{\mathrm{L}}_{\mathrm{face},\ell} x^{\mathrm{L}}_{\mathrm{face}}(t-\ell))$
$\alpha^{L}_{\mathrm{face}}$
Horizontal Position
of head by lecturer
$x^{\mathrm{L}}_{\mathrm{head}}(t-\ell)$
$w^{\mathrm{L}}_{\mathrm{head},\ell}$
$f(\sum w^{\mathrm{L}}_{\mathrm{head},\ell} x^{\mathrm{L}}_{\mathrm{head}}(t-\ell))$
$\alpha^{L}_{\mathrm{head}}$
Number of Pixels
in Face Region
by p-th student
$x^{\mathrm{S}}_{\mathrm{face}}(t)$
Number of Pixels
in Face Region
by p-th student
$x^{\mathrm{A,p}}_{\mathrm{face}}(t-\ell)$
$w^{\mathrm{S,p}}_{\mathrm{face},\ell}$
$\alpha^{S,p}_{\mathrm{face}}$
$f(\sum w^{\mathrm{S,p}}_{\mathrm{face},\ell} x^{\mathrm{S,p}}_{\mathrm{face}}(t-\ell))$
$\alpha^{S,q}_{\mathrm{face}}$
Number of Pixels
in Face Region
by q-th student
$x^{\mathrm{A,q}}_{\mathrm{face}}(t-\ell)$
$w^{\mathrm{S,q}}_{\mathrm{face},\ell}$
$f(\sum w^{\mathrm{S,q}}_{\mathrm{face},\ell} x^{\mathrm{S,q}}_{\mathrm{face}}(t-\ell))$

Figure 2: Neural network for approximation of the input-output relation defined by Eq. (1)

$$E = \sum_t E_t = \sum_t (x^{\mathrm{S,p}}_{\mathrm{face}}(t) - \hat{x}^{\mathrm{S,p}}_{\mathrm{face}}(t))^2, \tag{2}$$

where $\hat{x}^{\mathrm{S,p}}_{\mathrm{face}}(t)$ denotes the prediction value for $x^{\mathrm{S,p}}_{\mathrm{face}}(t)$. The learning law for weights $\alpha^{L}_{\mathrm{face}}$ can be represented by

$$\alpha^{L}_{\mathrm{face}} = \alpha^{L}_{\mathrm{face}} - \eta \frac{\partial E_t}{\partial \alpha^{L}_{\mathrm{face}}}, \tag{3}$$

where η denotes the learning coefficient. On the other hand, the learning law for weights $w^{L}_{\mathrm{face},\ell}$ can be represented by

$$w^{L}_{\mathrm{face},\ell} = w^{L}_{\mathrm{face},\ell} - \eta \frac{\partial E_t}{\partial w^{L}_{\mathrm{face},\ell}}. \tag{4}$$

3. EXPERIMENTAL RESULTS

We have recorded images (640×360 [pixels], 10 [fps]) for four lecturers and five students in lectures concerning on the derivation of the formula for some trigonometric functions. Table 1 shows weights α between hidden and output layers. These weights denote the strength for each feature in Eq. (1) and we can show the followings;

- The behavior by Student-1 is strongly influenced by the change of the face of oneself from the relation between $\alpha^{L}_{\mathrm{head}} = 0.73$ and $\alpha^{S,p}_{\mathrm{face}} = 1.35$ for Student-1.
- Weights $\alpha^{S,1}_{\mathrm{face}}$ by Student-1 satisfy $|\alpha^{S,1}_{\mathrm{face}}| > |\alpha^{S,q}_{\mathrm{face}}|$ for all lecturers. This means that the behavior by Student-1 is not influenced by behaviors by other students.
- Weights $\alpha^{S,3}_{\mathrm{face}}$ by Student-3 satisfy $|\alpha^{S,3}_{\mathrm{face}}| < |\alpha^{S,2}_{\mathrm{face}}|$ for all lecturers. This means that the behavior by Student-3 is strongly influenced by the behaviors by Student-2.
- Weights $\alpha^{S,p}_{\mathrm{face}}$ satisfy the relations $|\alpha^{S,p}_{\mathrm{face}}| > |\alpha^{L}_{\mathrm{face}}|$ and $|\alpha^{S,p}_{\mathrm{face}}| > |\alpha^{L}_{\mathrm{head}}|$ for all students. This means that the behaviors by students are strongly influenced by them rather than the behaviors by lecturers.

Table 1: Weights between hidden and output layers (Results for other lecturers are omitted due to limitations of space)

(a) Lecturer-1

Student	$\alpha^{S,p}_{\mathrm{face}}$					$\alpha^{L}_{\mathrm{head}}$	$\alpha^{S,p}_{\mathrm{face}}$
	A	B	C	D	E		
A	$\underline{-1.57}$	-0.26	0.43	-0.02	0.40	0.73	1.35
B	-0.68	$\underline{-1.39}$	-0.46	-0.33	-0.43	0.04	0.32
C	0.32	$\underline{2.31}$	0.34	-0.27	-0.39	-0.36	-0.35
D	0.50	0.28	1.20	$\underline{-2.60}$	-0.47	0.32	0.66
E	-0.25	0.31	-0.45	$\underline{-2.16}$	-1.13	-0.11	0.31

(b) Lecturer-2

Student	$\alpha^{S,p}_{\mathrm{face}}$					$\alpha^{L}_{\mathrm{head}}$	$\alpha^{S,p}_{\mathrm{face}}$
	A	B	C	D	E		
A	$\underline{1.87}$	0.33	-0.05	0.30	-0.24	-0.25	-0.31
B	0.81	$\underline{1.43}$	0.30	-0.26	0.26	0.52	0.26
C	0.29	$\underline{1.62}$	0.40	0.08	0.29	0.23	0.06
D	-0.26	0.29	$\underline{1.13}$	-1.06	-0.36	-0.24	0.31
E	-0.34	0.43	0.27	$\underline{-3.24}$	-0.64	-0.23	-0.30

4. CONCLUSIONS

In this paper, we have analyzed the interaction between behaviors by lecturer and students by using neural networks and shown the followings; (i) a specific student is strongly influenced by only the behavior by oneself, (ii) other students are strongly influenced by other students, (iii) all students are strongly influenced by the behavior by oneself and other students rather than lecturers.

5. ACKNOWLEDGMENTS

This work has been partly supported by the Grant-in-Aid for Scientific Research (C) from the Japan Society for the Promotion of Science (Grant No. 25330428).

6. REFERENCES

[1] T. Marutani, Y. Sugimoto, K. Kakusyo, and M. Minoh. Lecture context recognition base on statistical feature of lecture action for automatic video recording. In *IEICE Trans. Information and Systems*, volume 90-D, pages 2775–2786, 2007.

[2] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. *Parallel Distributed Processing*. MIT Press, 1986.

[3] E. Watanabe, T. Ozeki, and T. Kohama. Extraction of relations between lecturer and students by using multi-layered neural networks. In *Proc. of IMAGAPP 2011*. SCITEPRESS, 2011.

Analysis of Student Retention and Drop-out using Visual Analytics

Jan Géryk
CVT and KD Lab
Faculty of Informatics, Masaryk University
Brno, Czech Republic
geryk@fi.muni.cz

Lubomír Popelínský
KD Lab
Faculty of Informatics, Masaryk University
Brno, Czech Republic
popel@fi.muni.cz

ABSTRACT

Student retention is an important measure for higher education institutions. Exploration and interactive visualization of multivariate data without significant reduction of dimensionality remains a challenge. Visual analytics tools like Motion Charts show changes over time by presenting animations within two-dimensional space and by changing element appearances. In this paper, we present a new visual analytics tool intended for exploratory analysis of educational data. We also utilized the tool for analyzing the data in order to verify the hypothesis concerning student drop-out behavior. The hypothesis assumes the existence of a correlation between the changes of fields of study and the student retention.

Keywords

Student retention, student drop-out, visual analytics, motion charts, animation.

1. INTRODUCTION

Higher education institutions have a major interest in improving the quality and the effectiveness of the education. In [1], hundreds of higher education executives were surveyed on their analytics needs. Authors resulted that the advanced analytics should support better decision-making, studying enrollment trends, and measuring student retention. They also pointed out that management commitment and staff skills are more important than the technology. In [2], authors concluded that the increasing accountability requirements of educational institutions represent a key to unlocking the potential for analytics to effectively enhance student retention and graduation levels. The application of data mining techniques in higher education systems have some specific requirements not present in other areas, as pointed out in [3].

Effective analysis depends on the consistent and high-quality data. Exploration and interactive visualization of multivariate data without fundamental dimensionality reduction remains a challenge. Animations represent a promising approach to facilitate better perception of changing values. In [4], authors pointed out that animations help to keep the viewer's attention. Correctly designed animations significantly improve graphical perception at both the syntactic and the semantic levels, as concluded in [5]. However, visualizations are often engaging and attractive but a naive approach can confuse the analyst. Motion Charts represent an animated data presentation method which shows multiple elements and dimensions on a two dimensional plane, as described in [6].

Motion Charts allow exploring and formulating additional hypotheses, as well as it helps to easily identify hidden patterns and trends in the data. The variable mapping is one of the most important parts of the exploratory data analysis. Both the data characteristics and the investigative hypothesis should influence the selection of a variable mapping.

In this paper, we briefly describe the motivation and design of the enhanced version of the Visual Analytics (VA) tool EDAIME, firstly introduced in [7]. In the next section, we present several papers concerning with data analysis using Motion Charts. Subsequently, we describe the design of the tool. Then, we make use of the tool for analyzing educational data in order to verify the hypothesis concerning student dropout behavior. Finally, we conclude the paper with future work and summarize the conclusion of the results.

2. RELATED WORK

Number of papers concerning the Motion Charts has increased recently. In [6], authors incorporated examples using recent business and economic data series and illustrated how Motion Charts can tell dynamic stories. For the first analysis, they utilized data about Current Employment Statistics and presented differences between the perception of common static tables and graphs, and the dynamic manner of Motion Charts. They concluded that static presentation style serves well the purpose of relaying accurate and non-biased quantitative data to the analyst. They also emphasized that the benefit of Motion Charts lays in displaying rich multidimensional data through time on a single plane with the dynamic and interactive features. Users are allowed to easily explore, interpret, and analyze information behind the data. They concluded that the Motion Charts are an excellent and interesting way of presenting valuable information that may otherwise be lost in the data.

Beneficial feature for better visual perception of changes in time-series analysis is presented in [8]. Author emphasized both the benefits and the drawbacks of common data visualization methods, namely line chart and bar chart. Then, the author focused on dealing with issues with the time-series analysis. Subsequently, he presented capabilities of Motion Charts which are more suitable for this kind of analysis. Moreover, author stressed that patterns of change through time can take many meaningful forms and introduced new feature, called visual trails, designed for Motion Charts which allows seeing the full path that elements take from one point in time to another. He also demonstrated proposed improvements.

3. THE EDAIME TOOL

Two main challenges are addressed by the presented VA tool EDAIME. The tool enables visualization of multivariate data and the interactive exploration of data with temporal characteristics. Moreover, it is optimized to process educational data.

The main purpose of the tool is to increase the education effectiveness and the quality of the study. The motivation to develop an enhanced version of Motion Charts was to extend their abilities and to improve the expression capability to facilitate analysts to depict each student as the central object of interest. Moreover, the implementation enhances the portfolio of animations that express the student's behavior during their study more precisely. The main technical advantages over other implementations of Motion Charts are its flexibility, the ability to manage many animations simultaneously. Optimizations of the animation process were necessary, since even tens of animated elements significantly reduced the speed and contributed to the distraction of the analyst's visual perception.

The Force Layout component of D3 (http://d3js.org/) provides the most of the functionality behind the animations, and collisions utilized in the interactive visualization methods. Linearly interpolated values are calculated for missing and sparse data.

3.1 Analysis of Educational Data

The main aims to improve student retention and graduation levels, are closely connected with analyses of changes of the mode and changes of the field of study. We utilize the EDAIME tool for analyzing educational data in order to verify hypothesis concerning with student dropout behavior. The hypothesis supposes the existence of a correlation between the changes of fields of study and student retention.

The large elements that represent a particular field of study consist of small elements that represent individual students. Therefore, the size of the large elements corresponds to the number of students enrolled in a particular field of study. The size of the small elements corresponds to the number of credits gained in a particular semester of study.

Besides the study progress, animations are also utilized to express the study termination, the change of the mode of study and the change of the field of study. Dropout students turn red and fall down the chart in the semester when they left the studies. The stroke-width of the elements represents the state of the study and the element color represents the attributes of the study.

To verify the aforementioned hypothesis we examined educational data about students admitted to bachelor studies of the Faculty of Informatics Masaryk University between the years of 2006 and 2008. The semester number is mapped to time variable. The grade point average is mapped to x-axis. The average number of credits is mapped to y-axis. The number of gained credits is mapped to element size.

Motion Charts show that the number of students decreases in all fields of study besides applied informatics (BcAP) because it is frequent target of change for the students in the first two semesters. After that, the number of students decreases uniformly for all fields of study. It is visually clear that the majority of students change the field of study to BcAP. More precisely, the highest migration between two fields of study is from computer graphics (GRA) to BcAP. Analysis show that the most student dropouts occur in the freshmen year, but over the time the number of unsuccessful students decreases significantly. Motion Charts also reveal that the ratio of the number of successful students to the number of unsuccessful students is significantly higher for students that changed their field of study. The supposed correlation exists, but a further analysis with a different mapping is needed to better express the relation between the migration target and the study success.

4. CONCLUSION

Common data visualization methods have limitations in terms of the volume and the complexity of the processed data. Motion Charts are transparent methods that can present a good overview of the complex data and also enable analyst to observe interesting elements while the previous ones are still fresh in his or her memory.

In the paper, we have described the motivation and design of the VA tool EDAIME which is intended for exploratory analysis of educational data. We enhanced the concept of Motion Charts and successfully expanded it to be more suitable for such analyses. We have successfully employed it to verify the suggested hypothesis. A further in-depth analysis with different mapping of variables is needed to quantify the correlations more accurately. Despite the fact that common data visualization methods are quite beneficial, there are types of questions that cannot be examined using them. Since the questions involve quantitative relationship other than change through time.

The additional representation of the data gives the analyst more possibilities in exploring the data, but the additional functionality can also confuses the analyst. To verify user friendliness and usability of the tool, we will carry out a controlled experiment with two groups of users. They will use different VA tool and methods trying to understand the same dataset.

5. ACKNOWLEDGMENTS

We thank Michal Brandejs and Knowledge Discovery Lab for their assistance. This work has been partially supported by Faculty of Informatics, Masaryk University.

6. REFERENCES

[1] Goldstein, P. J. and Katz, R. N. 2005. Academic Analytics: The Uses of Management Information and Technology in Higher Education, ECAR Research Study Volume 8.

[2] Campbell, & Oblinger, D. 2007. Academic analytics. Washington, DC: EDUCAUSE Center for Applied Research.

[3] Delavari, N. and Phon-Amnuaisuk, S. and Beikzadeh, M. R. 2008. Data Mining Application in Higher Learning Institutions. Informatics in Education, 31-54.

[4] Tversky, B. and Morrison, J. B. and Betrancourt, M. 2002. Animation: Can It Facilitate? International Journal Human-Computer Studies, 247-262. DOI=10.1006/ijhc.2002.1017.

[5] Heer, J. and Robertson, G. 2007. Animated Transitions in Statistical Data Graphics. IEEE Transactions on Visualization and Computer Graphics, 1240-1247.

[6] Battista, V. and Cheng, E. 2011. Motion Charts: Telling Stories with Statistics. JSM Proceedings, Statistical Computing Section. Alexandria, VA: American Statistical Association, 4473-4483.

[7] Géryk, J. 2013. Visual Analytics by Animations in Higher Education. In Proceedings of the 12th European Conference on e-Learning ECEL 2013, 565-572.

[8] Few, S. 2007. Visualizing Change: An Innovation in Time-Series Analysis. In Visual Business Intelligence Newsletter.

Automatic Assessment of Student Reading Comprehension from Short Summaries

Lisa Mintz
University of Memphis
Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
lmintz@memphis.edu

Dan Stefanescu
University of Memphis
Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
dstfnscu@memphis.edu

Shi Feng
University of Memphis
Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
sfeng@memphis.edu

Sidney D’Mello
University of Notre Dame
Notre Dame
IN 46556
sdmello@nd.edu

Arthur C. Graesser
University of Memphis
Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
graesser@memphis.edu

ABSTRACT

This paper describes our research on automatically scoring students’ summaries for comprehension using not only text specific quantitative and qualitative features, but also more complex features based on the computational indices of cohesion available via Coh-Metrix and on Information Content (IC, a measure of text informativeness). We assessed whether human rated summary scores could be predicted by indices of text complexity and IC. The IC metric of the summaries was a better predictor of human scores than word count or any of the Coh-Metrix text complexity dimensions. This finding may justify the implementation of IC in future automated summary rating tools to rate short summaries.

Keywords Information Content, summarization, reading

1. INTRODUCTION

Summarizing content after reading a text is a well-established method of assessing comprehension [1]. Assessing students’ reading comprehension through summarization has many advantages over other methods, because summarization requires readers to actively reconstruct their mental representation of the text [2]. The purpose of the current study is to examine a method of automated summary grading using a small corpus of summaries written for a variety of texts. We explored the use of the computational linguistic tool Coh-Metrix [3], as well as informativeness of words to predict human rater’s scores of summaries.

Coh-Metrix is a computational linguistic tool developed to measure hundreds of indices related to syntactic complexity, text cohesion, lexical diversity, and other features of language and discourse [3]. Coh-Metrix’s five major dimensions of text complexity predict a number of psychological findings associated with comprehension, such as reading time and recall [4]. In this study we used CohMetrix to measure: *narrativity*, *syntactic simplicity*, *word concreteness*, *referential cohesion*, and *deep cohesion* (http://cohmetrix.memphis.edu) [4].

Information Content (IC) is a measure used by Resnik [5] to compute the informativeness of a concept in a hierarchical taxonomy such as WordNet [6]. IC relies on the assumption the informativeness of a concept is inversely dependent on its occurrence frequency: the more frequent a concept, the less informative it is. Resnik [5] computes the frequency of a concept *c* as the sum of the occurrence frequencies of the words defining the concept *c* and all the other words defining the subordinates. Once the occurrence frequency of a concept is defined, the IC value for each concept *c* is computed as the self-information measure of *c*:

$$IC(c) = \log\left(\frac{1}{P(c)}\right) = -\log(P(c)) - log\frac{\#c}{\sum_i \#c_i}$$

A method for transferring the IC values from concepts to words has been proposed [7]: a word is assigned the IC value corresponding to the most general concept that word can represent, which is the concept with the minimum IC value:

$$IC(w) = \min_{c|w\in c} IC(c)$$

This ensures that high IC values are only associated with informative words. We compute the IC of a text fragment as the sum of the IC values for the individual words occurring in that fragment. The resulting sum value can be used as a measure of informativeness of the entire text, or it can be normalized by the total number of words in that text. We experimented with both methods.

In this study, we asked human experts to rate a total of 225 summaries written after reading texts from different genres. Our goal was to use Coh-Metrix dimensions of text complexity and IC computed from WordNet to predict the human ratings beyond simple verbosity (word count). If successful, such an approach will allow us to estimate summary qualities without a gold standard or a large summary corpus. Thus, our algorithm would contribute to assessing summaries written for a variety of subject matters and text types.

2. METHOD

Seventy-five undergraduates from the University of Memphis participated in this study. We collected 73 texts of different genres on different topics, containing between 1000 and 1500 words (M = 1301.3, SD = 186.0). There were 24 Informational, 24 persuasive and 25 Narrative texts selected from various websites on the internet. The texts were measured on different levels of textual complexity and Flesch-Kincaid readability. The texts were each separated into multiple pages (screens) of 75-100 words each, keeping the original paragraphs and always ending on a sentence. Each participant read three texts, one from each genre. Each text was randomly selected from each genre. After reading each text, participants wrote a 75 to 100 word summary of the text that they just read. Thus, each participant wrote 3 summaries, one per genre. Three expert raters independently rated the summaries on a 1-4 scale for comprehension. Chronbach's alpha scores suggested high inter-rater agreement of α = .802 (N = 225).

4. RESULTS AND DISCUSSION

A Pearson-correlation analysis was conducted between the summary rating score, word count, and IC. Word count was included because previous research suggests a strong positive relationship between word count and perceived quality of writing. IC strongly correlated with word count, r = .952, n = 224 p < .001. Word count and summary score were strongly correlated, r = .562, n = 224, p < .001, with an r^2 of .316. A linear regression revealed that approximately 31.3% of the variance in comprehension score can be accounted for by the variance in word count (β = .559, SE = .001. F = 100.635, p < .0001). We found a strong correlation between IC and summary score, score, r = .617, n = 224, p < .001, with an r^2 of .377. A linear regression revealed that approximately 37.7% of the variance in comprehension score could be accounted for by the variance in IC (β = .614, SE = .004, F = 133.715, p < .0001). We found that IC explained 5.5% more of the variance in comprehension score than word count.

It was interesting to note that Coh-Metrix's dimension of deep cohesion was significantly correlated with IC (r = .22, n = 224 p < .01), but not with word count (r = .078, n = 224, p = .248). However, a multiple regression using word count and deep cohesion as predictors did not show significant contribution of deep cohesion as a predictor. The result suggests that although IC is highly correlated with word count, it is a better predictor of comprehension than word count, which suggests that summary scores are more than mere summary length. In the future, IC could possibly be implemented in automated summary grading tools to increase their accuracy in scoring summaries.

Table 1. Correlations between summary score, IC, word count and Coh-Metrix's indices of text complexity

Measure	Comprehension	IC	Word Count
Summary score	-		
IC	.617**	-	
Word Count	.562**	.952**	-
Deep Cohesion	.121	.222*	.078
Referential Cohesion	.001	.057	-.103
Syntactic Simplicity	-.045	-.008	-.180*
Word Concreteness	.019	.099	-.076
Narrativity	.006	.095	-.056

p<.01* p<.001**

5. CONCLUSION

In this study we attempted to use IC and the five dimensions of text complexity from Coh-Metrix to predict human ratings of summaries. Our results showed that surprisingly, the five dimensions of text complexity did not predict human ratings of comprehension from summarization. On the other hand, although IC was also highly correlated with word count, it explained more of the variance in comprehension score than word count. In future research we will explore using other linguistic indices as well as IC to predict summary scores on a larger corpus of summaries.

6. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (ITR 0325428, HCC 0834847, DRL 1235958) and Institute of Education Sciences (R305C120001). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. Our thanks to Eliana Silbermann, Megan Reed, Janay Stewart, and Mari Sanford for their assistance.

7. REFERENCES

[1] Madnani, N., Burstein, J., Sabatini, J., & O'Reilly, T. (2013). Automated Scoring of a Summary Writing Task Designed to Measure Reading Comprehension. In *Proceedings of the North American Association for Computational Linguistics Eighth Workshop Using Innovative NLP for Building Educational Applications* (pp. 163). Atlanta, Ga.

[2] Graesser, A. C., Wiemer-Hastings, P., & Wiemer-Hastings, K. (2001). Constructing inferences and relations during text comprehension. *Text representation: Linguistic and psycholinguistic aspects*, *8*, 249-271.

[3] McNamara, D.S., Graesser, A.C., McCarthy, P.M., Cai, Z. (2014). *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge, MA: Cambridge University Press.

[4] Graesser, A.C., McNamara, D.S., & Kulikowich, J. (2011). Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational Researcher, 40*, 223-234.

[5] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*. (IJCAI-95).

[6] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Cambridge, Mass.: MIT Press.

[7] Stefanescu, D., Banjade, R., Rus, V. (2014). *A Sentence Similarity Method based on Parsing and Information Content.* In Proceedings of CICLing 2014, Kathmandu, Nepal.

Building an Intelligent PAL from the Tutor.com Session Database - Phase 1: Data Mining

Donald M. Morrison, Benjamin Nye, Borhan Samei,
Vivek Varma Datla, Craig Kelly, & Vasile Rus
Institute for Intelligent Systems, University of Memphis

ABSTRACT

In this poster**,** we describe a new research project involving the analysis of nearly 250,000 human-human tutorial dialogue transcripts (in Algebra and Physics) supplied by Tutor.com, a leading provider of online tutorial services for children and young adults. This project involves training a panel of Subject Matter Experts (SMEs) recruited from among Tutor.com's expert tutors to hand-tag a "gold standard" training set of as many as 1,500 transcripts, involving hundreds of different tutors, and potentially totaling more than 100,000 separate utterances. The SMEs will use a theory-based coding scheme to classify utterances into *dialogue acts* and *mode switches*, i.e., dialogue acts that serve to initiate a change in dialogue mode. The resulting training set will be used to train a dialogue act classifier to automatically tag dialogue acts and modes in the remaining transcripts. Machine learning techniques will be used to discover patterns (e.g., sequences, clusters, Markov chains) associated with successful and less successful sessions, where success is measured by internal evidence of learning and also the learner and tutor ratings available in the transcript metadata. Due to the large number of sessions and tutors studied, this research promises to expand our understanding of the prevalence and types of strategies and tactics used by human tutors. Preliminary findings from this data set will be presented during the poster session.[1]

Keywords

Tutorial dialogue; Human tutoring; Data mining; Intelligent tutoring; Computational linguistics; Machine learning; Big data.

1. INTRODUCTION

In recent years, artificial intelligence researchers have begun to apply machine-learning techniques to the analysis of interaction logs generated by online, chat-based (keyboard-to-keyboard) tutorial systems, e.g., [1]. Generally speaking, this approach involves some combination of human tagging of session features (e.g., utterance types), automatic feature detection, and identification of sequential feature clusters. For example, in [1] the researchers tagged the various dialogue acts in a relatively small corpus of tutorial dialogue sessions, then used Hidden Markov Modeling to discover mixtures of dialogue acts associated with identifiable tutorial "modes" [2].

The work described here extends this research, focusing on a large database of nearly 250,000 transcripts of chat-based tutorial dialogues, a subset of a rapidly expanding database of more than 10 million sessions conducted to date by Tutor.com tutors. Our approach features hand-tagging of dialogue acts and mode switches in a training set consisting of more than 1,000 transcripts, the development of an automatic context-sensitive dialogue-act classifier, and a "top-down, bottom-up" cluster analysis aimed at identifying dialogue features associated with positive outcomes, as measured both by the participant quality ratings (available in the transcript metadata). Internal evidence of learning during sessions will also be considered, such as the tutor's feedback on student contributions or student expressions of new understanding, ("Oh, I get it now.")

2. CONCEPTUAL FRAMEWORK

The theory-based coding scheme we are developing views a tutorial dialogue as a special form of human conversation, a joint activity [3] consisting of a sequence of back and forth utterances, each of which represents one or more *dialogue acts* [1].

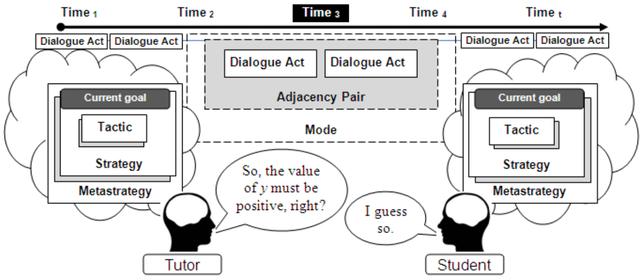


Figure 1. Anatomy of a tutorial dialogue

Dialogue acts are viewed as tactical choices, representing the interlocutors' hidden intentions and strategies, subject to biocultural constraints such as the need to establish common ground [4] and make contributions "relevant" [5]. As such, the significance of a given utterance must be understood in respect to previous utterances (e.g., *adjacency pairs*—[7]), and other higher-level organizational structures such as dialogue modes. Some dialogue modes, such as *openings* and *closings* [7], are common to most human conversation; others, such as *lecturing* and *collaborative problem-solving*, are characteristic of particular kinds of conversation, including the tutorial dialogues we focus on in this study. Successful tutorial dialogues, we hypothesize, are those in which the participants, both tutors and learners, manage to cooperatively align and accomplish their individual goals, drawing on sets of *tactics* (specific dialogue moves), *strategies* (algorithms or "policies" for selecting from among available tactics based on unfolding circumstances), and *metastrategies* (algorithms for selecting from among available strategies). This conceptual framework is explored more fully in related work [6].

[1] Corresponding Author: Donald M. Morrison (chipmorrison@gmail.com)

3. TUTORIAL TRANSCRIPT CORPUS

Our corpus consists of a set of 245,192 tutorial session transcripts shared with us by our partner, Tutor.com, a leading provider of online (chat-based) tutorial services. While this is only a subset of the 10 million (and counting) sessions available, we believe it is orders of magnitude greater than most prior analyses of human tutoring. The sessions represent attempts to help students solve problems and understand related concepts involving selected subtopics in Algebra (65% of the transcripts), and Physics (35%). The transcripts consist of more than 25 million time-stamped lines (corresponding roughly to utterances), representing more than 80,000 hours of dialogue, and containing more than 1,200,00 unique tokens (words and mathematical expressions). Each transcript is linked to a set of metadata, including both tutor and student ratings of session quality.

Table 1: Summary transcript statistics

Subject		**Mean**	**Std. Dev**	**Total**
Physics	Minutes	24.6	21.6	2,123,429
	Tutor lines	68.0	52.4	5,875,944
	Student lines	49.7	39.8	4,530,487
Algebra	Minutes	18.3	18.1	2,897,482
	Tutor lines	55.3	40.0	8,773,353
	Student lines	41.9	30.7	6,355,446

4. RESEARCH PLAN

This research involves five distinct development tracks, as summarized in Figure 1.

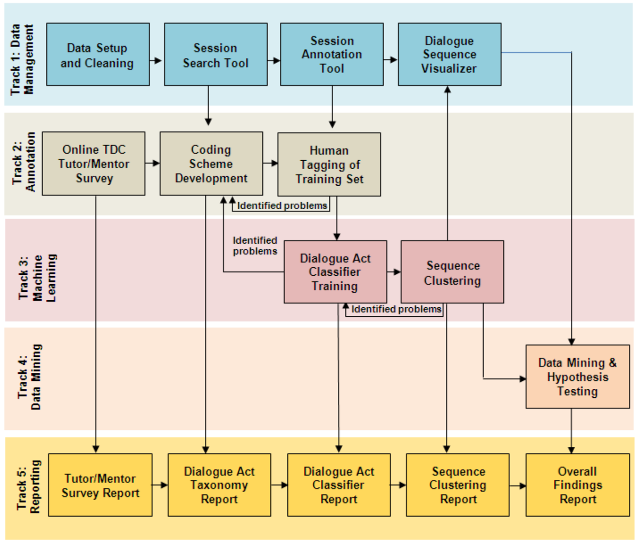

Track 1: Data Management
Data Setup and Cleaning
Session Search Tool
Session Annotation Tool
Dialogue Sequence Visualizer
Track 2: Annotation
Online TDC Tutor/Mentor Survey
Coding Scheme Development
Human Tagging of Training Set
Identified problems
Track 3: Machine Learning
Identified problems
Dialogue Act Classifier Training
Sequence Clustering
Identified problems
Track 4: Data Mining
Data Mining & Hypothesis Testing
Track 5: Reporting
Tutor/Mentor Survey Report
Dialogue Act Taxonomy Report
Dialogue Act Classifier Report
Sequence Clustering Report
Overall Findings Report

Figure 1: Research Plan Tracks

At this writing we are in the process of data cleaning and descriptive analysis, as well as development of the toolset we will use for session searches, annotation and visualization. One critical task is the development of a web-based annotation environment that will be used to train the human taggers, to hand-tag selected transcripts, and to review and revise transcripts tagged using automated tools.

We have also conducted an online survey of 250 Tutor.com tutors and tutor mentors, consisting of a set of open-ended questions aimed at eliciting the respondents' expert opinions regarding choices of particular tactics and strategies in different circumstances. We are using this data for two purposes: (1) to select a "blue ribbon" panel of tutors and mentors to serve as the Subject Matter Experts (SMEs); and (2) to ensure that our theory-based coding scheme is consistent with how the SMEs themselves think about the dynamics of the tutorial process.

A panel of 15 to 20 SMEs are being recruited to help modify the coding scheme, test the annotation environment, and hand-tag as many as 1,500 session transcripts for both dialogue acts and mode switches, i.e., dialogue acts that have the effect of turning on a particular mode ("Welcome to Tutor.com") or switching from one mode to another ("So, have you tried to do this problem yourself?").

Based on this training set, a dialogue act classifier will be tuned, which we will then use to auto-tag the remaining transcripts in the database. Finally, we will use sequencing and clustering algorithms to discover hidden patterns (interpretable as tactics and strategies) associated with successful and less successful sessions. Sequence-mining is one method we will use to detect patterns within sessions. Since Hidden Markov Modeling has a history of success for this type of analysis, we expect this to be one key technique [1][2]. Clustering will be applied to identify traits that characterize certain types of successful (or less successful) sessions.

The results of this data mining are intended to inform the design of future tutoring and adaptive learning systems. The project is the first phase in a planned multiyear research and development effort funded by the U.S. Department of Defense Advanced Distributed Learning (ADL), aimed at developing hybrid human and artificially-intelligent tutoring systems compatible with ADL's Personal Assistant for Learning (PAL) architecture.

5. ACKNOWLEDGMENTS

The research described here is supported by Tutor.com, an IAC company, under a contract with the U.S. Department of Defense Advanced Distributed Learning Initiative (W911QY-14-C-0019).

REFERENCES

[1] Boyer, K. E., Ha, E., Wallis, M. D., Phillips, R., Vouk, M. A., & Lester, J. C. (2009, July). Discovering tutorial dialogue strategies with Hidden Markov Models. In *AIED* (pp. 141-148).

[2] Cade, W. L., Copeland, J. L., Person, N. K., & D'Mello, S. K. (2008, January). Dialogue modes in expert tutoring. In *Intelligent tutoring systems* (pp. 470-479). Springer Berlin Heidelberg.

[3] Clark, H. H. (1996). *Using language*. Cambridge University Press.

[4] Garrod, S., & Pickering, M. J. (2007). Alignment in dialogue. *The Oxford handbook of psycholinguistics*, 443-451.

[5] Grice, P. 1975. Logic and conversation. *Syntax and semantics*, *3*, 41-58.

[6] Morrison, D.C. & Rus, V. (2014, to appear). Is it a strategy or just a tactic?: A Martian perspective on the nature of human pedagogical dialogue. In Sottilare, R., Hu, X., Graesser, A. and Goldberg, B. (Eds.) *Design Recommendations for Adaptive Intelligent Tutoring Systems: Adaptive Instructional Strategies, Volume II.* Army Research Laboratory.

[7] Schegloff, E. A., & Sacks, H. (1973). Opening up closings. *Semiotica*, *8*(4), 289-327

Building Automated Detectors of Gameplay Strategies to Measure Implicit Science Learning

[1] Elizabeth Rowe
EdGE @ TERC
Cambridge, MA
elizabeth_rowe@terc.edu

[2] Ryan S. Baker
Teachers College
Columbia University
New York, NY
baker2@exchange.tc.columbia.edu

[3]Jodi Asbell-Clarke &
[4] Emily Kasman
EdGE @ TERC
jodi_asbell-clarke@terc.edu
ekasman19@gmail.com

[5] William J. Hawkins
Worcester Polytechnic University
bhawk90@WPI.EDU

ABSTRACT

Educational games have the potential to be innovative methods of assessing learning. This research combines video analysis and educational data mining to measure the implicit science learning that takes place in games. By studying the video data from high school learners playtesting the game *Impulse*, we observed strategic moves that are consistent with an implicit understanding of relevant science concepts and reliably coded those moves in a sample of 69 high school students. This paper reports on work in progress to use educational data mining analyses that leverage coded video segments to build automated detectors of strategic moves from game log data.

Keywords

Automated detectors; Game-based learning; Implicit Science Learning; Game Strategies

1. INTRODUCTION

Nearly all youth and most adults participate in Internet-based games [1]. Games have been shown to foster scientific inquiry and problem-solving, and have enabled the public to participate in breakthrough scientific discoveries [2, 3]. As a result, many educators and researchers see digital games as key potential learning and assessment environments for the 21st century [4].

Our research group is designing web and mobile games that focus on high-school science concepts drawn from the U.S. standards for science education. These games use simplified game mechanics that emphasize the laws of nature and the principles of science. Players are able to dwell in scientific phenomena, building and solidifying their implicit knowledge over time.

It is not our intent that these games teach science content explicitly, but rather that they engage the learner with scientific phenomena in the effort to build their implicit understandings about these phenomena. To measure implicit learning in games, we explore the extent to which we can relate the development of strategies we see players building in the games to classroom learning of similar content. Thus, we address the question: *Do learners' strategic moves in the game correspond to increased implicit understanding of the science content driving the game mechanics?* Success in this design will result in a new way to think about game-based assessments, starting not from prescribed learning outcomes, but from watching what types of strategy development actually takes place. The first step of this research, reported in this paper, is to accurately predict the strategic moves that emerge in a physics-based game from the click data that is generated during gameplay.

2. THE GAME: *IMPULSE*

Our team designed the game *Impulse* to scaffold and measure players' implicit knowledge of forces and motions (Figure 1). In *Impulse,* particles have different masses and thus behave differently under the corresponding gravitational forces. Players use an impulse (made through a click) to apply a force to particles, with the goal of moving a specific particle to the goal while avoiding other ambient particles. If the player's particle collides with any ambient particle, she loses that round. In terms of the science, the player is immersed inside an N-body simulation with accurate gravitational interactions and elastic collisions among up to 30 ambient particles with varying mass.

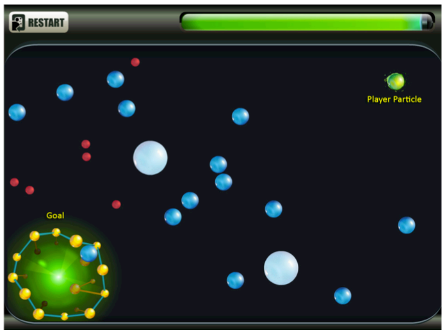


Figure 1: Impulse game

As players reach higher levels, they need to "study" the particles behavior to predict the motion of particles so that they can guide their particle to the goal, not run out of energy, and avoid collision with other particles.

Since there is no known best way for learners to build implicit understanding of these physics phenomena in games, our research captures the myriad of strategies players develop during gameplay. As a first step of this work, we have identified an initial set of strategic moves that we observe players making in the game *Impulse* that we theorize constitute evidence of implicit understandings of the underlying physics.

3. METHODS

Data were collected over six workshops conducted in March-June 2013 with 69 high school students (29 female) from urban and suburban schools in the Northeastern United States. Players were recorded with Silverback [5] that captures players' onscreen game activities and video of their faces and conversations. Students were asked to "think aloud" and explain their activities.

Two coders, a designer of Impulse with a physics background and a researcher without, independently watched the videos and coded two randomly selected three-minute segments from each player. The coding system was developed through repeated coding of hundreds of clicks with different play styles. A third coder with no physics background was trained using the coding system and coded randomly selected three-minute segments from all 69 videos. Two additional coders and one of the designers of the coding system double coded the segments from 10 videos. Table 1 includes definitions of the codes with Kappas exceeding 0.70.

Table 1. Video codes, definitions, and Kappas.

Code	Definition	Kappa
Float	The player particle was not acted upon for more than 1 second	0.759
Direction	The direction the learner intended the player particle to move	0.778
Target	Type of particle (player, other, both) the learner intended to move	0.920
Same as Last Target	The learner intended to move the same target as the last action	0.869
Intended strategy: Move toward goal	The learner intended to move the player particle toward the goal	0.809
Intended strategy: Stop/slow down	The learner intended to stop or slow the motion of the player particle	0.720
Intended strategy: Keep player path clear	The learner intended to move non-player particles to keep the path of the player particle clear	0.819
Intended strategy: Keep goal clear	The learner intended to move non-player particles to keep the goal clear	0.832
Intended strategy: Buffer	The learner intended to create a buffer between the player and other particles to avoid collision	0.772

Learner intentions are judged based not only on their screen actions, but also audio commentary and mouse over behaviors. Often players hold their mouse over spots, ready to click if needed, providing visible clues of their intended path. While not directly visible in the clickstream data, these behaviors are observable in video and aid interpretation.

The strategies identified through video analyses may provide evidence of players' implicit understanding of the mechanics related to Newton's first and second law. When a player uses a ***Float*** strategy, particularly when accompanied by a mouseover trailing along with the particle, the player is aware that an external force is not needed to keep the particle moving at a constant speed (Newton's First Law). Similarly, as evidence of an implicit understanding of Newton's Second Law, we are examining whether learners click more times when they are targeting particles of greater mass than they do of particles of lesser mass.

4. BUILDING AUTOMATED DETECTORS OF STRATEGIC MOVES

For each player action, a set of 66 features of that action are automatically distilled and aggregated at the click level to map to the labels provided by the video coders [6]. Classifiers for each code were created within RapidMiner 5.3 that map the student behaviors in the features distilled from the clickstream data to the training labels, using J48 decision trees with 4-fold cross-validation at the student level. (Kappa and A' values in Table 2).

Table 2: Detector Kappa and A' values

Code	Kappa	A'
Float	0.727	0.914
Intended strategy: Move toward goal	0.759	0.914
Intended strategy: Stop/slow down	0.522	0.804
Intended strategy: Keep player path clear	0.864	0.968
Intended strategy: Keep goal clear	0.772	0.943
Intended strategy: Buffer	0.756	0.928

5. CONCLUSIONS

During the playtesting of *Impulse,* we saw several strategic moves that are consistent with an understanding of Newtonian mechanics. Using the codes as ground truth, we are attempting to identify patterns in the clickstream data that predict players' strategic moves. These are early steps in developing an evidence model of implicit physics knowledge demonstrated via gameplay.

6. ACKNOWLEDGMENTS

We are grateful for NSF/EHR/DRK12 grant #1119144 and our research group, EdGE at TERC, which includes Erin Bardar, Teon Edwards, Jamie Larsen, Barbara MacEachern, and Katie McGrath. Our evaluators, the New Knowledge Organization, assisted with establishing the reliability of the coding.

REFERENCES

[1] Lenhart, A., et al. (2010) *Social Media & Mobile Internet Use Among Teens and Young Adults*. Washington, DC: Pew Internet & American Life Project.

[2] Steinkuehler, C. and Duncan, S. (2008). Scientific Habits of Mind in Virtual Worlds. *Journal of Science Education and Technology* 17(6): 530-543.

[3] Cooper, S., et al. (2010). Predicting protein structures with a multiplayer online game. *Nature, 466*(7307), 756-760.

[4] National Research Council. (2011). *Learning Science Through Computer Games and Simulations. Committee on Science Learning: Computer Games, Simulations, and Education*. M. Honey & M. Hilton (Eds.). Washington, DC: National Academies Press.

[5] Clearleft Ltd. (2013) Silverback (Version 2.0) [Software]. Available from http://silverbackapp.com.

[6] Sao Pedro, M.A., Baker, R.S.J.d., Gobert, J., Montalvo, O. Nakama, A. (2013) Leveraging Machine-Learned Detectors of Systematic Inquiry Behavior to Estimate and Predict Transfer of Inquiry Skill. *User Modeling and User-Adapted Interaction, 23* (1), 1-39

Challenges on applying BKT to model student knowledge in multi-context online learning environment

Wolney de Mello Neto Eduardo Barbosa Nicolau Wernerck

Felipe Garcia Leonardo Carvalho Pedro Carvalho

Geekie, Corp.
Sao Paulo, SP, Brazil
{wolney, eduardo.barbosa, nicolau, felipe.garcia, leonardo, pedro.carvalho}@geekie.com.br

ABSTRACT

Modelling student knowledge is a big challenge for online learning environments*(OLEs)*. One of the state-of-the-art models is the Bayesian Knowledge Tracing *(BKT)*, which estimates the probability of a student having learned a knowledge concept *(KC)* based on observable item answers over time. Nevertheless, BKT is based on a few assumptions that some real-world applications often struggle not to break, such as having homogeneous items presented to students in homogeneous contexts. Amongst other challenges pointed hereby, this poster focuses on the problem of having heterogeneous learning contexts. An experiment estimates multiple sets of parameters, one per learning context. The dataset is sampled from GeekieLab, an adaptive learning platform that is being used by more than 1 million Brazilian students.

Keywords

Bayesian Knowledge Tracing, Student Knowledge Modelling, Online Learning Environments, Adaptive Learning

1. INTRODUCTION

An online learning environment can be defined as a place where students can interact with content and/or people in order to achieve learning goals such as diagnosing knowledge gaps, learning new KCs and practicing those already known.

One major challenge in an OLE is how to measure the latent proficiency of each student in a given KC at some point in time. One could try to measure whether the student learned a KC or not, while other could try to measure how much the student knows of it. In both cases, the proficiency model should be continuously updated, that is, every interaction between the student and the platform might reflect on his proficiency, and most recent observations should have a stronger effect on calculating it. Finally, this model should estimate the probability of a student answering correctly the next item from some KC.

The BKT model[2] complies with the described requirements. Nevertheless, it comes with the expense of holding a few strong assumptions such as: (i) having a fine grained curriculum with KCs as specific as possible and dense answers data for each of them; (ii) providing homogeneous items that are related to only one specific KC; and (iii) collecting student answers from within **homogeneous learning contexts**, among many other. These are a few challenges that real-world tutoring systems face. This poster presents a brief discussion on (iii) and how online environments might not be able to hold this assumption.

2. HETEROGENEOUS CONTEXTS

The contextual effect on the BKT model has been broadly discussed in other studies, such as in [1] and [3]. In this poster, contexts refers to heterogeneous environments focusing on different stages of student learning, such as diagnosing, teaching and reinforcing.

The study case chosen for this paper is the online learning environment of GeekieLab[4], an online adaptive learning platform used by 1+ million Brazilian students. GeekieLab is a good example to illustrate the challenge discussed in this poster, since it is comprised of heterogeneous contexts where a student can answer to items. Some of its contexts are the following:

C_1 **lecture** short questions alternated with videos and slides;
C_2 **exercise list** set of questions without deadline, mainly for practice purposes;
C_3 **assessment** set of questions with short time-to-live. Exam-like environment accounted for grading.

2.1 Why should parameters be different?

As described in [2], BKT model estimates the probability $p(L)$ of a student having learned a KC by observing student answers based on a set P of the following parameters(probabilities):

$p(L0)$**:** student having learned a KC a priori;
$p(Transition)$**:** transition from unlearned to learned a KC between observations;
$p(Guess)$**:** unlearned state, but answer is right;
$p(Slip)$**:** learned state, but answer is wrong.

At first, a BKT application would estimate only one set $P_{C_{1,2,3}}$ of these 4 parameters for all observations of some KC. However, it looks more reasonable to update $p(L)$ with specific sets of parameters values (P_{C_1}, P_{C_2} and P_{C_3}) while observing item answers collected from their respective contexts C_1, C_2 and C_3. In order to investigate that, we will estimate parameters for each of the 3 contexts by training the model only with their respective observations.

3. EXPERIMENT

This experiment focuses on analyzing how parameters might vary among contexts. Upfront, the following hypotheses are proposed for further reflection:

H_1 regarding $p(S)$, it may get higher in C_3 since items get more tricky and due to the pressure of an assessment;

H_2 concerning $p(G)$, it may get higher in the case of a context where items provide easily detectable distractors. It could also increase in case one could have just watched a video or hint in C_1;

H_3 $p(T)$ might be higher for C_1 since a student is presented with an item resolution/hint in between two items. Moreover, assuming a student is supposed to learn a KC before the assessment (C_3), $p(T)$ might be lower within the latter;

H_4 assuming students face the contexts in the sequence C_1 lecture, C_2 exercise list and C_3 assessment, we expect to have an increasing $p(L0)$ throughout these them.

Based on those 3 contexts, a simple experiment was run aiming at testing the previously stated hypotheses. Further information on the data sample, estimation algorithm and method can be found in the following subsections.

3.1 Material

A sample has been extracted from GeekieLab[4]. 4 KCs were selected, each from a different domain field (Math, Portuguese Language, Natural Sciences and Human Sciences). For each of these 4 we have a data sample of 100k answers, in average ~3 answers per student, and 1 answer per item. Parameters estimation were run with BKT Brute Force algorithm shared by authors from [1].

3.2 Method

For each of the 4 KCs, its ~100k observations were divided into C_1, C_2 and C_3. BKT parameters were estimated per context and a full training set containing data from all contexts $C_{1,2,3}$ was used for training another set of parameters. All the search space, for earch parameter, is discretized by 0.01. Only $p(S)$ and $p(G)$ are upper-bounded by 0.1 and 0.3, respectively.

4. RESULTS AND DISCUSSION

Table 1 presents the results for estimating parameters per KC per context.

Results seem to be inconclusive for evaluating H_1 and H_2. On most scenarios, $p(S)$ and $p(G)$ are getting to the upper bound defined by the brute force implementation, intended to avoid model degeneracy. There are many possible causes to this symptom, such as noisy data from students answering without thinking fastidiously or items with easily recognizable distractors.

H_3 is somehow reflected in lecture (C_1) for KC_1 and KC_3, although KC_2 does not indicate the same. KC_4 can be discarded since there was only one item answer per student within C_1. This analysis draws attention to how important it is to filter answers for a KC from a student without some minimum number of answers, in case there should be some minimum value.

	context	p(L0)	p(T)	p(S)	p(G)	obs.
KC_1	C_1	0.06	0.20	0.1	0.3	36k
	C_2	0.46	0.03	0.1	0.3	36k
	C_3	0.34	0.15	0.1	0.20	36k
	$C_{1,2,3}$	0.27	0.13	0.1	0.29	109k
KC_2	C_1	0.77	0.041	0.1	0.3	36k
	C_2	0.33	0.001	0.1	0.3	35k
	C_3	0.33	0.231	0.1	0.3	37k
	$C_{1,2,3}$	0.58	0.001	0.1	0.3	109k
KC_3	C_1	0.73	0.09	0.1	0.3	36k
	C_2	0.46	0.02	0.1	0.3	35k
	C_3	0.13	0.08	0.1	0.3	36k
	$C_{1,2,3}$	0.47	0.04	0.1	0.3	107k
KC_4	C_1	0.79	0.001	0.02	0.16	8k
	C_2	0.51	0.18	0.1	0.3	36k
	C_3	0.21	0.54	0.1	0.3	21k
	$C_{1,2,3}$	0.54	0.24	0.1	0.3	65k

Table 1: BKT parameters estimation per context(C) for each knowledge concept(KC).

H_4 was rejected. Actually, it seems to be the contrary for KC_2, KC_3 and KC_4. C_1 has higher $p(L0)$ than C_2, which in turn has higher $p(L0)$ than C_3. This might be caused since students have better performance in C_1, tending to allow $p(L0)$ increase, indicating that the student already learned some KC.

A future investigation concerning this experiment scope could involve defining some heuristic for filtering and preprocessing the training dataset and executing this analysis on a larger dataset, with more answers per student, in order to achieve better results.

5. CONCLUSION

In general, results show that every context might need an exclusive set of parameters. In order reinforce this conclusion, an extension to the current experiment would be to evaluate model accuracy on estimating the correctness of the next answer for estimated parameters per context. This was left out of this poster due to size constraints.

All conclusions made hereby are based on simple criteria, but they manage to illustrate one of the challenging questions that one might face when implementing BKT.

6. REFERENCES

[1] R. d. Baker, A. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. *Intelligent Tutoring Systems*, 2008.

[2] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[3] R. S. d Baker, A. T. Corbett, S. M. Gowda, A. Z. Wagner, B. A. MacLaren, L. R. Kauffman, A. P. Mitchell, and S. Giguere. Contextual slip and prediction of student performance after use of an intelligent tutor. pages 52–63, 2010.

[4] Geekie. Geekie lab. available at http://www.geekielab.com.br/, May 2014.

Combining Statistical and Semantic Data Sources for the Improvement of Software Engineering Courses

Michael Koch, Markus Ring, Florian Otto, Dieter Landes
Coburg University of Applied Sciences and Arts
Friedrich-Streib-Straße 2
96450 Coburg, Germany
{michael.koch, markus.ring, florian.otto, dieter.landes}@hs-coburg.de

Abstract

The research project EVELIN aims at incrementally improving the quality of software engineering education. To this end, software engineering courses and their contents regularly need to be evaluated in order to provide meaningful feedback to lecturers with on the quality and its evolution. While statistical analyses of evaluation questionnaires with closed scaled questions can easily be done by most evaluation tools, open questions still need to be analyzed manually. Computer-based approaches to text analysis are still in their infancy and do not take advantage of software engineering domain knowledge. Although students' feedback is important, it is only one data source among many others such as transcribed interviews, exam statistics, or a course's entire context information. We argue that the combination of these data sources – especially natural language ones – in conjunction with innovative and semi-automated analysis techniques from, e.g., data mining and natural language processing will open up new opportunities for the improvement of software engineering courses.

Keywords

Software engineering education, multi-source analysis, natural language processing

1. Introduction

Software engineering offers a huge variety of methods and tools to design and realize complex software systems. Yet, software development also requires skilled individuals. Therefore, software engineering education and its improvement are paramount for being able to develop complex software systems successfully.

The research project EVELIN (Experimental improVEment of Learning software engINeering) tries to clarify which competencies students should possess and which didactical approaches are suitable to foster particular competencies. This includes continuous evaluation, reflection, and iterative adjustment and enhancement of software engineering courses.

Even though students' feedback – e.g. obtained by standardized questionnaires – is important for measuring educational quality, it is not the only data source to be considered. On the one hand, there are many statistical factors, like pass rate of a course, average, median, and standard deviation of exam grades, and scores of a given exam question. On the other hand, there are additional semantic data sources such as interview transcripts of graduates and industry partners, knowledge bases, or free-text feedback reports from students. It is also necessary to observe the evaluations' context with respect to positive or negative impact, as described in [2].

We aim at creating meaningful insights into the quality and quality evolution of a course by combining currently isolated heterogeneous data sources such as the ones mentioned above. In particular, we are searching for better ways to analyze and integrate natural language data sources for the iterative improvement of software engineering courses.

2. Related Work

Data mining approaches like [8] only focus on scaled questions when trying to find correlations between students' feedback and the lecturers teaching performance. Yet, they tend to neglect qualitative data sources.

Recent approaches like [1] try to analyze text-based questions, e.g. by determining the number of positive and negative statements related to different teaching aspect categories. While the classification of positive and negative statements is done automatically, the assignment to the corresponding teaching aspect categories is done manually.

Learning management systems may offer additional statistical data about students, their demographic background, their learning behavior, and engagement [5]. Especially in combination with online quizzes and exams, very detailed analytics are possible.

3. Methodology

It seems to be insufficient to focus only on a single data source like standardized questionnaires or drop-out quotes in order to measure the quality of a given course, especially when multiple additional sources are already available.

Due to legal and privacy reasons and our focus on in-situ courses, we can neither observe individual learning process of students at this level, nor do we have access to their personal data and social background, even though many promising research attempts on the latter are under way, trying to integrate data from social media platforms in the course evaluation.

Nevertheless we have access to exam questions and overall statistical parameters on results. Still, these data are very useful for analyzing the success of a course because exams and their questions may be linked to specific teaching goals and competencies.

Questionnaires and interview transcripts of different stakeholders are around for many years and evolve slowly, but constantly over time. Questionnaires contain a variety of open and scaled questions about the lecturer, the teaching method, the course material, the workload imposed on students, and possible improvements. Interviews aim at uncovering which competencies and skills are expected from students. These data sources are anonymous but can be attributed to the same student across multiple courses or project feedbacks by using an alphanumeric key. This offers the possibility of generating trend analyses over an extended period of time while respecting the students' privacy. Since a couple of semesters, we also collect post-mortem feedback reports of students in a software engineering capstone project. These short essays contain individual experiences including personal difficulties they encountered, gains, and feelings.

While common evaluation tools support statistical evaluations of scaled questions, only a few approaches like [1] and [7] tackle the computer-based evaluation of open questions. Qualitative feedback can be transformed into quantitative feedback by analyzing the relationships between statistical data and students' written responses [9].

Generic algorithms for text mining use word statistics and word correlations to identify meaningful propositions. Due to our focus on software engineering education, better results and a broader range of applications will be likely when domain knowledge is actually incorporated as guidance into text mining. To provide this domain knowledge we need to combine the terminology of software engineering with the terminology of educational processes, forming a suitable ontology for our needs. Promising sources of domain knowledge are – among many others –SWEBOK [3], the IEEE glossary of software engineering terminology [6], as well as ontologies for aspects like teaching programming. Manual or semi-automatic reviews of lecture notes should also be considered as additional knowledge source [4].

Even though the combination of multiple data sources – especially natural language based ones – is recommended and necessary to obtain a complete picture of a course, quantitative scores taken alone should not be underestimated and contain useful information about the relationships between students' feedback and teaching performance as described in [8].

Inspired by the approach described in [1], we want to automatically identify positive and negative statements of text-based feedback answers and assign them to a suitable category in the educational context. As a basis, the predefined categories of [1] may be used, namely Course in General, Instructor, Assessment, Material, Delivery, Equipment, Program, Schedule, and Teaching. While categories are assigned manually in [1], we want to automate this step by using available domain knowledge of the specific course. At least we want to automatically suggest a category by using domain knowledge – e.g. in form of a software engineering education ontology – and provide the ability of learning from manual assignments, especially when the system was not able to automatically suggest a suitable category. These cases may be also interesting with respect to revealing new rating aspects that were neglected or overlooked before. A fine-grained hierarchical categorization system based on the mentioned ontology may support a drill-down analysis, helping to find causalities for successes or failures, when necessary.

The categorized positive and negative statements may be used to supplement, validate, or even automatically adjust the statistical information gained from closed questions.

In addition, the detection of statements in feedbacks and exams that deviate significantly from the average or median can be very useful on multiple levels of the course evaluation.

4. Summary and Future Work

Teaching software engineering is difficult and needs to be continuously improved to stay current. However, this requires careful analysis of a variety of data sources, many of which are in textual format. In this contribution, we analyzed the potential that lies in of available course evaluation data sources and outlined how these sources might be combined with each other to yield meaningful insights into the quality of a course. In particular, we sketch how data mining might help to automatize this process.

In the future we will refine and realize our concepts. We will also consider designing our approach as generic as possible to be able to support other educational domains.

5. Acknowledgements

This research is supported by Bundesministerium für Bildung und Forschung under grant no. 01PL12022A.

6. References

[1] Abd-Elrahman, A., Andreu, M., Abbott, T. 2010. Using text data mining techniques for understanding free-style question answers in course evaluation forms. In *Research in Higher Education Journal*, 9, 12-23

[2] Alhija, F. N.-A., Fresko, B. 2009. Student evaluation of instruction: What can be learned from students' written comments? In *Studies in Educational Evaluation*, 35(1), 37-44.

[3] Bourque, P., Fairley, R. E. (Eds.) 2014. *Guide to the Software Engineering Body of Knowledge (SWEBOK): Version 3.0*, IEEE Computer Society Press, New York

[4] Gantayat, N. 2010. Building domain ontologies from lecture notes. Master Thesis. Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, India

[5] Hung, J.-L., Hsu, Y.-C., Rice, K. 2012. Integrating Data Mining in Program Evaluation of K-12 Online Education. In *Educational Technology & Society*, 13(3), 27-41

[6] Institute of Electrical and Electronics Engineers 1990. *610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology*.

[7] Jordan, D. W. 2011. *Re-thinking student written comments in course evaluations: Text mining unstructured data for program and institutional assessment*. Doctoral Thesis. California State University

[8] Mardikyan, S., Badur, B. 2011. Analyzing Teaching Performance of Instructors Using Data Mining Techniques. In *Informatics in Education*, 10(2), 245-257

[9] Sliusarenko, T., Clemmensen, L. K. H., Ersbøll, B.K. 2013. Text mining in students' course evaluations: Relationships between open-ended comments and quantitative scores. In *CSEDU 2013 - Proceedings of the 5th International Conference on Computer Supported Education*, 564-573

Comparing Learning in a MOOC and a Blended On-Campus Course

Kimberly F. Colvin
MIT, 26-331
77 Mass. Ave
Cambridge, MA 02139
617 324-4528
colvin@mit.edu

John Champaign
MIT, 26-331
77 Mass. Ave
Cambridge, MA 02139
617 324-4528
john.champaign@gmail.com

Alwina Liu
MIT, 26-331
77 Mass. Ave
Cambridge, MA 02139
617 324-4528
alwina@mit.edu

Colin Fredericks
MIT, 26-321
77 Mass. Ave
Cambridge, MA 02139
617 324-4528
colin.fredericks@gmail.com

David E. Pritchard
MIT, 26-241
77 Mass. Ave
Cambridge, MA 02139
617 253 6812
dpritch@mit.edu

ABSTRACT

We studied student learning in the MOOC “Mechanics ReView”, run on the edX.org open source platform as 8.MReV. We administered 13 conceptual questions both before and after the instructional period, analyzing the results using standard techniques for pre - post testing. Our students had a normalized gain slightly higher than typical values for a traditional course but lower than typical values for courses using interactive engagement pedagogy. All questions in the MOOC, including the pre-post test questions were analyzed using Item Response Theory (IRT). Both the normalized gain and the IRT results showed that initially low-skill cohorts learned as much as all cohorts with higher initial skills. We were able to compare MIT freshmen taking an on-campus course with the 8.MReV MOOC students because many common problems were administered to both groups. The freshmen were considerably less skillful than the 8.MReV students and showed no signs of closing the gap with the more experienced 8.MRev students while covering topics in common with the MOOC.

Keywords

MOOC, edX, Item Response Theory, learning gain

1. INTRODUCTION

The recent release of hundreds of free online courses in MOOCs (Massive Open Online Courses) by organizations such as coursera.com, edX.org, and udacity.com has been so dramatic that an article in the New York Times proclaimed 2012 the “Year of the MOOC” [4]. A central question remains: “is there learning in MOOCs?”

In this paper, we report an initial study of learning in a MOOC, 8.MReV – Mechanics ReView – offered from June 1 to August 27, 2013 on the open source platform edX.org. The course materials were written by the RELATE education group (REsearch in Learning, Assessing, and Tutoring Effectively, http://RELATE.MIT.edu). This is a “second course” in introductory Newtonian Mechanics, designed to help students familiar with the topic at a high school level gain a more expert-like perspective on the subject. In addition, we made a concerted effort to attract high school physics teachers to enroll in our course.

We used two major approaches to evaluate learning in a MOOC: (1) a pre- and posttest analysis on an identical set of, mostly conceptual, questions [5] and (2) an analysis of the overall and topic-by-topic performance using Item Response Theory (IRT). Given that the on-campus students also benefitted from four hours of instruction in a flipped classroom, we addressed the question of whether their skills increased week by week relative to those of the MOOC students.

2. DATA

2.1 Description of MOOC: 8.MReV

The 8.MReV course grew from a short Mechanics ReView course run at MIT, which used an online eText and pre-class homework questions. For the MOOC, these online materials were augmented by additional problems and weekly quizzes. The 8.MReV course studied here involved was delivered via the edX.org platform and in the summer of 2013 with both general and teacher-targeted publicity. The 1080 students who attempted more than half of the problems were included in this study.

2.2 Description of On-Campus Course: 8.011

The on-campus course, 8.011, is the spring version of Introductory Newtonian Mechanics at MIT. This subject, together with the subsequent Electricity and Magnetism course, are required of all MIT graduates, and most take it in their first semester. Students who earned less than a C in mechanics course required to retake the course before moving on to Electricity and Magnetism; these students make up about 80% of the population of 8.011. In Spring 2013, there were 47 students in 8.011, the first time the online segment of the course was run on the edX.org platform rather than LON-CAPA. Of these 47 students, 35 attempted more than half of the online problems. These 35 students were used in this study.

3. Pre-Post Testing in the MOOC

The pre- and posttests consisted of 15 questions, three of which came from the Mechanics Baseline Test [3] and four were from the Mechanics Reasoning Inventory [4]. These fifteen questions focused on conceptual knowledge more heavily than algebraic ability. The results were analyzed in terms of the fractional reduction in the number of incorrect answers on the pretest as measured by the posttest. This quantity is referred to as the normalized gain by Hake [1].

4. Item Response Theory (IRT)

IRT places students and items on the same scale, taking into account a student's specific performance on each item [2], even when students do not take the same set of items. Each item's difficulty and discrimination is taken into account. IRT stands in contrast to classical test theory whose unit of analysis is the entire test, usually graded by the total number of items correct.

5. PRE AND POST TEST RESULTS

The pre-posttest analysis was performed on several sets of questions, here we will report on two: (1) six questions involving force and motion that could be compared with Hake's study [1] and (2) five questions on more advanced topics.

Traditional analysis of pre-post testing requires students to have done all questions in that set on both tests, which limits the number of students in each cohort. The IRT-based pre to posttest change was a statistically significant increase for the 579 students who took at least 7 pre and posttest items.

We observed learning as measured by normalized gain that are greater than the traditional courses studied by Hake [1] (0.23) and less than the interactive courses (0.48). While both of our gains, 0.30 and 0.33 (+/-0.02) are closer to the traditional on-campus courses, they lie above *all* of the 14 traditional classes studied by Hake, suggesting that our students learn conceptual topics slightly better than in a traditional, lecture-based, class. When we examined the normalized gain for various cohorts, it is significant that we saw no cohorts significantly below or above the normalized gain for the whole group. This certainly should allay concerns that less well prepared students can't learn in MOOCs.

6. COMPARISON OF SKILLS BY COHORT

The skill distribution for all 8.MReV students has a mean of 0 and a standard deviation of 1, for convenience. Using the same scale, the teachers have a mean of 0.39 and a standard deviation of 0.97. The on-campus 8.011 students' skill averaged about one standard deviation below (-1.05) the overall average in 8.MReV and had less variation with a standard deviation of 0.50. In retrospect, this may not be surprising as the average 8.MReV student is better educated, older, and not juggling three or four other MIT courses.

We compared the topic-by-topic skills of various cohorts: teachers, on-campus students, strong background in mathematics, for example. In this analysis we were not looking at their absolute skills, which we knew to be different, but rather the pattern of the change in skills from one topic to the next. We wanted to know if perhaps a weaker cohort in terms of their overall skill, showed marked improvement throughout the course, for example. However, none of the cohorts showed a significant linear improvement across the topics.

7. CONCLUSIONS

It is also important to note the many gross differences between 8.MReV and on-campus education. Our self-selected online students are interested in learning, considerably older, and generally have more years of college education than the on-campus freshmen with whom they have been compared. The on-campus students are taking a required course and also had many ways to obtain help on the problems in addition to four hours of highly interactive class time. Moreover, there are more drop-outs in the online course (but over 50% of students making a serious attempt at the second weekly test received certificates) and these may well be students learning less.

In this MOOC, there was significant learning for the students studied here, in fact, slightly more than students in a traditional, lecture-based, on-campus classroom**.** It is also noteworthy that analyses of the pre-post testing using normalized gain and IRT approaches both provide evidence that students throughout the wide range of abilities in our course, including those of low ability, learn a comparable amount.

8. ACKNOWLEDGMENTS

We acknowledge support from a Google Faculty Award, MIT, and NSF. We thank Yoav Bergner for thoughtful comments on the manuscript.

9. REFERENCES

[1] Hake, R.R. 1998. Interactive-engagement vs traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*. 66 (1), 64-74.

[2] Hambleton, R. K., Swaminathan, H., & Rogers, H. J. 1991. *Fundamentals of item response theory*. Newbury Park, CA: Sage Publications.

[3] Hestenes, D. & Wells, M. 1992. A Mechanics Baseline Test. *The Physics Teacher*, 30, 159-165.

[4] Hestenes, D. Wells, M., & Swackhamer, G. 1992. Force Concept Inventory. *The Physics Teacher*, 30, 141-158.

[5] Pappano, L. (2012, November 2). The year of the MOOC. *The New York Times.*

[6] Pawl, A., Barrantes, A., & Pritchard, D. 2009. *Modeling Applied to Problem Solving.* Proceedings of the 2009 Physics Education Research Conference, M. Sabella, C. Henderson and C. Singh, (Eds.) pp. 51-54

[7] Pritchard, D. E., Lee, Y. J., & Bao, L. 2008. Mathematical learning models that depend on prior knowledge and instructional strategies. *Physical Review Special Topics-Physics Education Research*, *4*(1), 010109.

Cost-Effective, Actionable Engagement Detection at Scale

Ryan S. Baker, Jaclyn Ocumpaugh
Teachers College, Columbia University
525 W. 120th St., Box 118
New York, NY 10027 USA
1-212-678-8329 and 1-212-678-3854
baker2@exchange.tc.columbia.edu, jo2424@columbia.edu

ABSTRACT

Costs of educational measures and interventions have important real-world implications, made more pertinent when used at scale. Traditional measures of engagement (e.g., video and field observations) scale linearly, so that expanding from 10 classrooms to 100 can incur 10 times the cost. By contrast, the cost of applying an automated sensor-free detector of student engagement is independent of the size of the data set. While the development and validation of such detectors requires an initial investment, once this cost is amortized across large data sets, the cost per student/hour is quite modest. In addition, these detectors can be reused each year at minimal additional cost. In this paper, we provide a formal cost analysis of automated detectors of engagement for ASSISTments.

Keywords

Affective computing, sensor-free detection, ASSISTments, STEM, student engagement, cost-effectiveness

1. INTRODUCTION

Automated sensor-free detectors of student engagement are now available for several systems [3, 6], shifting the debate from whether this detection is possible to a discussion of the upper limits of its performance and generalizability (see [5]). Automated detectors have been used to drive interventions [1] and in discovery with models analyses [6, 8]. As researchers and policy-makers seek to identify the teaching methods and online learning systems that promote greater engagement, studies at considerable scale have become a priority [2]. Unfortunately, this scale is often achieved at considerable cost. An alternate option is to use EDM models on log files. With appropriately validated detectors, engagement among large numbers of students can be gauged rapidly, and as students continue using the same learning system, extensive, individualized data can be applied to interventions and to long-term predictions through discovery with models.

In this paper, we examine the cost of developing detectors of 7 constructs of student engagement for ASSISTments, outlining current expenses and applications. We include a brief description of their applicability towards discovery with models research, a technique that leverages such existing models, substantially increasing their worth.

2. ASSISTments

Detectors in this study were developed for ASSISTments [7], a formative assessment system that provides scaffolded math instruction and targeted hints. ASSISTments was developed at Worcester Polytechnic Institute and is available to educators at no cost. Typically, students spend 1 regular class period per week using ASSISTments, and some also use it for homework [3]. Currently, approximately 60,000 students use ASSISTments in schools throughout the Northeastern United States.

3. Methods

3.1 Overview of Detector Construction

For this study, we consider the cost of producing detectors for 7 different constructs, including 4 affective indicators of engagement (boredom, confusion, engaged concentration, and frustration) and 3 behavioral indicators of engagement (gaming the system, off-task behavior, and carelessness). As reported in previous research, different methods were used to obtain the ground truth labels used to develop these detectors.

For the 4 affective detectors and for 2 of the behavioral detectors (gaming the system and off-task behavior), ground truth labels were generated by BROMP-certified field observers [4]. Observers spent 379 hours in field, obtaining 5,564 observations of 590 students at 6 different schools. These observations were then used to train separate detectors for each construct, each of which was cross-validated at the student-level to ensure generalizability to new populations (e.g., [6]). Research shows that rural students' affect manifested differently in their interactions with ASSISTments compared to urban and suburban students, so affect detectors were constructed to reflect these demographic differences [5].

The construction of the carelessness detector was different than the other six detectors as no fieldwork was required. Instead, programmers used Bayesian Knowledge Tracing (BKT) algorithms to calculate Contextual Slip (e.g., [1]). Each time a student makes an incorrect answer on a problem, the probability that a student is making a careless error is calculated based on his or her previous performance on the same skill [8].

3.2 Calculation of Cost

Two major sources of funding were used to create the ASSISTments detectors. The first was a National Science Foundation award to the Pittsburgh Science of Learning Center for $100,000, used to develop initial detectors for ASSISTments and 4 other systems. The second, a grant from the Bill & Melinda Gates Foundation for $277,044 funded further enhancement and validation of the ASSISTments detectors and models for 2 other systems. Roughly, this means that the initial investment for cross-validated models of all seven constructs in ASSISTments (also tested across 3 populations) totaled $117,348 ($16,050 per construct or $7,823 per detector). A list of these detectors, their algorithms, and their performance metrics are provided in Table 1.

Table 1. Table captions should be placed above the table

Detector	Algorithm	Kappa	A'	r
Boredom (Urban)	Jrip	0.23	0.6	*na*
Boredom (Rural)	K*	0.24	0.7	*na*
Boredom (Suburban)	REPTree	0.19	0.7	*na*
Confusion (Urban)	J48	0.27	0.7	*na*
Confusion (Rural)	JRip	0.14	0.6	*na*
Confusion (Suburban)	REPTree	0.38	0.7	*na*
Engaged Concentration (Urban)	K*	0.36	0.7	*na*
Engaged Concentration (Rural)	REPTree	0.37	0.7	*na*
Engaged Concentration	J48	0.27	0.6	*na*
Frustration (Urban)	REPTree	0.29	0.7	*na*
Frustration (Rural)	JRip	0.2	0.6	*na*
Frustration (Suburban)	REPTree	0.17	0.6	*na*
Gaming the System	K*	0.37	0.8	*na*
Off-Task Behavior	REP-Tree	0.51	0.8	*na*
Carelessness	Linear	*na*	*na*	0.50

However, since opportunities to apply both interventions and discovery with analyses are predicated on the number of labels (not just the number of detectors), the cost per label is perhaps a better indicator of the value of this research. At present, these detectors have been applied to 231,543 hours of ASSISTments data produced by 54,401 students. For BROMP-trained detectors, a label has been applied to every 20 seconds of interaction within the system (41,677,740 intervals x 6 constructs = 250,066,440 labels), and carelessness labels have been applied to every problem incorrectly completed during that time (3,163,616 labels). This puts the current cost per label well under 1 penny ($0.00046/label), a price that will continue to drop over the years as these detectors are applied to new data.

The cumulative cost per student/hour, a calculation important to allocating financial resources in education, is also extremely low ($1.97) and becomes even lower when calculated as a cost per construct ($1.97/7 = $0.28). By comparison, even if we (incorrectly) assumed a single observer, paid the 2014 federal minimum wage of $7.25/hour, could replicate the granularity of this data, the cost would top $1.6 million. In reality, the minimum rate of a trained observer is likely closer to $25/hour (plus approximately 37% in benefits), totaling almost $8,000,000, and a 1-1 coder-student ratio would be needed. Such conditions would likely destroy the value of any data collected since classroom conditions would be so disrupted as to make any data meaningless, and using video coding to attempt to replicate this level of granularity would incur even further expenses.

4. DISCUSSION/IMPLICATIONS

Cost is not the sole criteria for evaluating educational research, but it is a necessary consideration when developing resources to be implemented at scale. In this report, we have discussed the costs involved in developing detectors for 7 measures of student engagement, demonstrating that their relative cost is quite low, particularly when amortized across the use of the detectors to label data. These costs will drop further still in the coming years.

Further research with these detectors demonstrates the long-term predictive prognostic power of these constructs, which can predict standardized test scores [6] and college attendance several years later [8], showing the importance of this granular data. Currently, we are working to make these predictions more accessible to educators, providing them with actionable reports about who most needs intervention and which student behaviors are most problematic. As such, these models are likely to be of value both for research and for practice, and as these detectors scale easily, interventions based on them will also.

5. ACKNOWLEDGMENTS

Thanks to the Bill & Melinda Gates Foundation and the National Science Foundation (#SBE-0836012) for support, and thanks to Fiona Hollands for useful suggestions. (All errors are ours.)

6. REFERENCES

[1] Cooper, D. G., Arroyo, I., & Woolf, B. P. (2011). Actionable affective processing for automatic tutor interventions. In *New Perspectives on Affect and Learning Technologies* (pp. 127-140). Springer New York.

[2] Corbett, A.T., and Anderson, J.R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 4, 253-278.

[3] D'Mello, S. K., Craig, S. D., Witherspoon, A., Mcdaniel, B., & Graesser, A. (2008). Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction*, *18*(1-2), 45-80.

[4] Kane, T. J., McCaffrey, D. F., Miller, T., & Staiger, D. O. (2013). Have We Identified Effective Teachers? Validating Measures of Effective Teaching Using Random Assignment. *Bill & Melinda Gates Foundation*.

[5] Mendicino, M., Razzaq, L., & Heffernan, N. T. (2009). A comparison of traditional homework to computer-supported homework. *J. Research on Comp. in Education*, *41*, 3, 331.

[6] Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., Heffernan, C. (2014). Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology*.

[7] Ocumpaugh, J., Baker, R.S.J.d., Rodrigo, M.M.T. (2012) *Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. Training Manual*. Technical Report. New York, NY: EdLab.

[8] Pardos, Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M. (2013) Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. *Proc. of the 3rd International Conference on Learning Analytics and Knowledge*, 117-124.

[9] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek. M.A. and Rasmussen, K.P. (2005). The Assistment project: Blending assessment and assisting. In *Proc. AIED 2005*, 555-562.

[10] San Pedro, M.O.Z., Baker, R.S.J.d., Bowers, A.J., Heffernan, N.T. (2013) Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. *Proceedings of the 6th International Conference on Educational Data Mining*, 177-184.

Data Mining of Undergraduate Course Evaluations

Sohail Javaad Syed, Yuheng Helen Jiang, Lukasz Golab
University of Waterloo, Canada
{sjavaad,y29jiang,lgolab}@uwaterloo.ca

1. INTRODUCTION

In this paper, we take a new look at an old problem of analyzing course evaluation data. We present an information-theoretic study to characterize courses whose ratings have high *entropy*, i.e., those which some classmates rate highly and some poorly. Our data set comes from the Engineering faculty of a large Canadian university, and, to the best of our knowledge, is an order of magnitude larger that those analyzed in previous work (see, e.g., [1, 2, 3]). After removing evaluations with fewer than 15 responses, we have 257,612 student evaluations of 5,740 undergraduate courses taught by 2,112 distinct instructors from 2003 till 2012.

Table 1 lists the 17 questions on our evaluation forms; we will refer to them by their abbreviations (e.g., Q1). Q1 through Q9 refer to teaching attributes and Q11 through Q16 refer to course attributes. Q10 and Q17 are the overall appraisals. Each question has five possible answers from A (best) to E (worst), where an A is assigned 100, B is 75, C is 50, D is 25 and E is zero. For each question, we have the frequencies of each possible answer and an average. We also have the course level, semester, and an anonymized instructor ID. Additionally, we obtained the following attributes by scraping online course calendars: class size, course type (compulsory or elective), time of lecture (we define morning classes as those which start before 10:00, day classes as those which start between 10:00 and 17:00, and evening classes as those which start after 17:00), and the number of lectures per week (one three-hour lecture, two 90-minute lectures or three one-hour lectures). Finally, we derived the following attributes for each course offering: teaching experience of the instructor (total number of times he or she taught in the past), attendance (the number of evaluations received divided by course enrolment–i.e., we assume that attendance on the evaluation day is a good indicator of average attendance throughout the course), and *specific* teaching experience (the number of times this instructor has taught this particular course).

2. RESULTS

For each course evaluation, we compute the entropy of each of the 17 questions as follows. Let p_A, p_B, p_C, p_D and p_E be the relative fractions of the students who chose options A, B, C, D and E, respectively. Then the entropy is

$$-p_A \log_2 p_A - p_B \log_2 p_B - p_C \log_2 p_C - p_D \log_2 p_D - p_E \log_2 p_E.$$

Higher entropy means that there is more variability in the responses among the students in a given class.

Table 1: Questions on course evaluation form

Q1	Instructor's organization and clarity
Q2	Instructor's response to questions
Q3	Instructor's oral presentation
Q4	Instructor's visual presentation
Q5	Instructor's availability and approachability outside of class
Q6	Instructor's level of explanation
Q7	Instructor's encouragement to think independently
Q8	Instructor's attitude towards teaching
Q9	Professor-class relationship
Q10	**Overall appraisal of teaching quality**
Q11	Difficulty of concepts covered
Q12	Workload required to complete this course
Q13	Usefulness of textbooks
Q14	Contribution of assignments to understanding of concepts
Q15	How well tests reflect the course material
Q16	Value of tutorials
Q17	**Overall appraisal of the course**

Table 2: Average entropy of each question

QID	Avg	QID	Avg	QID	Avg	QID	Avg
Q1	1.39	Q2	1.49	Q3	1.18	Q4	1.5
Q5	1.43	Q6	1.29	Q7	1.63	Q8	1.25
Q9	1.3	**Q10**	**1.47**	Q11	1.57	Q12	1.5
Q13	1.95	Q14	1.72	Q15	1.66	Q16	1.92
Q17	**1.63**						

We start by calculating the average entropy of each question, shown in Table 2. According to the t-test, Q17 has a statistically significantly higher average entropy than Q10, meaning that *classmates tend to agree more on teaching quality than overall course quality*. Of the teaching-related questions, quality of oral presentation (Q3) has the lowest entropy, which makes sense: good or bad speakers are uniformly perceived as such. Encouragement to think independently (Q7) has the highest entropy, which also makes sense since different things may make different students think. Of the course-related questions, usefulness of textbooks (Q13) and usefulness of tutorials (Q16) have the highest entropy. This is likely due to the different learning styles of different students: some learn on their own and/or from lectures, while others need a good textbook or effective tutorials. Workload (Q12) has the lowest entropy: a heavy course is perceived as heavy by the majority of students.

Table 3: Regression results

	Q10 RMSE	Q17 RMSE
Related survey attributes	0.15	0.24
All survey attributes	0.15	0.19
All survey attributes + other attributes	0.15	0.19

2.1 Predicting the Entropy of Q10 and Q17

We now turn our attention to predicting the entropy of Q10 and Q17 using linear regression. We compute the Root Mean Square Error (RMSE) of three models: First, we predict the entropy of Q10 and Q17 using only the entropy of the teaching or course-related survey attributes, respectively ("Related survey attributes"). Next, we use the entropy of all survey attributes ("All survey attributes"), followed by adding the values of other attributes we collected such as class size, instructor experience, etc. Results are shown in Table 3.

The entropy of teaching quality ratings (Q10) is explained by the entropy of the teaching-related survey questions (Q1-Q9); adding other attributes to the model does not improve the RMSE. The entropy of response to questions (Q2) and organization and clarity (Q1) have the largest regression coefficients of 0.28 and 0.27, respectively. Thus, classmates disagree on the overall teaching quality largely because they disagree on the organization and clarity of the instructor or his or her effectiveness in responding to questions.

The entropy of the overall course appraisal (Q17) can be explained by the entropy of all the survey questions, both teaching-related and course-related (using only the course-related questions has a higher RMSE, showing that teaching quality significantly influences the overall course appraisal). The entropy of usefulness of assignments (Q14) has the largest regression coefficient of 0.35, whereas the entropy of usefulness of tutorials (Q16) has the smallest coefficient of 0.01. This suggests that if classmates disagree on the overall course appraisal, they do so because some enjoy working on the assignments but others do not. On the other hand, disagreement in the rating of tutorials does not lead to disagreement in the overall rating of the course. One possible explanation is that students who do not find tutorials useful may choose not attend them, but if they like other aspects of the course, they will still rate it highly.

As for the other attributes, class size is positively correlated with the entropy of Q10, and teaching experience is slightly negatively correlated with the entropy of Q10 and Q17. Interestingly, optional courses have higher entropy of teaching quality, but lower entropy of course quality. We hypothesize that students who take an optional course are interested in the material and may rate the course uniformly well regardless of how it turns out; at the same time, some of these students may rate the instructor more highly than they normally would have, just because they liked the topic of the course, while others may rate the instructor normally. In terms of the time of lecture, evening classes have higher entropy of their appraisals. We hypothesize that some students who attend evening classes may sit in the back and do their homework instead of paying attention, and may give lower ratings; however, students who make an effort to wake up early and attend morning classes tend to pay attention and provide more consistent feedback. Finally, in terms of the course level, the entropy of the overall course appraisal is lower in first year, and then it increases significantly in the second and third years, and drops in the fourth year. The increase from first year might be because as students take more courses, they develop a better idea of what they like and do not like in a course, and as a result they express stronger opinions. The fourth-year drop is likely due to the fact that many fourth-year courses are optional, which we found to have lower course appraisal entropy.

2.2 Detailed Analysis

Our entropy analysis does not fully capture the polarity of opinions expressed by different students in the same class. For example, a course appraisal with 50 percent A's and 50 percent B's (and no other ratings) has the same entropy as an appraisal with 50 percent A's and 50 percent E's (and no other ratings). Clearly, the latter is more "controversial" as some students love it and others hate it. Motivated by this observation, we now further investigate how the responses to Q10 and Q17 are distributed over the five possible options. In general, we found that highly-rated courses have low entropy (mostly A's and perhaps a few B's) but poorly-rated courses have high entropy, meaning that they may have a non-zero number of all five possible responses. This suggests that good courses and instructors are rated highly by the majority of students, but mediocre ones may be rated highly or poorly, depending on the student.

We informally define a teaching or course appraisal (Q10 or Q17) with *no gaps* as one that has at least one of every possible option (A through E). Intuitively, courses with no gaps elicit the most variable opinions, ranging from best (A) to worst (E). Upon further investigation, we found that many courses rated between 50 and 60 contain no gaps, meaning that the average appraisal is C, but there is also at least one A, B, D and E. More surprisingly, even some courses rated as poorly as 20 have no gaps (some students liked them), as do some courses rated as highly as 80 (some students hated them)! One possible explanation for the former is that some students in bad courses may not take the evaluations seriously and they will simply choose the first answer for every question—which happens to be A—so they can complete the survey as soon as possible and leave. If true, this means that the real average appraisal of such courses is even lower than reported. For the latter, we hypothesize that even highly-rated courses may have a handful of unhappy students for various reasons.

Finally, there are no courses whose appraisals only contain A's and E's, and no other ratings in between. However, there are 13 courses whose teaching appraisals only have A's, B's and E's, and no C's and D's. The teaching quality scores of these 13 courses range from 76 to 96. Thus, these are courses that obtained mostly A and B ratings, with only a few E's. Digging deeper, we noticed that the lowest-rated questions for these courses are encouraging to think independently (Q7) and how well test reflect the course material (Q15); both of these contained many D's and E's. We hypothesize that these courses had good instructors but poorly-designed tests (or perhaps unfairly-graded tests that did not reward independent thinking); most students rated the instructor highly despite the problems with tests, but a few may have found these problems so serious that they felt the instructor deserved to be rated poorly.

3. REFERENCES

[1] B. Badur and S. Mardikyan. Analyzing teaching performance of instructors using data mining techniques. *Informatics in Education*, 10(2):245–257, 2011.

[2] K. A. Feldman. The superior college teacher from the students' view. *Research in Higher Education*, 5(3):243–288, 1976.

[3] H. W. Marsh. The use of path analysis to estimate teacher and course effects in student ratings of instructional effectiveness. *Applied Psychological Measurement*, 6(1):47–59, 1982.

Data Sharing: Low-Cost Sensors for Affect and Cognition

Keith Brawner
United States Army Research Laboratory
12423 Research Parkway
Orlando, Florida, 32826
Keith.W.Brawner@us.army.mil

ABSTRACT

The Educational Data Mining (EDM) community has experienced many benefits from the open sharing of data. Efforts such as the Pittsburgh Science of Learning Center Datashop have helped in the development of learning data storage and standards in the educational community. In other fields, standards of comparison have been created through publication, sharing, and competition on identical datasets. This ability to share, compare, and grow as a field has proven to be a success. This paper presents a new and unique dataset, and shares it with the EDM community. Initial offline analysis results and secondary online analysis results are presented as benchmarks for comparison by future researchers.

Keywords

Data mining, machine learning, data sharing, affect, cognition

1. INTRODUCTION

The Army Research Laboratory (ARL) Learning in Intelligent Tutoring Environments (LITE) Lab has an interest in Intelligent Tutoring Systems (ITS) research, and has developed the Generalized Intelligent Framework for Tutoring (GIFT) [10] as an architectural output for research. GIFT is composed of several interoperable modules for the communication of sensor, learner, instructional, and performance information, with projects involved in each area. As part of a project involving sensors and learner data, an interesting and unique dataset was collected.

The purpose of this work is to share this collected data for the purpose of ITS development with the research community at large. Among the goals of the GIFT project is to be able to rapidly transition research into the community. Transition tools, authoring tools, and multiple programming language plugins have been constructed for this purpose, are curated to ensure overall stability and use, and are freely and publicly distributed [2]. The purpose of the research described as part of this data-sharing paper was to determine the effectiveness of low-cost sensors, and to test alternative modeling techniques. It is clear that this dataset can answer additional research questions of interest to the EDM community, and will be shared publicly at http://litelab.arl.army.mil/public.

2. HARDWARE

In total, measurements were collected via two Electroencephalography (EEG) systems (from Neurosky and Advanced Brain Monitoring (ABM)), a custom-made eye tracker, a Zephyr heart rate monitor, embedded Phidget pressure sensors within the chair, a Venier sonar sensor for distance from the computer, and emotional self-report measure. The self-report measure of EmoPro and the ABM headset have previously been validated to produce accurate measures of affective and cognitive states, respectively [5; 7]. A summary of the measures which these sensors produce is included in Table 1. Larger discussion on the relevance of each of these states to learning outcomes and validation of the baseline measurements is available in prior work [3; 4; 8].

Table 1. Summary of sensors used and states measured.

Sensor	Affective State	Cognitive State
*ABM EEG**		Attention, Engagement, Distraction, Drowsiness, Workload
Neurosky EEG		
Eye-tracker		Attention, Drowsiness, Workload
*EmoPro**	Anger, Anxiety, Arousal, Boredom, Fear, Stress	Attention
Heart Rate Monitor		
Chair Pressure Sensor (posture)	Arousal, Boredom, Frustration	Engagement, Flow
Motion Detector (posture)		
** Indicates Ground Truth Measurement*		

3. INITIAL ANALYSIS

The Logistic Model Tree (LMT) method of analysis [9] was selected for classifier construction on this data from among a series of methods considered [8]. Ten-fold cross validation at the class level was used in an effort to avoid overfitting. The created trees were found to have a single node, rendering this method similar to logistic regression. The measure of Area Under the Curve (AUC) of the Receiver Operator Characteristic (ROC) [6] is used to evaluate overall model quality. In general, the AUC ROC method produces a value in the range [0,1], with 1.0 representing perfect classification accuracy and 0.5 representing baseline levels. The overall finding is that there is significant room for improvement of generalized model quality, but that data trends are available to do so. These findings are summarized with Table 2 and Table 3.

Table 2. Summary of which sensor data was used to create Initial Emotional Models

Low-Cost Sensor	EmoPro Measurements		
	Anger	Anxiety/Fear	Boredom
HR			X
Eye Track			
EEG		X	X
Chair		X	
Distance		X	X
AUC ROC	**N/A**	**0.83**	**0.79**

Table 3. Summary of which sensor data was used to create Initial Cognitive Models

Low-Cost Sensor	ABM Measurements		
	Engagement	Distraction	Workload
HR	X	X	
Eye Track			
EEG			
Chair	X	X	X
Distance	X		X
AUC ROC	**0.80**	**0.81**	**0.82**

Later projects investigated a realtime signal approach to data processing for the classification of emotional states in realtime. There is some evidence that adaptable approaches among cognitive state data are able to model more accurately, but there remain few attempts to model states in this way [1]. Additionally, there is evidence that models created from bodily sensor data may fail generalization tests for reasons such as electrode drift, changes in default impedance, and other non-linear behavioral factors [1]. The core idea of this approach is that highly adaptable and individualized approaches to modeling would be better able to model emerging states at the student level. This was found to be true for affective measurements, but not for cognitive measurements (without further feature detection).

The total of these efforts is the development of realtime algorithmic approaches which are able to classify with very little labeling information. These approaches can be compared side-by-side to the binary classification, regression-based, logistic model trees created in the earlier study. Using methods for individualized realtime model construction on multiple individuals provides evidence to how well the model is likely to transfer to a new population, while having a comparison benchmark assures that it is possible to create a model at all. Attempts to model these cognitive states have thus far met with failure, while affective ones have been met with success [3]. There is interesting research in the improvement of the cognitive models, but this research line has been abandoned in exchange for other projects.

4. CONCLUSION

The dataset in this paper has been collected at expense to the Army, but is useful to a wider public. An initial project analyzed this dataset in order to determine if low cost sensors are able to mimic the performance of high cost sensors when supplemented with classification improvements. The finding was that they were able to, but that more work was needed in order to mimic the performance of the high cost sensors in a generalized fashion with the data available.

A secondary look at this dataset investigated a different research question. This study sought to examine whether highly individualized (not generalized) affective/cognitive models could be created with the same data available to previous classifiers. The answer to this question was that it could be done for affective models, but not be done with the raw cognitive data alone. Further work would need to be done to develop filters, feature extraction, and other, differing, methods of processing for these models.

Future efforts in this line of research will likely have to abandon the limitation on the initial streams of data through the development of feature detectors and other means of data processing. Future datasets for this line of research should look to include a checklist of features (Table 4) which would render it relevant to the learning problem area.

Table 4. Checklist of features for Low Cost Sensor dataset (recommended for other studies)

Does the dataset have…	?
Relevant states to learning	
Ability to be transferred, without modification, to another domain of instruction	
Relevant population	
Relevant cost for classroom inclusion	
Labeled data	
Initial benchmarks for research comparison	

5. REFERENCES

[1] ALZOUBI, O., CALVO, R., and STEVENS, R., 2009. Classification of EEG for Affect Recognition: An Adaptive Approach. *AI 2009: Advances in Artificial Intelligence*, 52-61.

[2] ARL, 2012. Generalized Intelligent Framework for Tutoring Release Page Army Research Laboratories, http://www.gifttutoring.org.

[3] BRAWNER, K.W., 2013. Modeling Learner Mood In Realtime Through Biosensors For Intelligent Tutoring Improvements. In *Department of Electrical Engineering and Computer Science* University of Central Florida, 500.

[4] CARROLL, M., KOKINI, C., CHAMPNEY, R., SOTTILARE, R., and GOLDBERG, B., 2011. Modeling Trainee Affective and Cognitive State Using Low Cost Sensors. In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, Orlando, FL.

[5] CHAMPNEY, R.K. and STANNEY, K.M., 2007. Using emotions in usability SAGE Publications, 1044-1049.

[6] HANLEY, J.A. and MCNEIL, B.J., 1983. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology 148*, 3, 839-843.

[7] JOHNSON, R.R., POPOVIC, D.P., OLMSTEAD, R.E., STIKIC, M., LEVENDOWSKI, D.J., and BERKA, C., 2011. Drowsiness/alertness algorithm development and validation using synchronized EEG and cognitive performance to individualize a generalized model. *Biological Psychology*.

[8] KOKINI, C., CARROLL, M., RAMIREZ-PADRON, R., HALE, K., SOTTILARE, R., and GOLDBERG, B., 2012. Quantification of trainee affective and cognitive state in real-time. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)* NTSA.

[9] LANDWEHR, N., HALL, M., and FRANK, E., 2005. Logistic model trees. *Machine Learning 59*, 1-2, 161-205.

[10] SOTTILARE, R.A., BRAWNER, K.W., GOLDBERG, B.S., and HOLDEN, H.K., 2012. The Generalized Intelligent Framework for Tutoring (GIFT).

Diagnosing Algebra Understanding via Inverse Planning

Anna N. Rafferty
Computer Science Division
University of California, Berkeley, CA 94720 USA
rafferty@cs.berkeley.edu

Thomas L. Griffiths
Department of Psychology
University of California, Berkeley, CA 94720 USA
tom_griffiths@berkeley.edu

ABSTRACT

Students' solution processes can offer significant insight into their misunderstandings. However, freeform solutions can be difficult to interpret, leading many educational technologies to examine only students' final answers or to structure problem solving to make it more interpretable. We develop a new approach using Bayesian inverse planning for diagnosing algebra skills that interprets students' step-by-step problem solving, placing no restrictions on how students transform equations to reach solutions. We formalize understanding as several distinct skills, allowing us to identify the causes of errors; for instance, arithmetic skills are separated from systematic misapplications of algebra rules. In simulation, the algorithm recovers the true parameters of simulated learners relatively accurately. Using human data, we show that the algorithm can interpret over 98% of people's actions and its inferences about arithmetic skills are consistent with an assessment of arithmetic ability in isolation. Our work demonstrates that Bayesian inverse planning can successfully scale to the space of algebra and provides new technical solutions that may be relevant in other complex educational domains.

1. INTRODUCTION

The way that students approach and solve problems can provide significant insights into their understanding. Classroom teachers encourage students to "show [their] work," allowing them to gain insight into the students' difficulties. In principle, computers should be able to make such inferences automatically, drawing fine-grained inferences about students' skills based on their choices about problem-solving and the types of errors they make. However, many automated tutoring systems cannot interpret students' worked solutions. We focus on the case of teaching and remediating algebraic equation solving. Existing computer-based systems for helping students learn algebra typically take one of two approaches to assessing and modeling students' algebra understanding. They may structure the problems such that students enter their work in discrete parts, with each part corresponding to a different algebra skill; students generally must enter the current part correctly prior to moving on. Alternatively, these systems may use only final answers to infer understanding, with many systems solely checking whether an answer is correct.

While it may be pedagogically useful in some cases to structure students' behavior or interrupt their work to point out errors, it should not be necessary to do these things to infer students' understanding from their worked solutions. We develop a Bayesian inverse planning model that can diagnose a student's understanding from observing how she solves linear equations. This approach allows data from multiple problems to be incorporated into the diagnosis, and naturally accounts for inconsistent behavior across problems. The model extends existing work on algebra understanding by using freeform problem solving behavior to diagnose what a student understands and in what ways she misunderstands without requiring that individual steps be correct before the student continues.

2. MODELING ALGEBRA SKILLS USING BAYESIAN INVERSE PLANNING

Bayesian inverse planning uses Markov decision processes (MDPs) to model how people plan their actions in order to achieve their goals, and infers a diagnosis of their understanding as a distribution over possible *hypotheses* [5]. For linear equation solving, the hypotheses represent the possible misunderstandings that people might have about solving algebraic equations and carrying out mathematical operations. By detecting patterns in a person's equation transformations, the algorithm can determine which hypothesis is most likely to represent the person's knowledge. To develop a Bayesian inverse planning algorithm for linear equation solving, we must define how to model equation solving as an MDP and specify the space of possible hypotheses.

MDPs provide a decision-theoretic method for modeling sequential action planning (for an overview, see [7]). MDPs are defined by the set of states that characterize the environment in which the agent is acting and the actions the agent may take. The transition model in an MDP defines the conditional probabilities distributions $p(s'|s, a)$ that the next state will be s' given that the current state is s and the chosen action is a. The reward model then encodes the goals and incentive structure for the agent; in this case, the goal of solving an equation, with fewer actions favored over more actions. We represent the state in linear equation solving as the list of terms on each side of the equation, ignoring the ordering of these terms. The actions are the possible transformations that a student might apply to an equation. We include six types of actions: moving a term, dividing by a coefficient, multiplying by a constant, combining terms, distributing over a parenthesized term, and terminating solving. These actions are sufficient to represent typical problem solving behavior, with the final action occurring each time a student completes a problem or gives up.

To use Bayesian inverse planning to diagnose students' understanding, we must define the hypothesis space of possible knowledge states. In this case, the knowledge states correspond to possible transition models: how does a student believe the state should be transformed when she chooses a particular action? We represent each knowledge state as a vector θ of six parameter values. Four of these values relate to error tendencies in applying specific actions, based on mal-rules discovered in prior work [4, 6]. For example, one parameter represents the probability a student will make a *sign error* in which she moves a term from one side of the equation to the other without changing the sign: $2x+3=6$ becomes $2x=6+3$. The other two parameters relate to equation solving behaviors not tied to a specific action. The *arithmetic error* parameter is the probability that a student will make an arithmetic error in each operation in a transformation. Separating out this error term differentiates students who get problems wrong due to misunderstandings about the rules of algebra from students who have difficulties with arithmetic. The final parameter relates to the efficiency with which the student solves equations. Following prior work modeling human action planning [1, 5], we assume students choose their actions noisily optimally: $p(a|s) \propto \exp(\beta Q(s,a))$, where $Q(s,a)$ represents the long-term expected value of taking action a in state s. β controls the noisiness of the policy, with increasingly large β corresponding to more optimal action selection. We infer the value of β that best models an individual student's action planning, allowing us to detect how well a student is choosing her actions. Very low inferred values of β might also indicate that the student's data is not well fit by our model.

We represent the diagnosis of a student's algebra understanding as the posterior distribution $p(\theta|d_1,\ldots,d_N)$ over the possible parameters given N observed problem solutions. This distribution can be calculated using Bayes' rule, making use of the fact that each θ corresponds to a particular MDP. Because θ contains continuous parameters, we approximate the posterior via Markov chain Monte Carlo sampling [3]. For each sample, we must calculate a Q-function of long-term expected values given the current θ. Since the state and action spaces are infinite, this function can also only be approximated. We use discretization to aggregate both the state and action spaces, a common strategy in large or continuous MDPs (e.g., [2, 7]).

3. DIAGNOSING UNDERSTANDING

To evaluate the effectiveness of Bayesian inverse planning for diagnosing algebra understanding, we first tested the framework in simulation. Simulations allowed us to assess whether there was sufficient information in the problem traces (i.e., the series of equations representing the transformations from the initial equation to the final solution) to recover the true parameters of a learner. We found that recovery of these values was relatively accurate, with the median difference between actual and inferred values for five of the six parameters less than 0.1. The planning parameter had a median difference of less than 0.25; the larger difference is likely due to its increased range. This suggests that while the diagnosis computed by Bayesian inverse planning is approximate, it still provides accurate information about the learner and might be used to guide remediation.

We then evaluated our model's performance on human data by recruiting participants on Amazon Mechanical Turk to complete an online worksheet and solve twenty problems on the Berkeley Algebra Tutor website, which we designed to collect step-by-step equation solving data. Over 98% of equation transformations could be interpreted by our model, and through manual annotation of a subset of the equations, we found that the model's interpretation of the transformations were generally consistent with human observers' interpretations. By comparing the model's inferred parameter values for individual participants with these participant's worksheet performance, we found that the inferred arithmetic error parameter was correlated with scores on the arithmetic portion of the worksheet. These results demonstrated that the model can interpret real equation-solving data and compute diagnoses of individual algebra skills.

4. CONCLUSION

Developing a Bayesian inverse planning algorithm for algebra offers promise for both extending the scalability of the inverse planning framework and helping to remediate learners' algebra skills. Our work demonstrates that inverse planning can be applied to a complex educational domain, and provides solutions for common technical problems that may arise, such as infinite state and action spaces. Our simulation and experimental results suggest that the algorithm can effectively recover the parameters of simulated learners and interpret people's equation solving. We plan to further evaluate the model using data from current algebra learners and to test the effectiveness of personalized guidance for learners based on our diagnosis of their understanding.

Acknowledgements. This work was funded by an NaEd/Spencer Dissertation Fellowship to ANR and ONR MURI grant number N00014-13-1-0341 and NSF grant number IIS-0845410 to TLG.

5. REFERENCES

[1] C. L. Baker, R. R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[2] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst. *Reinforcement learning and dynamic programming using function approximators*. CRC Press, Boca Raton, FL, 2010.

[3] W. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov chain Monte Carlo in practice*. Chapman and Hall, Suffolk, UK, 1996.

[4] S. Payne and H. Squibb. Algebra mal-rules and cognitive accounts of error. *Cognitive Science*, 14(3):445–481, 1990.

[5] A. N. Rafferty, M. M. LaMar, and T. L. Griffiths. Inferring learners' knowledge from their actions. *Cognitive Science*, in press.

[6] D. Sleeman. An attempt to understand students' understanding of basic algebra. *Cognitive Science*, 8(4):387–412, 1984.

[7] R. S. Sutton and A. G. Barto. *Reinforcement learning*. MIT Press, 1998.

Discovering and Describing Types of Mathematical Errors

Thomas S. McTavish*
Center for Digital Data, Analytics & Adaptive Learning
Pearson
tom.mctavish@pearson.com

Johann Ari Larusson
Center for Digital Data, Analytics & Adaptive Learning
Pearson
johann.larusson@pearson.com

ABSTRACT

Given a large number of incorrect responses to mathematical exercises, we ask, "What errors might the learner have made to arrive at their answer?" Even though our data does not contain intermediate steps, we find that we are able to infer well over 50% and sometimes over 90% of the types of errors learners make on an exercise when they only supply final answers. Our approach capitalizes on the sheer volume of data to highlight patterns and the fact that these exercises come from item banks of mathematical templates. Since items generated from mathematical templates deliver different parameters to different learners (e.g., one learner might see $y = 2x + 3$ while another learner might see $y = 3x + 5$), misconceptions and mechanical errors are more easily recognized. We enumerated different errors for simpler-stated problems and utilized other forms of signal analysis in other cases to uncover error types. Our results show that there are many types of errors even for seemingly simple problems, and we can quantify their relative degrees of prevalence. We can also determine bias in the templates that make a problem easier or more difficult depending on which parameters are used. Since error categories correlate with knowledge components, our work highlights the relative degree of knowledge components embedded within a problem and exposes some knowledge components that may otherwise remain unconsidered.

Keywords

Error Analytics, Misconceptions, Knowledge Components, Template Bias, Automatic Item Generation

1. INTRODUCTION

The etymology of the word *demonstrate* comes from the latin *dē* ("concerning") + *mōnstrō* ("I show") [25]. The noun form of *mōnstrō* has become *monster* in English, the word for something that warns or instructs. Medicine and biology have a long heritage of learning from abnormal patients and aberrant individuals, these so-called "monsters". For example, in the 19th century, after treating patients who exhibited severe speech and language deficits, Drs. Broca and Wernicke proposed areas of the brain giving rise to speech and language after their postmortem analysis revealed brain injuries at specific sites [7, 24]. In similar spirit, MRI scanning of stroke victims today continues to reveal the functional map of the brain [4, 18]. Likewise, the field of genetics is largely built around animal models such as the mouse and fruit fly where standard practice is to "knock in" or "knock out" genes and observe the phenotypes of the mutants [5, 21].

Indeed, errors help frame "normal". In the context of learning, the types of errors that are revealed in a task demonstrate areas of confusion and the hurdles that need to be overcome to attain mastery. Errors therefore have a strong correspondence with the knowledge components (KCs) – the skills, concepts, and rules of a problem [12, 16, 17, 22]. It has been found that those who have achieved mastery categorize problems differently than those who are novices, as their categorization is more shallow [3]. Prior work has shown that more often than not, student errors in simplistic fraction multiplication word problems concern the vocabulary used rather than the actual math itself [8, 13]. Furthermore, evidence exists demonstrating not only a causal relationship between students' prerequisite knowledge, or lack thereof, and errors in problem solving, but also that gaps in prior knowledge negatively impacts students' accrued learning [23]. Also, strategic errors in instruction, have been shown to be at fault and contribute to particular limitations in prior knowledge [2]. Such research highlights that errors can help sculpt and define KCs. We will argue that in many cases, they may be two sides of the same coin.

In the work presented in this paper, we evaluated incorrect responses to mathematical exercises to determine and quantify specific types of errors learners made. Even though our data contained only final answers, we were still able to label 60-90% of the errors without intermediate step data. Our approach exploited the relatively large number of samples of each problem (hundreds or low thousands) and took advantage that the exercises came from templates, each instance of the template having different parameters (e.g., "$4 + 7$" or "$3 + 5$"). Comparing across template instances permitted us to see repeated patterns. Sometimes setting incorrect responses against the backdrop of correct solutions provided clearer interpretations of the incorrect response to more eas-

*Corresponding author.

ily label it. We found we could ascribe several types of errors to even seemingly straightforward exercises, demonstrating that several KCs may go into an exercise. Additionally, we were able to determine bias in the template that made the problem easier or more difficult depending on the parameters given. As such, bias illuminated the challenges delivered to some learners that were not given to others. While such bias has strong implications for assessments, it also permits us to dissect the exercise further and label a KC associated with some values of the template's parameters that is otherwise absent in other instances.

2. METHODS

2.1 Data

Our data consisted of student responses to online math problems from a college level developmental math book. Students, largely from the U.S., were enrolled in courses spanning Fall semester 2012 through 2013 that used the Pearson MathXL® homework system. All responses were from quizzes or tests. Students may have seen an exercise in a prior homework, or, somewhat rarely, may have taken a quiz or test multiple times to have multiple exposures to the exercise, but we did not factor this into our analysis. We used the final answer to the problem, which was a free text field. Therefore, learners could enter strings that could remain string literals, or be parsed into numbers. Alternatively, students could use an equation editor to enter mathematical expressions into the field. Responses were either labeled "correct" or "incorrect".

The system employs automatic item generation from math templates, randomly creating instances of the exercise, often with certain constraints in the hopes of keeping the problem within the same domain and similar range of difficulty. By the combinatoric nature of the parameters used in some templates of the system, some exercises have nearly an infinite number of possible instances. For this study, we concentrated on some templates that had few instances (3 - 24) for all but the "GPA" example, which had 1022 instances.

2.2 Our approach

We collected student responses to each instance of each exercise. For the cases with few instances, we looked at the distribution of incorrect responses. Our null hypothesis is that incorrect responses are random guesses. Since the response field is free text, this means that the distribution of possible answers will be very large under the null hypothesis. It was straightforward, then, to find those cases where several students converged to enter the same incorrect response. What was not so apparent, and remains the bottleneck of our approach, was determining *how* students arrived at their particular response. For this, we looked across the instances of the template, comparing the peak responses to determine consistent patterns. This is perhaps best illustrated with an example as shown in Table 1 that shows a sampling of responses where four or more students gave the same response to the question "Find the reciprocal of the number x." Looking across the instances where x is 2, 3, 4, or 5, there are repeated patterns of users giving the negative of the number, or framing the response as $x/1$ instead of the correct response of $1/x$. The table also shows many students simply echoing the number given. Because these patterns repeat themselves across the different instances, we can more confidently define the type of error by seeing how it generalizes. After determining the type of error, we wrote mathematical expressions to match the specific error for the given input parameters of the instance. We therefore tagged those responses that had one or more errors attributed to them. (Because an incorrect response might match more than one error formula, the "Total inferred" row at the bottom of many of the tables we provide in our results is not a subtotal of each error category. Each tagged incorrect response was only counted once.) We then filtered out these cases and iteratively considered the remaining responses in attempts to further ascribe possible types of errors.

Our "GPA" example had a different output from the others. In this case, only a handful of students saw the same instance. We therefore contrasted the distributions of correct and incorrect responses through visualizations as described in Section 3.6.

We determined bias in a template in the following ways: Firstly, we considered the fraction correct from students that saw a particular instance as compared with the rest of the instances in a binomial test. This permitted us to see if a set of specific instances were easier or harder. We then looked at all instances that had a variable set to a specific value. Taking each variable across all of its values and performing the same binomial test allowed us to determine if and when a variable showed bias. We carried this one step further and performed the same binomial test on pairs of instance variables as illustrated in Figure 1.

3. RESULTS

We applied our method across several different templates as highlighted in each subsection.

3.1 Find a quotient example

Students were presented with the exercise "What is the quotient of x and 5?" where x was a uniformly random multiple of 5 in the range $[1000, 1250]$. We evaluated 2467 responses of which 379 were incorrect. With 51 possible instances with $x \in \{1000, 1005, \ldots, 1245, 1250\}$, there were only 7 incorrect responses per instance on average. Nevertheless, we noted that 31% of the errors followed the form $x \times 5$ and 12% of errors matched $x + 5$, indicating a misconception or misunderstanding surrounding "quotient". Interestingly, while many errors either multiplied or added, less than 1% of students gave a response that matched $x - 5$, implying that these students realized "quotient" did not involve subtraction. About 5% of error responses simply echoed the numerator, x, or the denominator, "5". With the remaining responses, we could see that if the correct answer contained a "0" digit, that many student responses omitted it. For example, if users were asked to find the quotient of 1015 and 5, which is 203, they might give "23". Such a response indicates a mechanical error or misconception surrounding place values and accounted for 10% of all errors. In fact, while the mean probability correct was 85% for this problem, when contrasting problems that had a "0" in their correct answer vs. those that did not, the probability of success was 81% ($p = 0.008$) and 87% ($p = 0.025$), respectively, indicating that the problem added another knowledge component when asking students to deliberately consider the place value. Collectively,

Discovering Prerequisite Relationships among Knowledge Components

Richard Scheines, Elizabeth Silver
Department of Philosophy
Carnegie Mellon University
{scheines,silver}@cmu.edu

Ilya Goldin
Center for Digital Data, Analytics, and Adaptive Learning
Pearson Education
ilya.goldin@pearson.com

ABSTRACT

Knowing the prerequisite structure among the knowledge components in a domain is crucial for instruction and assessment. Treating Knowledge Components as latent variables, we investigate how data on the items that test these KCs can be used to discover the prerequisite structure among the KCs. By modeling the pre-requisite relations as a causal graph, we can then search for the causal structure among the latents via an extension of an algorithm introduced by Spirtes, Glymour, and Scheines in 2000. We validate the algorithm using simulated data.

Keywords

Domain models, knowledge components, q-matrix, prerequisites, causal discovery

1. INTRODUCTION

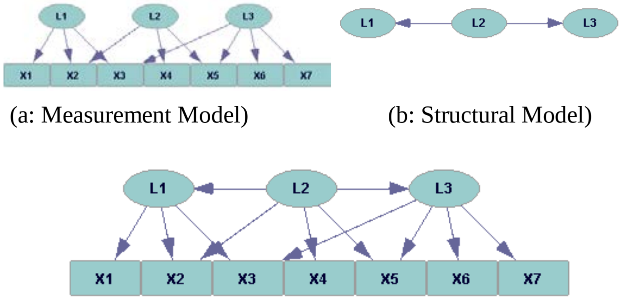
In general, we need to determine the prerequisite structure of a domain. [3,4] Instead of relying on expert knowledge, which is subject to an “expert blind spot,” in this paper we explore using causal model search to discover prerequisite structures from data.

As prerequisite relations are a form of causal relations, and as skills can be modeled as “latent” (unmeasured) variables, our approach is a generalization of causal structure discovery algorithms involving latent variables (Build Pure Clusters (BPC) [6] and MIMbuild [2, page 319]). In these algorithms, however, items are assumed to be “pure,” that is, direct measures of only a single latent skill. In education this assumption is unreasonable, so we need to generalize the algorithms to handle models with impure measures. Further, BPC was written for continuous items. It would need to be extended to work on binary data before it could be applied to “correct/incorrect” test items.

We begin with a simplifying assumption that we hope to eventually relax: that the Q-matrix (the matrix that specifies which item measures which skills) is known. We know of no current method for learning the prerequisite structure among skills in cases where there are very few pure items; so although the method we propose here is limited to cases where the Q-matrix is known, our method solves a novel problem. There are existing techniques for discovering and refining a Q-matrix, so there will be many cases where the Q-matrix is known or can be estimated to some approximation.

2. PREREQUISITE DISCOVERY

We model skills as continuous variables that represent the degree to which a student has mastered or has knowledge of a particular skill. We treat items as continuous variables that reflect the degree to which a student completed a task correctly. In practice, the measure of task completion is often a binary variable with values = correct/incorrect. A binary item can, however, be considered as a projection of a continuous item, and correlations among idealized continuous items can be estimated by computing the tetrachoric correlation matrix among the measured binary items.



(a: Measurement Model) (b: Structural Model)

(c: Full Structural Equation Model)

Figure 1: Structural Equation Models

The Q-matrix typically defines which items “load” on which latent skills. We can define a “measurement model” that relates latent skills to measured items (Fig 1-a). By modeling the relations among the skills as a path analytic causal model among the latent variables (Fig 1-b), called the “structural model,” we can then combine the “measurement model” and the “structural model” to form a full *linear structural equation model* [1].

By assuming that the measurement model is known, we can search for the structural model with the PC causal discovery algorithm [2], in which the inputs are the independence and conditional independence relations that hold among the latent variables. We compute or test the independence relations among the latents by constructing a distinct structural model and fitting it to the data for each particular independence test required. Our model construction method produces a provably consistent test of each conditional independence relation. [7]

3. VALIDATION ON SIMULATED DATA

To measure the method’s ability to recover prerequisite structure, we conducted a large simulation study in which we varied (i) the structural model, (ii) the purity of the measurement model, (iii) the sample size, and (iv) whether the observed data were continuous or binary. In each of these conditions, we performed 100 simulations with different parameterizations. We used three structural models representing different causal relations between the latent skills, and varying degrees of impure measurement models (complicated Q-matrices). We ran the PC algorithm using the new test for independence involving a constructed structural equation model, and produced an equivalence class for the structural model in which we assumed no additional latent confounding, called a *pattern* [5].

We then scored each graph on the following metrics:

1. *True positive adjacency rate* (# correct adjacencies in output / # adjacencies in true graph), a.k.a. recall

2. *True positive orientations* or *orientation recall* (# correctly oriented edges in output / # orientable edges in true equivalence class). Defined to be 1 if none of the edges in the true equivalence class are orientable.

4. Results

Our results show that the algorithm performs well for discovering adjacencies (Figure 2). Even in the most difficult (and most realistic) case, where the sample size is 150, the measurement model is impure, and the data are binary, we still recover 74%, 76%, and 89.5% respectively for the three generating models.

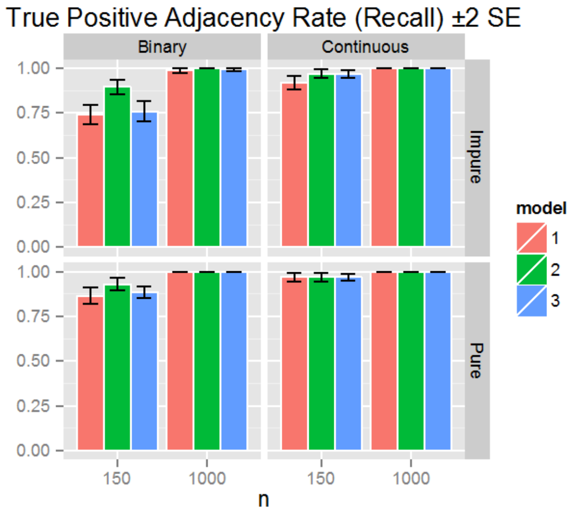


Figure 2: True positive adjacency rate

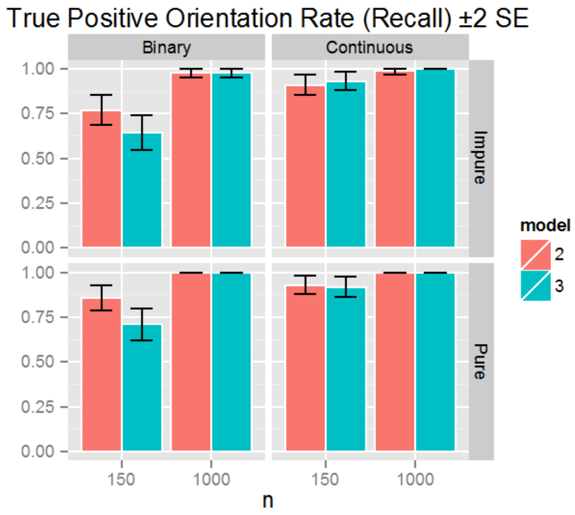


Figure 3: True positive orientation rate

The true positive orientation rate (recall) is shown in Figure 3. The worst score is 64.5% (for Model 3, with binary data, an impure measurement model and sample of 150), which is still quite good. Results for other metrics (omitted for lack of space) are also very good [7], including *false positive adjacency rate, true adjacency discovery rate, false positive orientation rate, true orientation discovery rate*, and *false negative orientation rate*.

5. CONCLUSIONS

The prerequisite graph is an important pedagogical artifact in itself, because we can use it to examine the structure of a domain, and it is furthermore a critical element of adaptive learning environments, where it can be used to create personalized and efficient learning trajectories for students. We expect that our algorithm can be used to discover fine-grained prerequisite structures to make student learning more efficient and more effective.

Unlike prior work [8], our method of prerequisite discovery only requires a single assessment from a point in time, and it applies to an assessment of any scope, regardless of whether it covers multiple problem-solving strategies on a skill, or multiple skills on a single learning objective, or multiple objectives in a syllabus, or multiple courses in a multi-year curricular sequence (e.g., a standardized test). Our algorithm is the only method currently available for inferring latent structure when the measurement model contains few pure items (i.e. items that load on only one latent). It performed well in our simulations, but has several important limitations including the assumption of linear relations, that the Q-matrix is known, and that the models are identified.

We intend to extend the work by expanding the range of models that can be identified, by investigating the robustness of the procedure to errors in the Q-matrix specified, and by including steps for Q-matrix discovery.

6. REFERENCES

[1] Bollen, K. (1989). *Structural Equation Models with Latent Variables*. Wiley.

[2] Spirtes, P, Glymour, G., Scheines, R. (2000). *Causation, Prediction and Search*, 2nd Edition, MIT Press.

[3] Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., & Johannesen, L. (1990). Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review*, 97(2), 201.

[4] Tatsuoka, K. K. (2012). *Cognitive Assessment: An Introduction to the Rule Space Method*. Routledge.

[5] Pearl, J. (2009). *Causality: Models and Reasoning, and Inference*, 2nd Edition. Cambridge University Press.

[6] Silva, R., Scheines, R., Glymour, C., Spirtes, P. (2006). "Learning the structure of linear latent variable models."*The Journal of Machine Learning Research* 7: 191-246.

[7] Scheines, R., Silver, E., Goldin, I. (in preparation) Discovering Prerequisite Relationships among Knowledge Components. Pearson Research Report.

[8] Vuong, A., Nixon, T., & Towle, B. (2011). A Method for Finding Prerequisites Within a Curriculum. In *Proceedings of 4th International Conference on Educational Data Mining* (pp. 211–216).

Dynamic Re-Composition of Learning Groups Using PSO-Based Algorithms

Zhilin Zheng
Department of Computer Science
Humboldt University of Berlin
Berlin, Germany

zhilin.zheng@hu-berlin.de

Niels Pinkwart
Department of Computer Science
Humboldt University of Berlin
Berlin, Germany

niels.pinkwart@hu-berlin.de

ABSTRACT

In collaborative learning contexts, the problem of automatically forming effective learning groups gets considerably complex with larger class sizes, e.g. in MOOCs. Additionally, group dynamics caused by high dropout rates currently observable on online open course platforms poses challenges to learning group formation strategies. To address these problems, this paper presents PSO-based algorithms to dynamically re-compose learning groups. In addition to static grouping criteria (such as MBTI personality types), the algorithms take into account factors of the group success rate and group satisfaction during re-composition. We carried out simulations based on randomly generated sample data. The experimental results show that the proposed approach performs better than traditional exhaustive or random methods.

Keywords

Group Formation; Collaborative Learning; Group Composition; Group Dynamics

1. INTRODUCTION

The concept of re-composing learning groups was introduced by Oakley et al. [4]. They dissolved dysfunctional teams to re-form more effective teams. However this is just one single motivation for re-forming learning groups. There is another important reason that should not be forgotten in the light of the growing popularity of massive open online courses (MOOCs): the dropout rate. According to Dung Clow's findings, the dropout rate on MOOC platforms is considerably higher than in traditional education [2]. Only 3% of the initial participants took the final exam in a bioelectricity MOOC Duke offered through Coursera [5]. In collaborative learning contexts, this high dropout rate may cause learning groups to collapse. Therefore, it is crucial to re-compose learning groups in order to enable an effective collaborative learning setting also in later parts of a course when many participants might have left.

The rest of the paper is organized as follows. The following two sections describe our research methods and the proposed approach. We then present the simulation results. Finally, the last section concludes this paper.

2. METHOD

In this paper, we assume that a class S is composed of a given number of n students, $S = \{s_1, s_2, \dots s_n\}$. Before taking a course, instructors must divide these n students into r groups, $G = \{g_1, g_2, \dots g_r\}$. Each student can only be a member of a single group. G is the initial group formation which can, for instance, be formed by diversifying MBTI personality and distributing even gender. Subsequently, these r groups of students are instructed to complete their first group tasks. When they finish, every group's work is rated, $SR = \{SR_1,\ SR_2,\ \dots, SR_r\}$. From the rating data, we then estimate the pair success rate PS_{ij} (i.e. if s_i and s_j are in group g_k, then $PS_{ij} = SR_k$). In parallel and in addition to the performance rating, the participating students are invited to state their personal satisfaction rating, SA_{ij}, with respect to their teammates. SA_{ij} stands for s_i's subjective satisfaction rating about working in one group with s_j. The satisfaction rating indicates how much one student is willing to work with each of his teammates. When the satisfaction rating is low, the student will very likely not want to stay in the same team with his counterpart. Between group tasks, we also assume some certain percentage of students dropping out from the course. Then, in the next group task, we intend to re-compose the remaining students into learning groups aiming at meeting the initial grouping criteria as well as maximizing group success rate and pair satisfaction. We then follow this strategy to re-compose learning groups task by task.

3. PSO-BASED APPROACH

In order to solve our group re-composition problem, a Discrete Particle swarm optimization (DPSO) algorithm is proposed in this study which was previously introduced to the manufacturing cell design problem [3] and the travelling salesman problem [1]. We use a list representation for a group formation: *n* students are simply permutated in a list of length *n*. The PSO starts with initial solutions which are called particles, then updates these initial solutions and searches for the optimal solution iteratively. The velocity vector v_k^{t+1} which is used to update a particle P_k for the next iteration can generally be calculated using (1).

$$v_k^{t+1} = c_1 * v_k^t + c_2(P_{k,best} - P_{k,current}) + c_3(G_{best} - P_{k,current}) \quad (1)$$

In (1), t stands for the number of the current iteration and k indicates the number of the updated particle. $P_{k,current}$, $P_{k,best}$ and G_{best} indicate the current state of P_k, the personal best prior state of P_k and the global best particle state. c_1, c_2, c_3 are learning coefficients. Representation-wise, a velocity vector v_k is a set of pairwise permutations (i, j) that will be used to update P_k^t to P_k^{t+1} as shown in (2).

$$P_k^{t+1} = P_k^t + v_k^{t+1} \quad (2)$$

In DPSO two fitness functions should necessarily be designed to evaluate the quality of each group formation at the initial stage and re-composition phases respectively, as shown in (3) and (4). Here, $D(g_i)$ is an indicator of diversity of MBTI personality and gender distribution in a learning group (the larger the better).

$$Z_{ini}(P_k) = \frac{\sum_{i=1}^{r} D(g_i)}{r}, 0 \le Z_{ini}(P_k) \le 1 \qquad (3)$$

$$Z_{re}(P_k) = \frac{\sum_{i=1}^{r} w_d \times D(g_i) + w_{sr} \times SR(g_i) + w_{sa} \times SA(g_i)}{r} \qquad (4)$$

The complete DPSO algorithm is described in Figure 1.

```
Step1: initialize a population of N particles  randomly;
Step2: do {
            evaluate the fitness of each particle by the equation (3);
            for each particle
                update the personal best and global best;
                update velocity by the equation (1);
                update the particle by the equation (2);
            end for
        }
      while (the maximal number of iteration is not reached)
Step3: output the global best;
Step4: do{
            Do Step1, Step2 and Step3 but use the equation (4) to
            evaluate the fitness of each particle instead of (3);
        }
      while (all group tasks not finished)
```

Figure 1. DPSO algorithm to re-compose groups

4. SIMULATION RESULTS

As shown in the formula (4), the group quality is calculated based on the MBTI and gender diversity, the group success rate and the group satisfaction rate. The impacts of these three factors are controlled by three weights (i.e. w_d, w_{sr}, w_{sa}). Basically, we have two ways to determine the weight factors. One way is to use fixed weights (possibly gained by experience of through systematic research and test).We simply set $w_d = 0.3$, $w_{sr} = 0.3$, $w_{sa} = 0.4$ for our tests. The other way is to define the weight factors w_d and w_{sr} adaptive to the students' co-working experience. If one group of students has worked together for many times, we can emphasize their previous group success rate and pair satisfaction and reduce consideration of their personal traits diversity. Technically, we set $w_d = 0.3 \times (1 - \alpha^{1/3})$, $w_{sr} = 0.3 \times (1 + \alpha^{1/3})$, $w_{sa} = 0.4 \times (1 + \alpha^{1/3})$. α is a co-working experience factor. We conducted an experiment to test the two methods (fixed vs adaptive weights) on a randomly generated dataset in comparison to two traditional methods, the random method and the exhaustive method.

4.1 Synthetic Data

Participating students' personal traits which exactly contain gender and MBTI personality are typically collected via online surveys. In our research, we generated this data randomly (i.e. each student was randomly assigned a gender information and an MBTI personality type). We designed 4 data sets (made up of 150, 300, 900 and 3000 students respectively) and used this dataset for 8 group re-compositions to test our algorithms. At the stages of group re-compositions, we modeled a dropout rate of 40%, 20%, 10%, 8%, 6%, 4%, 2%, 2% from the first group re-composition to the last one. Group performance and pair satisfaction were also randomized.

4.2 Performance Analysis

The proposed DPSO algorithm has been implemented in MATLAB and tested on the synthetic data illustrated in the previous subsection. The group size in the experiment was set to three. The simulation was conducted on a personal computer with an Intel(R) Core(TM) i7-4600U CPU 2.10GHz and 8GB RAM. We evaluate the DPSO algorithm's performance by computational time and quality of grouping (as measured by the fitness value). As a result, the DPSO algorithm can achieve a near-best solution to our re-composition problem in comparison to the exhaustive method, and runs considerably faster (for 3000 students, the time cost on our machine was just 11 minutes at maximum). The fixed-weights method performs closely similar as the adaptive-weights method in terms of group quality and time cost. As anticipated, the adaptive-weights method composes fewer groups with low satisfaction pairs by comparison of the fixed-weights method, as shown in Table 1 (the percentage indicates how many groups contain a pair of members with a pair satisfaction lower than 0.3).

Table 1. Low pair satisfaction percentages

	fixed-weights				**adaptive-weights**			
	150	**300**	**900**	**3000**	**150**	**300**	**900**	**3000**
Comp.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1st re-Comp.	50.7%	60.5%	62.7%	68.5%	28.0%	43.3%	51.8%	62.1%
2nd re-Comp.	57.3%	56.8%	63.3%	66.6%	31.0%	44.0%	49.3%	63.0%
3rd re-Comp.	42.0%	54.3%	56.4%	65.5%	28.0%	30.0%	47.8%	60.0%
4th re-Comp.	39.1%	48.6%	57.3%	63.8%	23.6%	30.9%	44.8%	56.3%
5th re-Comp.	25.0%	48.1%	52.1%	62.7%	10.0%	25.0%	37.9%	54.5%
6th re-Comp.	45.0%	44.2%	50.0%	65.6%	6.7%	18.3%	37.8%	50.7%
7th re-Comp.	52.0%	45.0%	51.7%	60.6%	40.0%	8.0%	37.3%	49.2%
8th re-Comp.	47.5%	37.5%	47.1%	57.9%	25.0%	12.5%	26.7%	46.5%

5. CONCLUSION AND FUTURE WORK

In this paper, we presented a new method for dynamically re-composing students into learning groups by taking into account both (static) personal characteristics, dynamic data (student group success and satisfaction) and student dropout rates. We also proposed a DPSO algorithm to dynamically re-compose collaborative learning groups based on the method. The proposed algorithm is able to search for the near-best solution to our group re-composition problem in an acceptable computational time as compared to the exhaustive method. Additionally, the adaptive-weights method is able to largely reduce the violation of low pair satisfaction. In our future research, we will test this algorithm against real data collected from large online courses.

6. ACKNOWLEDGMENTS

This research is supported by the China Scholarship Council (CSC).

7. REFERENCES

[1] CHANGSHENG, Z., JIGUI, S., YAN, W., and QINGYUN, Y., 2007. An Improved Discrete Particle Swarm Optimization Algorithm for TSP. In *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, 2007*, 35-38.

[2] CLOW, D., 2013. MOOCs and the funnel of participation. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (Leuven, Belgium2013), ACM, 2460332, 185-189.

[3] DURAN, O., RODRIGUEZ, N., and CONSALTER, L.A., 2008. A PSO-Based Clustering Algorithm for Manufacturing Cell Design. In *First International Workshop on Knowledge Discovery and Data Mining, 2008. WKDD 2008.*, 72-75.

[4] OAKLEY, B., FELDER, R.M., BRENT, R., and ELHAJJ, I., 2004. Turning student groups into effective teams. *Journal of student centered learning 2*, 1, 9-34.

[5] RIVARD, R., 2013. Measuring the MOOC dropout rate. *Inside Higher Ed, March 8, 2013*.

Educational Data Mining and Analyzing of Student Learning Outcome from the Perspective of Learning Experience

SHU Zhong-mei
School of Education, Sun Yat-sen University
No. 135, Xingang Xi Road
Guangzhou, 510275, P. R. China
isszm@mail.sysu.edu.cn

QU Qiong-fei
School of Education, Sun Yat-sen University
No. 135, Xingang Xi Road
Guangzhou, 510275, P. R. China
quqf@mail.sysu.edu.cn

FENG Lu-qi
School of Education, Sun Yat-sen University
No. 135, Xingang Xi Road
Guangzhou, 510275, P. R. China
fenglq3@mail2.sysu.edu.cn

ABSTRACT

Student Learning Outcome (SLO) has been a hot issue in the research fields of higher education quality assurance and institutional research. Based on the classic college student development theories and students' learning experiences, this paper combines two educational data mining techniques, regression analysis and neural network modeling to explore the influential mechanism of SLO by the means of empirical analysis. Finally, from the perspective of learning analytics and considering students' individual factors and university factors, a predication model of SLO is constructed to provide universities with essential reference to enhance SLO and teaching quality.

Keywords

Student Learning Outcome; Educational Data Mining; Learning Experience; Learning Analytics

1. INTRODUCTION

SLO has been a hot issue in the international research fields of higher education quality assurance, providing important evidence to reflect the educational effectiveness of a university [1]. A major part of the assessment of SLO is to obtain information and evidence of learning outcomes with qualitative or quantitative measurement methods. However, the diverse categories and levels of universities and colleges, as well as the complicated learning objectives and learning process, result in the multi-dimension and complexity of college-SLO.

It is recognized that the academic research on SLO has made a marked progress. In particular, research that explores models of SLO and their influential mechanism regarding the environments and conditions of schools, students' individual characteristics and student engagement in learning, contribute significantly to the decisions on how to promote SLO. Yet, there is little research examining SLO from the perspective of learning experience, for instance, student emotion, behavior and cognition. Empirically, students' participation in school activities and their rich experiences will influence their learning outcomes to a certain extent. And through those activities and experience they will have various cognitive experiences and respond with different emotional reactions. The data involved are huge and extensive that it becomes difficult for traditional statistical methods to discover the hidden laws. As a result, emerging data statistics and analysis methods, such as educational data mining, are urgently needed. In order to better understand the influential factors of SLO, this research uses stepwise regression analysis and neutral network to mine data of SLO, and reconstruct the meaning of mining results from the perspective of learning analytics. The ultimate purpose is to establish the intricate relationship between student learning experience, school characteristics, student characteristics and SLO, and thereby provide crucial reference for improving the training quality of university talents.

2. INSTRUMENT OF THE SURVEY

Research has shown that questionnaire survey precisely reflects the overall level of SLO by indirectly measuring learning outcomes with students' self-reports. This study devises the Sun Yat-sen University Student Learning Status Survey based on Astin's and Pace's student survey assessment models [2, 3]. Students' relevant experiences are emphasized, for instance, how much they involve in learning and work hard on it, and their interactions with teachers. Besides, students' emotional learning outcomes and general-knowledge education outcomes are also included. The survey divides student learning experiences and outcomes into 6 dimensions, illustrated in Figure 1. The coefficient of internal consistency among items in each dimension is above 0.9, demonstrating a high degree of reliability. This research analyzes the university and student factors' impacts on SLO in the logical framework of "Inputs-Learning-Outcomes", and from the standpoint of the broad learning experiences like student emotion, behavior and cognition.

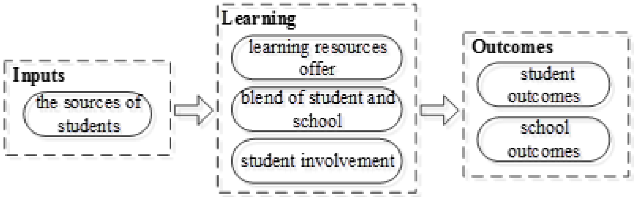


Figure 1. The Instrument of the Survey.

3. ANALYSIS

Educational data mining shows its potential value in determining the influential factors of SLO, such as identifying student characteristics and the dimensions of learning experiences that really affect learning outcomes in datasets. This paper uses educational data mining to deal with SLO data, interprets the mining results from the perspective of learning analytics, and explores the influential mechanism of SLO imposed by student learning experience, school characteristics, and student characteristics.

Data of this study came from the campus-wide online survey officially conducted in 2012, which was part of the Sun Yat-sen University Student Learning Status Survey Project, covering 36 departments and 33,000 undergraduates. These students completed the items in the questionnaire under the circumstance without stress. A total of 7,051 questionnaires were returned with a 21.2%

response rate, representing a considerable satisfaction compared with the international response rate of questionnaires. This research focuses on the samples of undergraduates, and selects 6,673 effective questionnaires out of the total returned with an effective rate of 94.6% based on principles such as response time and questionnaire quality standard.

3.1 Determining the Influential Factors of SLO with Regression

Stepwise regression provides an approach to identify the concrete experiences relevant to SLO. Specifically, it groups together similar items among the total 227-item in the survey, identifies those related to learning outcomes using forward and backward stepwise regression, thoroughly examines the residual plot and the diagnostics, and ultimately determines 17 independent variables in the multivariate regression model. 4 dimensions with 17 variables in student learning experiences are important factors affecting SLO, which are respectively the availability of learning resources, student involvement in learning, campus culture and school outcomes, including university factors like assessment of coursework and major study experience, guidance of academic norms, equal cultures and the atmosphere of cultivating multiple abilities, together with student factors like self-regulated learning, activity engagement, extra-curricular reading, thesis writing, peer communications, discussion contents, student-teacher interactions, academic activities and allocation of personal spare time. While school outcomes combining student and university factors, for instance, satisfactions towards school experiences and capability cultivation, as well as overall satisfaction, also have certain influential power on student learning outcomes.

These findings are consistent with Vincent Tinto's theoretical model on dropout problems of college students [4]. Whether students could obtain better learning outcomes depends on how well they fit their own experiences and objectives into the academic and social systems within the school system.

3.2 Optimizing Predictions of Learning Outcomes by Neutral Network Modeling

On the basis of the above analysis and for the purpose of increasing predictive accuracy, this study optimizes predictions of learning outcomes by neutral network modeling. The authors consider the influential factors of SLO determined by the regression analysis as experts' prior knowledge, and then further promote the intelligent processing through optimization model of neutral network, achieving the optimized prediction of SLO.

The number of input-layer nodes is determined by the number of factors affecting SLO, that is, 17 independent variables identified by regression analysis are considered as the input-layer nodes in the neutral network. Meanwhile, SLO is identified as the output-layer node. Finally, the optimized number of hidden-layer nodes is set to be 7 with the method of cut-and-trial. As such, the optimum topological structure for predicting SLO based on the neutral network model is 17-7-1.

Among the 6,673 effective questionnaires from the student learning condition survey, 8 samples with missing values have been eliminated. The 6,665 samples left are divided into two subsets, with 4,680 training samples (70.2%) for network model training, and 1,985 testing samples (29.8%) for testing whether the model meets the fundamental function required. Given the minimum relative change of training deviation is .0001 and that of training error is .001, and considers 1, 985 samples as testing samples. Through experiments, the output O represents the prediction of SLO, which ranges within [0, 1]. According to the overall evaluation of SLO by the 5-level rankings, i.e., A($0.9<=O$), B($0.8<= O <0.9$), C($0.7<= O <0.8$), D($0.6<= O <0.7$) , E($O <0.6$), the output has a sound consistency and accuracy with the self-assessment results of learning outcomes by the respondents. Table 1 shows a random list of the comparisons between network output and students' self-assessment results. Due to the large number of samples, Table 1 does not exhibit all the results.

Table 1. Sample Comparison

Testing Sample	Output	Expected	Relative deviation	Level
Sample 1	0.53	0.53	0.00	E
Sample 2	0.71	0.72	0.01	C
Sample 3	0.64	0.66	0.02	D
Sample 4	0.63	0.62	0.01	D
Sample 5	0.73	0.72	0.01	C
Sample 6	0.66	0.65	0.01	D
Sample 7	0.82	0.83	0.01	B
Sample 8	0.92	0.92	0.00	A
Sample 9	0.90	0.91	0.01	A

4. CONCLUSIONS

Predictive model of student learning outcomes which is constructed by the combination of neutral network and experts' prior knowledge established in the regression analysis, has a simple structure, better objectivity and accurate predictive effects. The trained neutral network model mentioned above could be used to scientifically and appropriately predict and evaluate student learning outcomes.

5. ACKNOWLEDGMENTS

We would like to acknowledge the funding on Key project of Ministry of Education (China) grant DIA130303 as well as funding on NSFC grant 61202345.

6. REFERENCES

[1] Terenzini, P.T. Assessment with open eyes: pitfalls in studying outcomes [J]. The Journal of Higher Education, 1989, 60(6): 644-64.

[2] Astin A W. Assessment for Excellence: the Philosophy and Practice of Assessment and Evaluation in Higher Education [M]. Phoenix, AZ: Oryx, 1993.

[3] Pace,C.R. Quality, Content, and Context in the Assessment of Student Learning and Development in College[M]. Washington, DC: Office of Educational Research and Improvement. 1986.

[4] Tinto, V. Dropout from Higher Education: A Theoretical Synthesis of Recent Research [J]. Review of Educational Research, 1975 (45):89-125.

Using EEG in Knowledge Tracing

Yanbo Xu Kai-min Chang Yueran Yuan Jack Mostow

Carnegie Mellon University

{Yanbox, kkchang, yuerany, mostow}@cs.cmu.edu

ABSTRACT

Knowledge tracing (KT) is widely used in Intelligent Tutoring Systems (ITS) to measure student learning. Inexpensive portable electroencephalography (EEG) devices are viable as a way to help detect a number of student mental states relevant to learning, e.g. engagement or attention. This paper reports a first attempt to improve KT estimates of the student's hidden knowledge state by adding EEG-measured mental states as inputs. Values of *learn*, *forget*, *guess* and *slip* differ significantly for different EEG states.

Keywords

EEG, knowledge tracing, Logistic regression.

1. Introduction

Knowledge tracing (KT) is widely used in Intelligent Tutoring Systems (ITS) to measure student learning. In this paper, we improve KT's estimates of students' hidden knowledge states by incorporating input from inexpensive EEG devices. EEG sensors record brainwaves, which result from coordinated neural activity. Patterns in these recorded brainwaves have been shown to correlate with a number of mental states relevant to learning, e.g. workload [1], associative learning [2], reading difficulty [3], and emotion [4]. Importantly, cost-effective, portable EEG devices (like those used in this work) allow us to collect longitudinal data, tracking student performance over months of learning.

Prior work on adding extra information in KT includes using student help requests as an additional source of input [5] and individualizing student knowledge [6]. Here we use students' longitudinal EEG signals as input to dynamic Bayes nets to help trace their knowledge of different skills. An EEG-enhanced student model allows unobtrusive assessment in real time. The ability to detect learning while it occurs instead of waiting to observe future performance could accelerate instruction dramatically. Current EEG is much too noisy to detect learning reliably on its own. However, as this paper shows, adding EEG to KT may allow better detection of learning than using KT alone.

2. Approach

KT is a Hidden Markov Model using a binary latent variable ($K^{(i)}$) to model whether a student knows the skill at step i. It estimates the hidden variable from its observations ($C^{(i)}$'s) in previous steps of whether the student applied the skill correctly. KT usually has 4 (sometimes 5) probabilities as parameters: initial knowledge (L_0), learning rate (t), forgetting rate (f) (usually assumed to be zero, but not in this paper), guess rate (g), and slip rate (s).□We add another observed variable ($E^{(i)}$), representing the EEG measured mental state estimated from EEG signals and time-aligned to the student's performance at step i.

EEG-derived signals are often described as a type of measure of human mental states. For example, NeuroSky uses EEG input to derive proprietary attention and meditation measures claimed to indicate focus and calmness [7]. We hypothesize that a student may have a higher learning rate t and/or a lower slip rate s when focusing or calm at a given step. Thus EEG-KT, shown in Figure 1, extends KT by adding variable $E^{(i)}$ computed from EEG input.

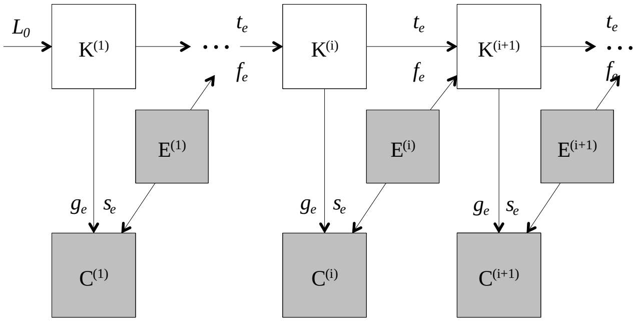


Figure 1. EEG-KT uses a binary EEG measure in KT

3. Evaluation and Results

To evaluate this approach, we compare EEG-KT to the original KT on a real data set. Our data comes from children 6-8 years old who used Project LISTEN's Reading Tutor at their primary school during the 2013-2014 school year [8]. We measure the growth of oral reading fluency by labeling a word as fluent if it was accepted by the automatic speech recognizer (ASR) as read correctly without hesitating or clicking on it for help.

EEG raw signals are collected by NeuroSky BrainBand at 512 Hz, and are denoised as in Chang et al. [3]. We use NeuroSky's proprietary algorithm to generate 4 channels: signal quality, attention, meditation, and rawwave. We then use Fast Fourier Transform to generate 5 additional channels from rawwave: delta, theta, alpha, beta, and gamma. In total, excluding signal quality, we obtain 8 EEG measures. We also compute a confidence-of-fluency (Fconf) metric as our 9th EEG measure by using training pipeline similar to [9]. It pre-balances the data by under-sampling, computes the average and variance of each channel's values over each word's duration as 16 features, and trains Gaussian Naïve Bayes classifiers to predict fluency (61.8% accurate, significantly above chance with $p < 0.05$ in Chi-squared test). We compute Fconf as Pr(fluent | 16 features) – Pr(disfluent | 16 features).

We normalize each of the 9 measures within student, discretize it as a binary variable (TRUE if above zero; FALSE otherwise), and use it to fit an EEG-KT model. We also evaluate Rand-KT, which replaces EEG with randomly generated values from a Bernoulli distribution. We use EM algorithms to estimate the parameters, and implement the models in Matlab Bayesian Net Toolkit for Student Modeling (BNT-SM) [10, 11].

The data has 6,313 observations from 12 students, with 83% labeled as fluent. We use leave-1-student-out cross-validation (CV), which trains word-specific models on 11 out of 12 students and tests on the remaining single student. To maintain enough data for EM to estimate the parameters, we keep 4 students who have many more than 500 observations in the training data and cross-validate only the other 8 students. We use AUC (area under the curve) to assess model prediction, as shown in Table 1. Fconf-KT and Theta-KT beat KT, but not significantly. The other 7 models did worse than KT, the bottom 5 significantly so.

Table 1. AUC scores by 8-fold CV

(underlined if p <0.05 in pair[1]ed t-test comparison to KT)

Models	AUC	Models	AUC
Fconf-KT	0.6613	Gamma-KT	0.6317
Theta-KT	0.6568	RAW-KT	0.6275
KT	**0.6479**	MED-KT	0.6230
ATT-KT	0.6435	Delta-KT	0.6224
Alpha-KT	0.6429	Rand-KT	0.6146
Beta-KT	0.6355		

Table 2 reports the estimated parameters for the two most interpretable EEG measures, meditation and attention. Students in a meditative state according to EEG were significantly less likely to forget, guess, or slip. Students in an attentive state according to EEG were significantly less likely to forget or slip.

Table 2. Avg. estimated parameters in EEG-KT across words

(underlined if p < 0.05 in paired t-test across high/low state)

	Meditation		Attention	
Parameters	High	Low	High	Low
t_e	0.32	0.33	0.38	0.43
f_e	0.10	0.25	0.15	0.30
g_e	0.53	0.62	0.55	0.56
s_e	0.03	0.07	0.03	0.08

4. Conclusion and Future Directions

To improve KT's estimates of students' hidden knowledge states, we tried adding different binary EEG measures as an input. This simple approach produced significantly different estimates of forgetting, guessing, and slip rates according to the attention and meditation indicators, but did not improve model fit significantly. Our subsequent approach achieved much higher accuracy (AUC .7665) by using logistic regression to merge EEG measures [12].

With months of data and many words per minute, fluency development offers a rich domain for studying EEG-enriched KT, but it can apply to other types of learning as well. Another future direction is to analyze its practical impact on learning. As Beck and Gong [13] pointed out, tiny improvements in predictive accuracy don't matter -- actionable intelligence does. We want to estimate the possible speedup in learning from using EEG to detect it as it occurs rather than wait to see it in later performance.

5. Acknowledgements

This work was supported in part by the National Science Foundation under Cyberlearning Grant IIS1124240. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

6. References

[1] Berka, C., D.J. Levendowski, M.N. Lumicao, A. Yau, G. Davis, V.T. Zivkovic, T. Vladimir, R.E. Olmstead, P.D. Tremoulet, D. Patrice, and P.L. Craven. EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks. *Aviation, Space, and Environmental Medicine*, 2007. *78 (Supp 1)*: p. B231-244.

[2] Miltner, W.H.R., C. Braun, M. Arnold, H. Witte, and E. Taub. Coherence of gamma-band EEG activity as a basis for associative learning. *Nature*, 1999. *397*: p. 434-436.

[3] Chang, K.M., J. Nelson, U. Pant, and J. Mostow. Toward Exploiting EEG Input in a Reading Tutor. *International Journal of Artificial Intelligence in Education*, 2013. *22*(1-2): p. 19-38.

[4] Heraz, A. and C. Frasson. Predicting the three major dimensions of the learner's emotions from brainwaves. *World Academy of Science, Engineering and Technology*, 2007. *31*: p. 323-329.

[5] Beck, J.E., K.-m. Chang, J. Mostow, and A. Corbett. Does help help? Introducing the Bayesian Evaluation and Assessment methodology. *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, 383-394. ITS2008 Best Paper Award. 2008. Montreal.

[6] Pardos, Z. and N. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. *Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization*, 255-266. 2010. Big Island, Hawaii.

[7] NeuroSky. NeuroSky's eSense™ meters and detection of mental sate. 2009, Neurosky, Inc. At http://www.neurosky.com/Documents/Document.pdf?DocumentID=809fde40-0fa6-4ab6-b7ad-2ec27027e4eb.

[8] Mostow, J. and J.E. Beck. When the Rubber Meets the Road: Lessons from the In-School Adventures of an Automated Reading Tutor that Listens. In B. Schneider and S.-K. McDonald, Editors, *Scale-Up in Education*, 183-200. Rowman & Littlefield Publishers: Lanham, MD, 2007.

[9] Yuan, Y., K.-m. Chang, Y. Xu, and J. Mostow. A Public Toolkit and ITS Dataset for EEG. *Proceedings of the ITS2014 Workshop on Utilizing EEG Input in Intelligent Tutoring Systems* 2014. Honololu.

[10] Chang, K.M., J.E. Beck, J. Mostow, and A. Corbett. A Bayes net toolkit for student modeling in intelligent tutoring systems. In *8th International Conference on Intelligent Tutoring Systems*. 2006: Jhongli, Taiwan, p. 104-113. At http://link.springer.com/chapter/10.1007%2F11774303_11?LI=true.

[11] Xu, Y. and J. Mostow. Extending a Dynamic Bayes Net Toolkit to Trace Multiple Subskills. *25th Florida Artificial Intelligence Research Society Conference (FLAIRS-25)*, 574. 2012. Marco Island, Florida.

[12] Xu, Y., K.-m. Chang, Y. Yuan, and J. Mostow. EEG Helps Knowledge Tracing! In *Proceedings of the ITS2014 Workshop on Utilizing EEG Input in Intelligent Tutoring Systems*. 2014: Honolulu.

[13] Beck, J.E. and Y. Gong. Wheel-spinning: Students who fail to master a skill. *The 16th International Conference on Artificial Intelligence in Education*, 431-440. 2013. Memphis, Tennessee.

Exploring Engaging Dialogues in Video Discussions

I-Han Hsiao
EdLab, Teachers College,
Columbia University.
525 W. 120th Street
New York, NY, 10027, USA
+1 212 678 3448
ih2240@columbia.edu

Hui Soo Chae
EdLab, Teachers College,
Columbia University.
525 W. 120th Street
New York, NY, 10027, USA
+1 212 678 3448
chae@tc.columbia.edu

Manav Malhotra
EdLab, Teachers College,
Columbia University.
525 W. 120th Street
New York, NY, 10027, USA
+ 1 212 678 3448
mm2625@columbia.edu

Ryan S.J.d. Baker
Teachers College,
Columbia University.
525 W. 120th Street
New York, NY, 10027, USA
+1 212 678 8329
baker2@exchange.tc.columbia.edu

Gary Natriello
EdLab, Teachers College,
Columbia University.
525 W. 120th Street
New York, NY, 10027, USA
+1 212 678 3087
gjn6@columbia.edu

ABSTRACT

Learning from dialogues is a powerful pedagogy. Video-based and dialogic learning have become increasingly commonplace over the last decade and gradually evolve as one of the most popular teaching & learning strategies for modern e-learning (i.e. MOOCs). Identifying high-quality video dialogues is increasingly challenging because of the sheer number of video discussions being produced daily. In this paper, we explore online video discussions by considering both structural discourse of discussion and user interaction.

Keywords

dialogue-based learning, video discussion, vialogue, engagement

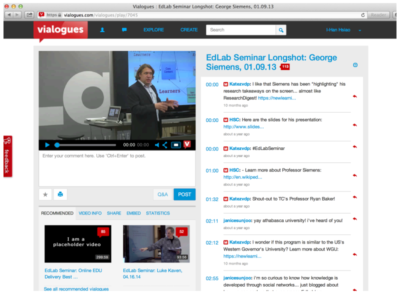
1. INTRODUCTION

Learning from dialogues is a powerful pedagogy, which involves several diverse cognitive instructional strategies, such as self-explanation, scaffolding tutorial dialogues, group discussions and among others [1-2]. The juncture of ITS/AIED & Learning Science literature has successfully demonstrated that students can learn from a wide range of such dialogue-based instructional settings [3-6]. Recently, studies show an alternative instructional context by *learning from observing others learn* [3] and is considered as a promising learning paradigm [4] due to such paradigm addresses the major limitations on development time in ITSs & liberated the domains from procedural skills to less structured fields. However, less is explored is whether such paradigm can be successfully applied on discussions around videos or other multimedia, which is one of the most popular teaching & learning strategies for modern e-learning (i.e. MOOCs).

Video-based and dialogic learning are not only becoming increasingly commonplace as research over the last decade, but more importantly, discussing within a multimedia-rich environment creates a wide range of educational benefits [7-10]. Essentially, discussing around videos includes more complex interactions rather than having dialogues alone/among groups or merely performing video annotations. In addition, given the accelerated pace of online media generation and discussion around that media, identifying high-quality video discussions is increasingly challenging and important. In fact, the task requires more than just text analysis. Interactions on video discussions tend to be more sophisticated than posts and replies on discussion forums. Much of the time spent on video discussions is with the video rather than static objects in traditional forums (i.e. reading a post, an article or an image and reflect by responding text). Moreover, users' engagement can be heavily influenced by the user interface and its associated process flow [11]. Therefore, in this paper, we attempt to address these challenges by exploring the engagement activities in video dialogues. The goal of this project is to model the rich user interactions and structural discourse of video dialogues for deeper inferences on users' engagement.

2. VIALOGUES: VIDEO DIALOGUES

Vialogues is a video-based discussion tool purposively devised for reflective adaptive collaborative learning. We provide a brief overview on the core features of Vialogues in Figure 1. Vialogues allows users to comment directly on specific portions of a video, as opposed to only posting comments on a discussion board that references an entire video. All the comments are *time coded* to a specific point in the video. Thus, the comments and related portions of the video can be mutually referenced. Detail design rationales were published in [9].


Vialogues : EdLab Seminar Longshot: George Siemens, 01.09.13
https vialogues.com/vialogues/play/7045
Reader
vialogues
EXPLORE
CREATE
Search
I-Han Hsiao
feedback
00:00 00:00
Q&A
POST
RECOMMENDED
VIDEO INFO
SHARE
EMBED
STATISTICS
I am a placeholder video
85
299:59
52
93:56
EdLab Seminar: Online EDU Delivery Best ...
EdLab Seminar: Luke Kaven, 04.16.14
See all recommended vialogues
EdLab Seminar Longshot: George Siemens, 01.09.13
113

Enter your comment here. Use 'Ctrl+Enter' to post.

00:00 Katezvdp: I like that Siemens has been "highlighting" his research takeaways on the screen... almost like ResearchDigest! https://newleami... 10 months ago

00:00 HSC: Here are the slides for his presentation: http://www.slides... about a year ago

00:00 Katezvdp: #EdLabSeminar about a year ago

01:00 HSC: - Learn more about Professor Siemens: http://en.wikiped... about a year ago

01:32 Katezvdp: Shout-out to TC's Professor Ryan Baker! about a year ago

02:11 janicesunjoo: yay athabasca university! i've heard of you! about a year ago

02:12 Katezvdp: I wonder if this program is similar to the US's Western Governor's University? Learn more about WGU: https://newleami... 10 months ago

02:55 janicesunjoo: i'm so curious to know how knowledge is developed through social networks... just blogged about

Figure 1. Vialogues, http://vialogues.com

Since December 2011, 3~5 vialogues have been featured on the Vialogues homepage weekly. At the moment of writing, there were 357 featured vialogues with a total of 3995 comments. Based on the reviewed literature, we selected three features to detect the most engaging comments from online video discussions: 1) Comment Syntactic, including basic discourse structure: comment length (total characters), word counts, words per sentence and comment density, comment novelty, comment readability; 2) Comment Semantic, including comment psychometrics and comment sentiments and 3) User Interactions, including Vilaogues moderation, user activity and user behavioral patterns.

3. EVALUATION

Our goal is to construct a generic model that can predict users' behavior based on the discussion structure, content and their interactions. To capture whether the observed assumptions on the features would account for the variation in engagement prediction, we performed logistic regression. Overall, the full model was able to successfully predict user behavior at $F(1, 334)= 3.25$, $p<.001$, $adjusted\text{-}R^2=0.139$. We tested the goodness of the models reserving 20% of the observations for testing with 10-fold cross validation ($\mathrm{MAE}_{10FOLD}=2.59$) and selected a final model.

3.1 Effects of Syntactic on Engagement

In predicting user engagement based on comment syntactic features, we found that *the relative position of the comment* to the video has a significant positive effect on user engagement. A possible explanation is that once a user starts playing a video, disregarding the video length, it is common to spend some time in the beginning getting oriented to the context and participate later. Such results provide very useful information for instructors or instructional designers to be aware of the natural tendency of "warming-up" phase of a discussion and can adaptively moderate discussions early on. We anticipated that the more novel words in the comments, the more engaged with the discussion a user might be. However, the results demonstrated otherwise. Possible reasons could be that new words or new information may be useful, but may also be distracting, losing user focus. Among other comment syntactic features, we see a tendency of less lengthy comments and slightly complex words tend to promote video discussion. Although these are not significant predictors of user engagement, we think the short and complex words phenomenon can be somehow attributed to the hashtags (#).

3.2 Effects of Semantics on Engagement

Some online discussion literature has already suggested that discussions may remain at a surface level, such as sharing or comparing information, without diving into deeper levels [11]. To prove that the vialogues effective promote meaningful discussion rather than surface level communications, we looked at the comment semantics. Based on the logistic regression model, we found that only cognitive words attributed significantly positive to users' engagement; perceptual and relative words negatively attributed to users' engagement, social and biological words were marginally negatively attributed to users' engagement; and emotional words (affection words or sentence sentiments) in the comments do not affect users' engagement. The results seemed to be counterintuitive to our understanding at the beginning; however, the results supported the design of vialogues to facilitate meaningful video discussions and appeared to be engaging when the comments are highly cognitive but not superficially conversational.

3.3 Effects on User Interactions

We found that the number of moderators' comments, the number of views and the number of timecode clicks are positively and significantly attributed to the vialogues engagement. However, there were significant negative correlations among the number of moderators, the moderation ratio, the number of vialogues were embedded and the number of vialogues being bookmarked as favorites. Such results revealed the importance on the comments quality instead of quantity. Meanwhile, users were found engaged with immediate interactions (*time-code* clicks to reference to specific video fragment and the comment) with the video discussions rather than *post*-interactions (such as favorite the vialogue or embedded it to elsewhere).

4. REFERENCES

[1] Aleven, Vincent, Ogan, Amy, Popescu, Octav, Torrey, Cristen, & Koedinger, Kenneth. (2004). Evaluating the Effectiveness of a Tutorial Dialogue System for Self-Explanation. In J. Lester, R. Vicari & F. Paraguaçu (Eds.), Intelligent Tutoring Systems (Vol. 3220, pp. 443-454): Springer Berlin Heidelberg.

[2] VanLehn, Kurt, Graesser, Arthur C., Jackson, G. Tanner, Jordan, Pamela, Olney, Andrew, & Rosé, Carolyn P. (2007). When Are Tutorial Dialogues More Effective Than Reading? Cognitive Science, 31(1), 3-62.

[3] Chi, Michelene T. H., Roy, Marguerite, & Hausmann, Robert G. M. (2008). Observing Tutorial Dialogues Collaboratively: Insights About Human Tutoring Effectiveness From Vicarious Learning. Cognitive Science, 32(2), 301-341.

[4] Muldner, Kasia, Lam, Rachel, & Chi, Michelene T. H. (2014). Comparing learning from observing and from human tutoring. Journal of Educational Psychology, 106(1), 69-85.

[5] Aleven, Vincent, McLaren, Bruce, Roll, Ido, & Koedinger, Kenneth. (2006). Toward Meta-cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. International Journal of Artificial Intelligence in Education, 16(2), 101-128.

[6] Boyer, Kristy Elizabeth, Phillips, Robert, Ingram, Amy, Ha, Eun Young, Wallis, Michael, Vouk, Mladen, & Lester, James. (2011). Investigating the Relationship Between Dialogue Structure and Tutoring Effectiveness: A Hidden Markov Modeling Approach. International Journal of Artificial Intelligence in Education, 21(1), 65-81

[7] Rich, Peter J., & Hannafin, Michael. (2009). Video Annotation Tools: Technologies to Scaffold, Structure, and Transform Teacher Reflection. Journal of Teacher Education, 60(1), 52-67.

[8] Derry, Sharon J., Pea, Roy D., Barron, Brigid, Engle, Randi A., Erickson, Frederick, Goldman, Ricki, . . . Sherin, Bruce L. (2010). Conducting Video Research in the Learning Sciences: Guidance on Selection, Analysis, Technology, and Ethics. Journal of the Learning Sciences, 19(1), 3-53.

[9] Agarwala, Megha, Hsiao, I-Han, Chae, Hui Soo, & Natriello, Gary. (2012). Vialogues: Videos and Dialogues Based Social Learning Environment. Pp.629-633, ICALT

[10] Hsiao, I-Han, Malhotra, Manav, Chae, Hui Soo, & Natriello, Gary. (2013). Dissecting Video Discussions and Coordination Strategies. Paper presented at the Computer Supported Collaborative Learning, Madison WI.

[11] Gunawardena, Charlotte N, Lowe, Constance A, & Anderson, Terry. (1997). Analysis of a global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing. Journal of Educational Computing Research, 17(4), 397-431.

Exploring indicators from keyboard and mouse interactions to predict the user affective state

Sergio Salmeron-Majadas
aDeNu Research Group. UNED
Calle Juan del Rosal, 16. Madrid
28040. Spain
+34 91 398 93 88
ssalmeron@bec.uned.es

Olga C. Santos
aDeNu Research Group. UNED
Calle Juan del Rosal, 16. Madrid
28040. Spain
+34 91 398 93 88
ocsantos@dia.uned.es

Jesus G. Boticario
aDeNu Research Group. UNED
Calle Juan del Rosal, 16. Madrid
28040. Spain
+34 91 398 91 97
jgb@dia.uned.es

ABSTRACT

Following a low cost and non-intrusive approach, in this paper we discuss how prediction rates from 5 different data mining algorithms using 4 different emotional labeling approaches differ when exploring the usage of keyboard and mouse interaction sources for affective states detection in a math problem solving experiment.

Keywords

Data Mining, Affective Computing, Affective States, Human-Computer Interaction, Keyboard, Mouse.

1. INTRODUCTION

Due to the existing relations between emotions and cognitive processes in learning, there is a need to take into account the learners' affective state when supporting the learning process [7]. With this context in mind, in this paper we explore the potential of using mouse and keyboard interaction data as affective information sources, which are low cost and non-intrusive. We compare the results obtained from them with those provided by alternative data sources, such as sentiment analysis and physiological signals.

The most common approach reported in the literature regarding emotion detection is based on using a single data source as affective indicator [2, 12]. Usually, keyboard and mouse interactions as well as physiological sensors are used. Regarding keyboard, keystroke features extracted from single events are used to detect affective states [2], although combined keystroke events indicators have also been considered [4]. On the mouse side, some works have used features such as speed or direction to detect affective states [12]. A review of different studies carried out to detect emotions from keyboard and mouse interactions can be found in [6]. Physiological sensors have been widely used with affective purposes, but usually using intrusive ways to get data [5].

2. EXPERIMENT & RESULTS

A math problem solving experiment was carried in our lab with 75 participants (details in [11]) in order to research how to detect affective states with data mining [9]. To gather emotional data we used different data sources: keyboard interactions (***K***), mouse interactions (***M***), webcam recording, computer screen recording, Kinect recording and physiological recording (i.e. heart rate, skin conductance, breath frequency and skin temperature) (***P***). The experiment collected participants' emotional baseline. The mathematical tasks consisted of 3 series of 6 problems. For each problem, participants had to select one answer from a set of 4 possibilities and fill in the 9-point Self-Assessment Manikin (SAM) [3] scale to report their valence (i.e. pleasure) and arousal (i.e. activation) state. After each group of problems (task), participants had to type their feelings about it. Emotions were elicited by giving less time than required to do some tasks, or changing their difficulty level. All along the experiment each participant had an affective tutor, who supervised the progress and took timestamps on the physiological recordings on every task beginning.

Representing the affective states occurred during a session is an open issue [8], so several approaches to emotionally label interactions were considered: i) SAM scores provided by participants during the experiment (***Label 1***), calculating the mean and standard deviation for each task; ii) SAM scores provided for each task by two psychologists (with experience in motivational and educational issues) after reading the corresponding emotional reports (***Label 2***); iii) a categorical classification (positive, negative, neutral and positive-negative) provided by another expert (with 10 years of experience in supporting learners in e-learning platforms) when reading those emotional reports (***Label 3***), and iv) the average value from the 9-point SAM scores per task given by the participant and the psychological experts (***Label 4***).

For data processing, indicators were grouped by task. For keyboard interactions, depending on the event aggregation performed, the indicators generated were the following: i) number of key press events, ii) average time between press events, iii) average time between a press and its following release event and iv) number of times a certain key or a group of keys has been pressed (backspace key, delete key, alphabetical characters keys, etc), and v) the indicators proposed in [4], which were generated from creating combinations of two or three keystrokes events. On the mouse interactions side, indicators were as follows: i) number of clicks (per button and aggregated); ii) overall distance; iii) covered distance (distance the cursor has traversed) between two button press events, between a button press and its following release event, between a button release and its following press event and between two button release events; iv) the Euclidean distance in the four previously described cases; v) the difference between the covered and the Euclidean distances calculated; and vi) time durations between the proposed combinations of events.

When processing the physiological signals, differences between the values in each task and the baseline value for each signal were calculated and used to compute the average for all those values. Additionally, sentiment analysis (***S***) was used to automatically generate an affective score for each emotional report, counting the number of positive and negative terms according to the MPQA Opinion Corpus affective database.

Following previous works [10], our goal here was to predict the valence dimension as higher correlations were found with valence than with arousal. As suggested in the literature [1], the 9-point valence values were grouped into three categories (i.e., positive (>6), negative (<4) and neutral (4-6)). Different algorithms were used, namely C4.5 (C), Naïve Bayes (N), Bagging (B), Random Forests (R) and AdaBoost (A). Results in Table 1 show the best prediction rate depending on the labelling and the data source used and the algorithm applied to achieve that rate. The analysis was done on the data from 17 participants, who are the ones whose interactions have already been emotionally labeled with the four aforementioned approaches. When processing the data, some filtering decisions were taken, such as removing the registers with SAM values per task with a standard deviation higher than 2, as well as the registers corresponding to neutral and positive-negative categories.

Table 1. Best prediction rates depending on the labels and the input data sources. Best result per data labeling is bolded.

	Label 1	Label 2	Label 3	Label 4
K	0,65 (C)	0,74 (B)	0,58 (C)	0,67 (R)
M	0,65 (C)	0,74 (B)	0,57 (R)	0,67 (R)
S	0,82 (R)	**0,83 (A)**	**0,66 (A)**	0,81 (C,B,A)
K+M	0,67 (R)	0,74 (B,R)	0,59 (R,A)	0,56 (R)
K+S	0,75 (C)	0,74 (B)	0,64 (R)	**0,86 (C)**
M+S	**0,85 (C)**	0,74 (B)	0,6 (A)	0,81 (R)
K+M+S	0,75 (B,R)	0,74 (B)	0,62 (B)	0,77 (A)
P	0,67 (C)	0,74 (B)	0,52 (C,R)	0,53 (C)

3. DISCUSSION & FUTURE WORK

From Table 1, sentiment analysis seems to be the best data source, but its results can be improved when combined with keyboard or mouse. This suggest that combining different data sources would produce improvements, but this should be clarified with further experiments as using different prediction algorithms and alternative labeling approaches seem to induce significant differences in the results. Up to our knowledge, there are no works in the literature that report a deep comparison of the benefits of each labeling approach. Due to this, it seems of interest to study different approaches to label emotions by performing a comparative analysis using a large number of algorithms depending on their predictive features (using feature selection techniques). Another future step of interest to take is exploring the idea of mining these sources separately and then mining the obtained outputs, in search for a system that would be able to automatically choose the data source to be used depending on their individual success in the prediction.

ACKNOWLEDGMENTS

Authors would like to thank MAMIPEC project (TIN2011-29221-C03-01) colleagues for their support and the Spanish Government for its funding. Sergio Salmeron-Majadas acknowledges the FPI grant (BES-2012-054522).

REFERENCES

[1] Baran, B., Pace-Schott, E.F., Ericson, C. and Spencer, R.M.C. 2012. Processing of Emotional Reactivity and Emotional Memory over Sleep. *The Journal of Neuroscience*. 32, 3 (Jan. 2012), 1035–1042.

[2] Bixler, R. and D’Mello, S. 2013. Detecting Boredom and Engagement During Writing with Keystroke Analysis, Task Appraisals, and Stable Traits. *Proceedings of the 2013 International Conference on Intelligent User Interfaces* (New York, NY, USA, 2013), 225–234.

[3] Bradley, M.M. and Lang, P.J. 1994. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*. 25, 1 (1994), 49–59.

[4] Epp, C., Lippold, M. and Mandryk, R.L. 2011. Identifying emotional states using keystroke dynamics. *Proceedings of the 2011 annual conference on Human factors in computing systems* (2011), 715–724.

[5] Handri, S., Yajima, K., Nomura, S., Ogawa, N., Kurosawa, Y. and Fukumura, Y. 2010. Evaluation of Student’s Physiological Response Towards E-Learning Courses Material by Using GSR Sensor. *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on* (2010), 805–810.

[6] Kolakowska, A. 2013. A review of emotion recognition methods based on keystroke dynamics and mouse movements. *2013 The 6th International Conference on Human System Interaction (HSI)* (2013), 548–555.

[7] O’Regan, K. 2003. Emotion and e-learning. *Journal of Asynchronous learning networks*. 7, 3 (2003), 78–92.

[8] Porayska-Pomsta, K., Mavrikis, M., D’Mello, S., Conati, C. and Baker, R.Sj. 2013. Knowledge elicitation methods for affect modelling in education. *International Journal of Artificial Intelligence in Education*. 22, 3 (2013), 107–140.

[9] Salmeron-Majadas, S., Santos, O.C. and Boticario, J.G. 2013. Affective State Detection in Educational Systems through Mining Multimodal Data Sources. *6th International Conference on Educational Data Mining* (Memphis, 2013), 348–349.

[10] Salmeron-Majadas, S., Santos, O.C. and Boticario, J.G. 2014. An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective modeling in an educational context. *18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems* (Poland, 2014), in press.

[11] Saneiro, M., Santos, O.C., Salmeron-Majadas, S. and Boticario, J.G. 2014. Towards Emotion Detection in Educational Scenarios from Facial Expressions and Body Movements through Multimodal Approaches. *The Scientific World Journal*. 2014, (Apr. 2014), e484873.

[12] Tsoulouhas, G., Georgiou, D. and Karakos, A. 2011. Detection of Learner’s Affective State Based on Mouse Movements. *Journal of Computing*. 3, 11 (Nov. 2011), 9–18.

Extracting Latent Skills from Time Series of Asynchronous and Incomplete Examinations

Shinichi OEDA
Department of Information and Computer Engineering,
Kisarazu National College of Technology,
CREST, JST
oeda@j.kisarazu.ac.jp

Yu ITO
Department of Mathematical Informatics,
The University of Tokyo
yu_ito@mist.i.u-tokyo.ac.jp

Kenji YAMANISHI
Department of Mathematical Informatics,
The University of Tokyo,
CREST, JST
yamanishi@mist.i.u-tokyo.ac.jp

ABSTRACT

Examinations are tools for measuring examinees' skills. A question item in an examination requires several skills to solve it. In order to grasp latent skills, it is important to find which skills an item requires. The relationship between items and skills can be represented by a Q-matrix. Recent studies have attempted to extract a Q-matrix by non-negative matrix factorization (NMF) from a set of examinees' test scores. In order to apply NMF, examination results without missing values are required as the matrix to be decomposed. However, it is difficult to assemble complete examination results because users of intelligent tutoring systems solve different items at different times. In this paper, we propose a method which extracts a Q-matrix by aggregating incomplete examination results asynchronously.

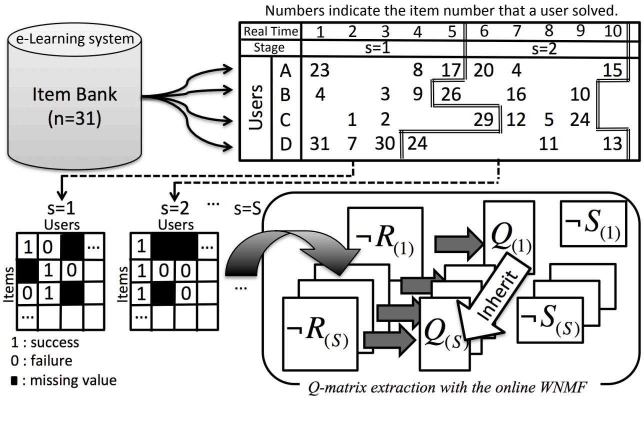
1. INTRODUCTION

The concept of a Q-matrix was developed on the basis of the rule space method (RSM) by Tatsuoka et al. [4]. A Q-matrix allows us to determine which skills are necessary to solve each item of an examination. Recently, there have been several studies on how to extract a Q-matrix from a set of examination results [1, 2]. These studies applied the non-negative matrix factorization (NMF) method to decompose the results of an examination into a Q-matrix and an S-matrix (which gives the relationship between skills and users). In particular, the online NMF with regularization (online NMF) has been proposed in order to extract a constant Q-matrix from examination time series in an online fashion [2]. Although these NMF methods require the input matrix to have no missing values in order to be able to factorize it, real examination results have many missing values because users of general intelligent tutoring systems (ITSs) do not always solve all items. In this paper, we introduce a novel method for extracting a time-invariant Q-matrix from a time series of asynchronous and incomplete examination results. The key ideas are as follows: 1) to synchronize time-series data per *"stage"* (the period during which a fixed number of items are given) to obtain multiple item-user matrices having missing values and 2) to apply the weighted NMF [3] to each matrix in an online manner as with the online NMF. We empirically demonstrate the effectiveness of the proposed method by using artificial data sets.

2. COLLECTING EXAMINATION RESULTS FROM REAL ITS

In the Q-matrix extraction by the online NMF [2], the Q-matrix $\mathbf{Q}$ was assumed to be constant, while S-matrix $\mathbf{S}_t$ varied over time because users acquired knowledge by learning and experiences. An examination result $\mathbf{R}_t$ changed whenever $\mathbf{S}_t$ changed because $\neg\mathbf{R}_t$ was obtained according to the equation $\neg\mathbf{R}_t = \mathbf{Q} \circ (\neg\mathbf{S}_t)$, where the operator $\neg$ denotes Boolean negation. In order to extract a constant Q-matrix from such variable examination results, the key idea of the online NMF was that the initial values of the matrix are inherited from the Q-matrix of the previous decomposition; this is in contrast to the conventional NMF-based method, in which the initial values are set at random.

However, in a real ITS, it is impossible to collect all users' answers as examination results because each user solves different items at different times, even the online NMF needs all answers. In this paper, we introduce the novel concept of a *stage* to resolve this problem. We define one stage as a period during which a fixed number of items are given and a fixed number of skills are required for each stage. In this paper, one stage is defined as when items in an ITS are given c items to a user. Elements of $\neg\mathbf{R}_s$ are collected as users' results from every stage s. The overall flow of the proposed method is shown in Figure 1. In this figure, the $\neg\mathbf{R}_s$ are constructed using a stage with $c = 3$.



Numbers indicate the item number that a user solved.

Real Time		1	2	3	4	5	6	7	8	9	10
Stage		s=1					s=2				
Users	A	23			8	17	20	4			15
	B	4		3	9	26		16		10	
	C		1	2			29	12	5	24	
	D	31	7	30	24				11		13

Figure 1: Overall flow of the proposed method

3. Q-MATRIX EXTRACTION FROM INCOMPLETE EXAMINATION RESULTS

Whereas the online NMF needs a filled matrix, the examination results collected as $\neg\mathbf{R}_s$ have missing values. In order to factorize the incomplete matrix, we apply the WNMF. The WNMF can cope with missing values in an observed matrix. Suppose $\mathbf{W}$ is a binary matrix of the same size as $\neg\mathbf{R}$ such that $\mathbf{W}_{ij} = 1$ when $\neg\mathbf{R}_{ij}$ is known and $\mathbf{W}_{ij} = 0$ when $\neg\mathbf{R}_{ij}$ is missed. The update rules of the extraction of the Q-matrix with WNMF are as follows:

$$\mathbf{Q}_{ik} \leftarrow \mathbf{Q}_{ik} \frac{((\mathbf{W} * \neg\mathbf{R})\neg\mathbf{S}^\top)_{ik}}{((\mathbf{W} * (\mathbf{Q}\neg\mathbf{S}))\neg\mathbf{S}^\top)_{ik}}, \tag{1}$$

$$\neg\mathbf{S}_{kj} \leftarrow \neg\mathbf{S}_{kj} \frac{(\mathbf{Q}^\top(\mathbf{W} * \neg\mathbf{R}))_{kj}}{(\mathbf{Q}^\top(\mathbf{W} * (\mathbf{Q}\neg\mathbf{S})))_{kj}}, \tag{2}$$

where $*$ denotes element-wise multiplication.

The online WNMF produces a Q-matrix by letting the initial values be those obtained at the previous stage. The cost function of the online WNMF can be written as

$$\min_{\mathbf{Q}_s,\mathbf{S}_s} \{\|\neg\mathbf{R}_s - \mathbf{Q}_s\neg\mathbf{S}_s\|_F^2 + \lambda(s)(\|\mathbf{Q}_{s-1} - \mathbf{Q}_s\|_F^2)\}, \tag{3}$$

where $\lambda(s)$ is a monotonic increasing function of time given by $\lambda(s) = \alpha s/S$, where s is a stage in $(1, \ldots, S)$, and α is the constant parameter determining the rate of increase. At each stage s, we find $\mathbf{Q}_s$ and $\neg\mathbf{S}_s$ according to (3), so that the sum of the factorization error and regularization term is minimized. In the optimizations with respect to $\mathbf{Q}_s$ and $\neg\mathbf{S}_s$, we set $\mathbf{Q}_s$ by inheriting $\mathbf{Q}_{s-1}$, and choose $\neg\mathbf{S}_s$ by taking random non-negative values.

4. EXPERIMENTAL RESULTS

In order to verify the effectiveness of our methods, we made a synthetic examination time series. We generated a time-varying S-matrix and a fixed Q-matrix to obtain $\neg\mathbf{R}_s$ according to the equation $\neg\mathbf{R}_s = \mathbf{Q} \circ (\neg\mathbf{S}_s)$. A conjunctive Q-matrix consisted of 31 items and 5 skills. We designed a time series of $\neg\mathbf{S}_s$ as a process of acquiring skills, on the basis of the item response theory.

As a measure of the performance for Q-matrix extraction, we introduce a Q-matrix error e_s between $\mathbf{Q}$ and an extracted matrix $\widehat{\mathbf{Q}}_s$ as $e_s = \|\widehat{\mathbf{Q}}_s - \mathbf{Q}\|_F^2$. Note that the factorized solutions obtained using the NMF may not be unique due to the randomness of the initial matrices. Hence, we calculated the mean and standard deviation of Q-matrix errors from 10 simulations.

To begin with, we need to investigate the factorized performance of WNMF which concerns with the missing rate of an input matrix. The matrix was made by simulating random $\{0, 1\}$ as a filled matrix. We made incomplete matrices from the matrix by giving various masks with different missing rate to it. Figure 2 shows the relationship between factorized errors and missing rates of matrices. The factorized errors are low when the missing rates of matrices are less than 30%. As a result, we defined one stage as $c = 23$, namely the missing rate of each $\mathbf{R}_s$ was 25% because a user solved 23 out of 31 items. Figure 3 shows the Q-matrix errors for both the WNMF and the online WNMF for examination results having a missing rate of 25%. Although, in the WNMF only, the Q-matrix error did not become zero at any stage and the error gradually increased after stage=8, the online WNMF overcame this problem.

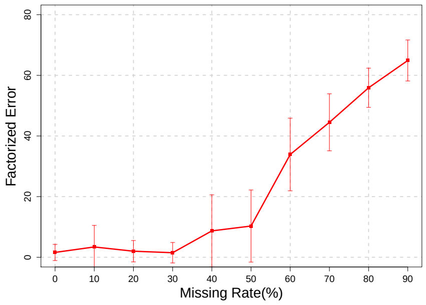


Figure 2: The correlation between factorized errors and missing rates of an input matrix.

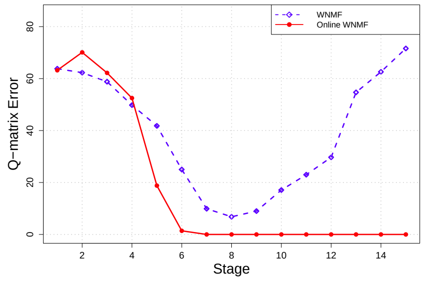


Figure 3: The Q-matrix errors with the WNMF and the online WNMF over missing rate 25%.

5. CONCLUSIONS

In this paper, we have introduced the concept of a stage to collect users' answers asynchronously and have proposed the online WNMF for the purpose of extracting a constant Q-matrix from a time series of incomplete examination results. We have designed the method so as to decompose examination results with missing values. Finally, we applied the proposed method to a synthetic data set to demonstrate that it could find a constant Q-matrix stably.

6. ACKNOWLEDGMENTS

This research was supported by JST, CREST, MEXT KAKENHI 25750095 and 23240019.

7. REFERENCES

[1] M.C. Desmarais et. al. Item to skills mapping: Deriving a conjunctive q-matrix from data. In *ITS2012*, pages 454–463.

[2] S. Oeda and K. Yamanishi. Extracting time-evolving latent skills from examination time series. In *EDM2013*, pages 340–341.

[3] S. Zhang et. al. Learning from incomplete ratings using non-negative matrix factorization. In *Society for Industrial and Applied Mathematics*, pages 548–552, 2006.

[4] K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20:345–354, 1983.

Generalizing and Extending a Predictive Model for Standardized Test Scores Based on Cognitive Tutor Interactions

Ambarish Joshi, Stephen E. Fancsali, Steven Ritter, Tristan Nixon, Susan R. Berman
Carnegie Learning, Inc.
437 Grant Street, Suite 918
Pittsburgh, PA 15219 USA
888.851.7094
{ajoshi, sfancsali, sritter, tnixon, sberman}@carnegielearning.com

ABSTRACT

Recent work demonstrates that process data from intelligent tutoring systems (ITSs) can be used to predict student outcomes on high-stakes, standardized tests. Such models are important if ITSs are to be used for formative assessment and as replacements for external assessments. Recent work used various measures of learning efficiency and performance from problem-level, aggregate data from Carnegie Learning's Cognitive Tutor to predict standardized test scores on the state of Virginia's Standards of Learning exam. We generalize this model to a different school district, state, and standardized test and examine extending the model using finer-grained data.

Keywords

Formative assessment, standardized tests, intelligent tutoring systems, Cognitive Tutor, off-task behavior, gaming the system

1. INTRODUCTION & BACKGROUND

Advanced learning systems like Cognitive Tutor (CT) [8] and ASSISTments [4], which assess students as they teach, have the potential to reduce time taken away from instruction to assess student knowledge. By fully integrating instruction with assessment, they ensure that the two are well aligned. Recent work has aimed to demonstrate correlations between such unconventional assessments and exams to determine whether they prepare students for assessments or could replace conventional high-stakes assessments (e.g., [5,7,9]).

The CT mathematics intelligent tutoring system (ITS) is known to improve student learning and performance on standardized tests (e.g., [6]), and recent work [9] demonstrated that a model incorporating CT process data predicts middle school outcomes on Virginia's (VA's) Standards of Learning (SOL) exam [10]. We generalize this result to a new population of students from a different school district and U.S. state on a different test. We also investigate extending the model by using finer-grained data.

We ask: what process data should instructional/analytics systems track, and what level of granularity (e.g., problem-level vs. problem-solving steps) is sufficient for tracking? Both questions are important if ITSs are to be used for instruction *and* formative assessment, and if instructor dashboards and diagnostic systems are to be useful and scalable.

1.1 Cognitive Tutor

CT curricula are sequences of topical sections in instructional units. Sections contain a set of problems, each of which targets one or more knowledge components (KCs). Problems are adaptively presented to students according to KCs that a student has yet to master, as probabilistically assessed by CT. Students proceed to the next section when they master all KCs in a section. A student can choose to ask for a hint at any problem-solving step. As assessing KC mastery depends on errors made and hints requested, students may require different numbers of problems.

1.2 Previous Work & VA's SOL Exam

Past work has associated learning system process data with standardized test scores. For example, data from ASSISTments (e.g., counts of help requests) have been used to predict Massachusetts Comprehensive Assessment System (MCAS) scores (e.g., [5]). Problem-level features used in these models do not require logging data at the level of individual student actions, but they still provide satisfactory models of test scores.

Predictions of sensor-free, data-driven "detectors" of gaming the system [2], off-task behavior [1], and affect [3] with ASSISTments data have been used to predict MCAS scores [7]. Detectors require fine-grained tracking of student actions in ITSs rather than features like aggregate counts of hints. A natural question is whether finer-grained data provide information about test scores beyond that provided by problem-level data.

Previous work [9] predicted outcomes of VA's SOL test from problem-level CT process data. Data included usage for 3,224 students in Grades 6-8 across 12 schools. Grade 7 data were used to build an ordinary least squares (OLS) linear regression model. Five variables were significant: (1) *Total Problem Time* - problem-solving time; (2) (number of) *Skills Encountered* (3) (number of) *Sections Encountered*; (4) *Assistance Per Problem* - average sum of # of hints requested and errors made for each problem; and (5) (average number of) *Sections Mastered Per Hour*. Model parameter estimates for Grade 7 data (model adjusted $R^2 = 0.43$) were used to predict outcomes for Grade 6 ($R^2 = 0.46$), Grade 8 ($R^2 = 0.18$), and overall ($R^2 = 0.38$).

2. GENERALIZING THE MODEL

2.1 West Virginia's WESTEST 2

We generalize the model that predicted SOL scores by modeling data from a school district in West Virginia (WV) that uses a different standardized test, WESTEST 2. For math, WESTEST 2 assesses a student's on defined standards, objectives, and skills, using multiple-choice questions and gridded response items [11].

2.2 Data & Results

We build models of data from the 2012-2013 school year, including usage information for 636 students, mostly 9th graders taking Algebra 1, with 5+ hours of CT usage and scores above the "novice" ranking for WESTEST 2 achievement descriptors, as students with a novice ranking are likely from a different learner sub-population than that which we target here.

Our approach starts with the variables found in the prior work. Skills *Encountered* and *Sections Encountered* are highly correlated (r = 0.989), so we disregard the variable with lower correlation to WESTEST 2 scores (*Sections Encountered*). Building a stepwise regression model from remaining variables, including neither *Skills Encountered* nor *Total Problem Time* improves the model over that of Table 1 ($R^2 = 0.295$). Student-level (10-fold) cross validation does not lead to models that differ substantially, so two variables generalize to WV's exam.

Table 1: Standardized regression coefficients, & significance for generalized model of WESTEST 2 scores (*p < .001)**

Variable	Coefficient
Assistance Per Problem	-0.225***
Sections Mastered Per Hour	0.372***

3. EXTENDING THE MODEL

Data-driven "detectors" of gaming the system [2] (e.g., abusing hints or excessive guessing), off-task behavior [1] and affect [3] have been used to predict MCAS scores [7]. Detectors use features "distilled" from problem-solving-step-level (i.e., finer-grained data) logs.

We construct *Steps Gamed* and *Steps Offtask* variables using data for 5 million+ student actions, to capture the proportion of student problem-solving steps detected as instances of these behaviors. Table 2 reports the regression model ($R^2 = 0.322$) including these variables and correlations to WESTEST 2 outcomes.

Table 2: Regression coefficients for extended model & correlations with learning outcome (p <.01; ***p<.001)**

Variable	Regression Coefficient	Correlation with WESTEST 2
Assistance Per Problem	-0.07	-0.47***
Sections Mastered Per Hour	0.396***	0.52***
Steps Gamed	-0.224***	-0.51***
Steps Offtask	0.129**	-0.2***

Measures of student efficiency, gaming the system, and off-task behavior are significant predictors of outcomes. Off-task behavior is relatively weakly correlated with WESTEST 2 outcomes. *Assistance Per Problem* and *Steps Gamed* are highly correlated (r = 0.8, two-tailed p < .001); if gaming is a common cause of more assistance (i.e., hint abuse) and less learning, conditional on *Steps Gamed*, *Assistance Per Problem* and learning would be independent, so *Assistance Per Problem* would be insignificant. *Sections Mastered Per Hour* is negatively correlated with *Steps Gamed* (r = -0.68, p < .001) and *Steps Offtask* (r = -0.59, p < .001), so gaming and off-task behavior seem to provide the same information about outcomes as variables in the generalized model.

4. DISCUSSION

Two features from a model that predicts VA's SOL test generalize to WV's WESTEST 2. We attempted to extend the model by including features that require finer-grained data about problem-solving steps; we find that gaming the system and off-task behavior do not substantially improve our predictions for WESTEST 2, in part because assistance is highly correlated with gaming. Our results thus suggest that the benefits of collecting fine-grained data needed to construct sophisticated features may not be substantial for use in test score prediction. Nevertheless, engineered features and fine-grained data may provide for real-time assessment to target interventions.

5. ACKNOWLEDGMENTS

Sujith M. Gowda and Ryan S.J.d. Baker provided both code for and assistance with detector models used in this work.

6. REFERENCES

[1] Baker, R.S.J.d. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proceedings of ACM CHI 2007: Computer-Human Interaction* (San Jose, CA, April 28 – May 3, 2007). ACM, New York, 1059-1068.

[2] Baker, R.S.J.d., de Carvalho, A. M. J. A. 2008. Labeling student behavior faster and more precisely with text replays. In *Proceedings of the 1st International Conference on Educational Data Mining* (Montreal, 2008). 38-47.

[3] Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G.W., Ocumpaugh, J., Rossi, L. 2012. Towards sensor-free affect detection in Cognitive Tutor Algebra. In *Proceedings of the 5th International Conference on Educational Data Mining* (Chania, Greece, 2012). 126-133.

[4] Feng, M., Heffernan, N.T., Koedinger, K.R. 2009. Addressing the assessment challenge in an intelligent tutoring system that tutors as it assesses. *User Model. User-Adap.* 19 (2009), 243-266.

[5] Feng, M., Heffernan, N.T., Koedinger, K.R. 2006. Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (Jhongli, Taiwan, 2006). 31-40.

[6] Pane, J., Griffin, B. A., McCaffrey, D. F., Karam, R. 2014. Effectiveness of Cognitive Tutor Algebra I at scale. *Educ Eval Policy An* 36 (2014), 127-144.

[7] Pardos, Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda, S.M., and Gowda, S.M. 2013. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Proceedings of the Third International Learning Analytics and Knowledge Conference* (Leuven, Belgium, 2013). ACM, New York, NY, 117-124. DOI= http://doi.acm.org/10.1145/2460296.2460320

[8] Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.T. 2007. Cognitive Tutor: applied research in mathematics education. *Psychon. B. Rev.* 14 (2007), 249-255.

[9] Ritter, S., Joshi, A., Fancsali, S.E., Nixon, T. 2013. Predicting standardized test scores from Cognitive Tutor interactions. In *Proceedings of the 6th International Conference on Educational Data Mining* (Memphis, TN, 2013). 169-176.

[10] Virginia Department of Education. 2014. Standards of Learning (SOL) & Testing. Retrieved February 23, 2014. http://www.doe.virginia.gov/testing/

[11] West Virginia Department of Education, Office of Assessment and Accountability. 2014. WESTEST 2 Overview. Retrieved February 23, 2014. http://wvde.state.wv.us/oaa/westest_index.html

How patterns in source codes of students can help in detection of their programming skills?

Štefan Pero
Institute of Computer Science, Faculty of Science, Pavol Jozef Šafárik University
Košice, Slovakia
stefan.pero@student.upjs.sk

Tomáš Horváth
Institute of Computer Science, Faculty of Science, Pavol Jozef Šafárik University
Košice, Slovakia
tomas.horvath@upjs.sk

ABSTRACT

A technique to detect patterns in student's program source codes. First, we represent a source code in the form of an Abstract Syntax Tree (AST). The detection of patterns is done with the SLEUTH algorithm for frequent subgraph mining on trees. We provide experiments using real data from a programming course at our university. In the paper, we discuss the relation between patterns and skills as well as some use cases and further directions of our research.

Keywords

pattern, source code, student, skills

1. INTRODUCTION

One of the best-known problems in educational data mining (EDM) is predicting student's performance [6]. A great deal of algorithms have been applied to predict academic success of students. However, we are interested mainly in the following issues/questions: What lies in the background and prerequisities of students' success? What skills do students have and what is their level? We are inspired by a real situation from one programming course at our university. Students solve programming tasks where their produce a program source code. Evaluation of the solutions for the tasks solved by students is a complex process driven mainly by subjective evaluation criteria of a given teacher. Each teacher is somehow biased meaning how strict she is in assessing grades to solutions. Besides the teacher's bias there are also some other factors contributing to grading, for example, teachers can make mistakes, the grading scale is too rough-grained or too fine grained, etc. Latent programming skills of students are somehow "encoded" in their source codes provided. Automatic detection of these latent skills (with or without the assistance of the teacher) remains still an open issue.

Pardos and Heffernan presented a model called "Knowledge tracing" [5] and they used it to model students' knowledge and learning over the time assuming that all students share the same initial prior knowledge. However, by considering the needed (listed) skills as attributes of the task, it is straightforward to use them also as features in prediction models [7]. Desmarais [3] introduces different linear models of student skills for small, static student test data that does not contain missing values. They compare the predictive performance of their model to the traditional psychometric Item Response Theory approach, and the k-nearest-neighbors approach. In [1] they present wrapper-based method for finding the number of latent skills.

This work focuses on designing the technique to detect programming patterns from students' source program codes. Using real data, we illustrate how do patterns related to skills of students predefined by the teacher (author) are discovered. The contributions of this work are the following: i) we introduce a model for representing source codes in a tree-structure, ii) we propose an approach to detect patterns in source codes utilizing pattern mining algorithm.

2. THE PROPOSED APPROACH

Our approach is based on pattern detection from source codes and on the analysis of the relationships between the found patterns and the skills for programming tasks predefined by the teacher.

Source Code Representation is a critical issue in designing the process of pattern recognition. We utilize a representation scheme of source codes in the form of Abstract Syntax Trees which provide detailed information about the source code which can be used for various types of analysis [4]. Since AST contains lots of abundant information from a pattern detection point of view, we have implemented our own filter to generate a representation of a source code in XML format from AST. which provides us with a better abstraction of a source code in different levels which allows us to better specify its important parts needed for the next step of our approach.

Pattern Mining Method which we use in our approach is SLEUTH, an efficient algorithm for mining frequent, unordered, embedded subtrees in a database of labeled trees [8]. Given the particular source codes, first, we represent them in relevant trees in XML format at the given level of abstraction. Second, we apply SLEUTH on the prepared dataset of trees.The aim is to find all patterns (frequent, unordered, embedded subtrees) in the input dataset. We are especially interested in so-called *maximal frequent patterns*, i.e. maximal frequent subtrees which are defined as those frequent subtrees none of which proper supertrees are frequent [2]. Finally, we cluster the resulting maximal frequent patterns (since many of them may be similar) and extract a set of representative patterns from each cluster of maximal frequent patterns.

Relation between Patterns and Skills. Consider the following instance of the *for loop* construction in the Java programming language.

$$\texttt{for(int i=0;i<5;i++) \{...\}} \tag{1}$$

To understand this construction of a for loop, and thus, to be able to use it during programming, we must first understand the following four programming concepts we call *prerequisites* for the for loop : i) variable declaration (`int i`), ii) variable assigment (`i=0`); iii) relational operators (`i<5`), iv) increment/decrement operators (`i++`). An important issue to mention is "The whole is greater than the sum of its parts" principle. For example, if one knows all of these prerequisites for the for loop individually it does not necessarily mean that she is also able to construct a for loop itself.

3. FIRST EXPERIMENTS

Experiments were performed with a real-world dataset, labeled "PAC"[1]. The dataset contains the following information about students' solutions: *studentID, taskID, teacherID, grade, review, solution (source code)*. Main characteristics of the dataset are described in the table 1. Each task belong to one set of tasks, i.e. we consider a set of tasks as one complex task containing several subtasks. We realized experiments on the sets of tasks according to the described steps of our approach above, such as representation in AST, conversion to XML, pattern mining with SLEUTH and clustering the maximal patterns.

Table 1: Characteristics of the dataset used

Dataset PAC	#Students	#Sets	#Tasks	#Codes
2011/2012 A	82	7	33	578
2011/2012 B	36	9	21	381
2012/2013 A	85	6	28	769
2012/2013 B	33	10	20	397
2013/2014 A	78	7	31	510

The result shown in Figure 1 refer to the number of patterns and the number of maximal patterns detected in the data for different sets of tasks. Using maximal patterns we are able to filter out repetitive and meaningless patterns. In pattern mining method we used support 0.8, 0.9, 1 corresponding to 80%, 90% and 100% coverage, respectively.

4. CONCLUSIONS

We presented a model for mining patterns in source codes in order to map these patterns to corresponding programming skills. The proposed model consists of several phases such as source code representation in the form of AST and its transformation to XML at different levels, mining frequent maximal patterns and choose their representatives by utilizing clustering techniques. Since our work is in its beginning we provided only some early-bird experiments. We have also discussed three use cases of the mined programming patterns we would like to focus on in our future work.

[1] Collected from the "Programming Algorithms Complexity" course at the Institute of Computer Science at Pavol Jozef Šafárik University during the years 2011–2014.

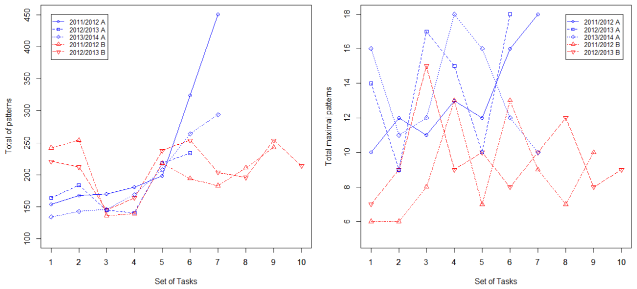


Figure 1: The number of detected patterns and maximal patterns with support=1 (i.e. 100% coverage).

5. ACKNOWLEDGMENTS

This publication is the result of the Project implementation: **University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology**, ITMS: 26220220182, supported by the Research & Development Operational Programme funded by the ERDF and partially supported by the research grants *VEGA 1/0475/14* and *VVGS-PF-2013-102*.

6. REFERENCES

[1] B. Beheshti, M. C. Desmarais, and R. Naceur. Methods to find the number of latent skills. *International Educational Data Mining Society*, 2012.

[2] Y. Chi, R. Muntz, S. Nijssen, and J. Kok. Frequent subtree mining - an overview. 2005.

[3] M. C. Desmarais, R. Naceur, and B. Beheshti. Linear models of student skills for static data. In *UMAP Workshops*, 2012.

[4] M. K. and S. Yamamoto. A case tool platform using an xml representation of java source code. *Proccedings of Fourth IEEE International Workshop on Source Code Analysis and Manipulation*, 2004.

[5] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. *Proceedings of International Conference on User Modeling, Adaptation and Personalization, (UMAP 2010)*, 2010.

[6] C. Romero and S. Ventura. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2012.

[7] N. Thai-Nghe, L. Drumond, T. Horváth, and L. Schmidt-Thieme. Multi-relational factorization models for predicting student performance. *ACM SIGKDD 2011 Workshop on Knowledge Discovery in Educational Data*, 2009.

[8] M. J. Zaki. Efficiently mining frequent embedded unordered trees. *Fundamenta Informaticae*, 2005.

A Preliminary Investigation of Learner Characteristics for Unsupervised Dialogue Act Classification

Aysu Ezen-Can
Department of Computer Science
North Carolina State University
aezen@ncsu.edu

Kristy Elizabeth Boyer
Department of Computer Science
North Carolina State University
keboyer@ncsu.edu

ABSTRACT

For tutorial dialogue systems, classifying the dialogue act (such as questions, requests for feedback, or statements) of student natural language utterances is a central challenge. Recently, momentum is building for the use of unsupervised machine learning approaches to address this problem because they reduce the manual tagging required to build dialogue act models from corpora. However, unsupervised models still do not perform as well as supervised models in terms of accuracy. This paper presents an unsupervised dialogue act modeling approach that leverages the influence of learner characteristics, particularly students' perceptions of their own skill, on their language use. The experimental findings show that leveraging skill perception within dialogue act classification improves performance of the models, producing better accuracy. This line of investigation will inform the design of next-generation tutorial dialogue systems, which leverage machine-learned models to adapt to their users.

Keywords

Tutorial dialogue, learner characteristics, dialogue act classification, unsupervised machine learning.

1. INTRODUCTION

Tutorial dialogue is a highly effective form of instruction, and much of its benefit is thought to be gained from the rich natural language dialogue exchanged between tutor and student [2]. In order to model tutorial dialogue for the purposes of building tutorial systems or for studying human tutoring, *dialogue acts* provide a valuable level of representation. Dialogue acts represent the underlying intention of utterances (for example, to ask a question, agree or disagree, or to give a command) [1]. For tutorial dialogue systems, dialogue act classification is crucial to understanding students' utterances and developing tutorial strategies [6].

Today's tutorial dialogue systems utilize a variety of dialogue act classification strategies. Historically when machine learning has been used to devise tutorial dialogue classifiers, these have been *supervised* classifiers, which require training on a manually labeled corpus. However, supervised techniques face substantial limitations in that they are labor-intensive due to the manual annotation and handcrafted dialogue act taxonomies that are usually domain-specific. To overcome these challenges, unsupervised dialogue act modeling techniques have been investigated in recent years.

Despite this growing focus on developing unsupervised dialogue act classifiers, these models still underperform compared to supervised approaches in their accuracy for classifying according to manual tags. However, while unsupervised models to date have considered such things as lexical features (the words found in the utterance) and syntactic features (the structure of the sentence), they have not considered learner characteristics, such as skill perception, which are believed to influence the structure of tutorial dialogue [3]. Learner characteristics also play an influential role in learning in web-based courses [5].

This paper investigates whether the performance of an unsupervised dialogue act classifier can be improved by taking a specific learner characteristic into account. We utilize *skill perception*, a student's ranking of her own skill as she perceives it compared to others. Specifically, we train unsupervised dialogue act models that are tailored to students of a specific skill perception level, and we compare those models to ones trained without restricting by that learner characteristic. This unsupervised training is conducted entirely without the use of manual tags. We then test the models on held-out test sets within leave-one-student-out cross validation, and compare the resulting classification accuracy according to their previously applied manual tags. The results can inform the way that next-generation tutorial dialogue systems conduct their real-time dialogue act classification.

2. DIALOGUE ACT MODELING

The corpus used in this study consists of computer-mediated student-tutor interactions during an introductory computer science programming task [4]. Throughout the data collection, students and tutors communicated through a textual dialogue-based learning environment while working on Java programming. Students were given a pre-survey that included items on computer science. The pre-survey included an item that asked students to rate how skilled they are in the domain compared to others. We refer to this response as skill perception. Students (n=42) were divided into groups (high and low skill perception) based on the median score.

The corpus containing 1,640 student utterances was manually annotated with dialogue act tags in a previous work [4]. There are seven student dialogue acts in total (*Answer, Acknowledgement, Statement, Question, Request for Feedback, Clarification and Other*) where the majority class baseline chance is 39.95%. As required by unsupervised modeling, these dialogue act tags are not available during model training, but we use them for evaluation purposes to calculate accuracy on a held-out testing set.

We hypothesize that dialogue act models built using unsupervised machine learning will perform substantially better when customized to specific learner group skill perception. The corpus is partitioned by skill perception and we examine whether an unsupervised dialogue act classifier trained only on students with high skill perception performs better on a test set of dialogue acts from high skill perception students, compared to a classifier trained on a mixture of high and low skill perception students.

In order to gather accuracy data across these characteristics, we conduct leave-one-student-out training and testing folds. The testing set for each of the *n* folds consists of all of a single student's dialogue utterances and the model is trained on the remaining *n-1* students. We compute the average test set performance of the model across all folds for each learner characteristic partition. The performance metric utilized in this study is *accuracy* compared to the manually labeled dialogue acts where accuracy is the number of utterances in the test set that were classified the same as their manual label, divided by the total number of utterances in the test set. Our unsupervised dialogue act classifier leverages the *k*-medoids clustering technique.

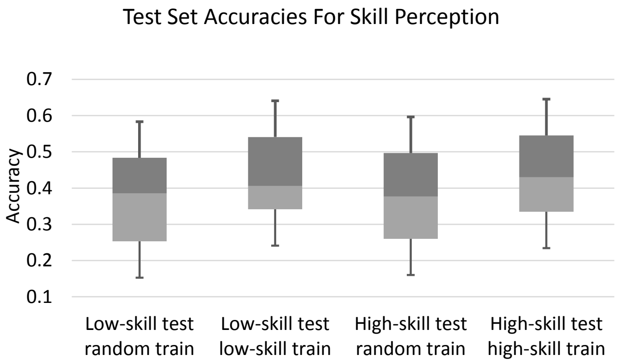


Figure 1: Leave-one-student-out test set accuracies for models by skill perception

For students with low skill perception ($n_{lowSkill}$=26) the average performance of the dialogue act classification model trained on utterances of randomly selected students is 0.39 (σ=0.17) whereas the accuracy rises to 0.43 (σ=0.17) for a tailored model trained only on students with low skill perception (Figure 1). This is not a statistically significant difference after Bonferroni correction. For these students, 11 out of 26 cases improved their performance by utilizing the learner characteristic (five of them above 5% and five of them above 15%), six of them were affected negatively (four of them below 5% decrease) and nine of them achieved the same performance.

The same pattern is visible for students with high skill perception ($n_{highSkill}$=16). For these students, the average test set accuracy increases from 0.38 (σ=0.14) to 0.42 (σ=0.13) gained by using utterances of students with high skill perception rather than learning from utterances selected randomly, again not statistically significant after Bonferroni correction. Six of the cases out of sixteen improve test set accuracy (four of them above 15%), three of them degrades (two of them below 5%) and seven cases perform equally.

Although the differences in model performance were not statistically reliable for students in different skill perception groups, we observed some interesting patterns within these groups (Table 1). Students with low skill perception tended to use more utterances such as, "ok I am getting it," which may be a type of affective or face-saving dialogue move. Students in the high skill perception group seem to exhibit more social, relaxed utterances, reflected by examples such as, "cool cool" and "yeah haha."

	Low Skill Perception	High Skill Perception
Acknowledgments	- oh - ok I am getting it - ok I get it! - interesting - oh ok	-cool cool -yeah haha - yep lol -yep! exciting stuff! - sure
Questions	- what do i do now - can you explain more about the scanner line - so what is this doing exactly? -why is not it prompting me to enter my name?	-comments are just a way to write notes to others to help them understand right? - out of curiosity would not it make sense to switch those last two lines of code?

Table 1: Selected utterances from clusters tailored to skill perception

3. CONCLUSION

Understanding student natural language within intelligent tutoring systems is a critical line of investigation for tutorial dialogue systems researchers. For dialogue act classification in particular, the field has only begun to explore unsupervised approaches and to investigate the range of features that are beneficial within this paradigm. We have presented a first attempt to leverage learners' perception of their own skill within a dialogue act classification model. It is hoped that the research community can continue to build richer models of natural language understanding for students of all learner characteristics in order to enhance learning.

ACKNOWLEDGMENTS

Thanks to the members of the JavaTutor project and the LearnDialogue group at NC State University. This work is supported in part by the National Science Foundation through Grant DRL-1007962 and the STARS Alliance, CNS-1042468. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

REFERENCES

[1] Austin, J.L. 1962. *How To Do Things With Words*. Oxford University Press.

[2] Bloom, B.S. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-one Tutoring. *Educational Researcher*. 4–16.

[3] Boyer, K.E., Vouk, M.A. and Lester, J.C. 2007. The Influence of Learner Characteristics on Task-Oriented Tutorial Dialogue. *Proceedings of AIED*, 365–372.

[4] Ha, E.Y., Grafsgaard, J.F., Mitchell, C.M., Boyer, K.E. and Lester, J.C. 2012. Combining Verbal and Nonverbal Features to Overcome the "Information Gap" in Task-Oriented Dialogue. *Proceedings of thel SIGDIAL Meeting on Discourse and Dialogue,* 247–256.

[5] Hershkovitz, A. and Nachmias, R. 2011. Online Persistence In Higher Education Web-Supported Courses. *The Internet and Higher Education*. 14, 2.,98–106.

[6] Marineau, J., Wiemer-Hastings, P., Harter, D., Olde, B., Chipman, P., Karnavat, A., Pomeroy, V., Rajan, S. and Graesser, A. 2000. Classification of Speech Acts in Tutorial Dialog. *Proceedings of the Workshop On Modeling Human Teaching Tactics And Strategies at ITS*, 65–71.

Improving Retention Performance Prediction with Prerequisite Skill Features

Xiaolu Xiong
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA, USA
xxiong@wpi.edu

Seth A. Adjei
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA, USA
saadjei@wpi.edu

Neil T. Heffernan
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA, USA
nth@wpi.edu

ABSTRACT

This paper describes our experiment and analysis of utilizing prerequisite skill features to improve the predicting of student retention performance. There are two aspects that make this paper interesting. First, instead of focusing on short-team performance, we investigated the student retention performance after a delay of 7 days. We explored several prerequisite skill features that can be captured in an intelligent tutoring system; in our particular case, these prerequisite skill features were acquired from Common Core standard skills and student data while working on these skills. We showed that some of these features have encouraging predictive power. Our analysis confirmed the value of prerequisite skill features in predicting retention performance, the prediction results showed an improvement from an R^2 of 0.182 with a baseline feature set to an R^2 value of 0.192.

Keywords

Educational data mining, feature selection, knowledge retention, intelligent tutoring system

1. INTRODUCTION

Inspired by the notion of robust learning [1] and the design of the enhanced ITS mastery cycle proposed by Wang and Beck [3], we developed a system called the Automatic Reassessment and Relearning System (ARRS) to make decisions about when to review skills that the student have mastered in the ASSISTments system (www.assistments.org). One of the important compounds of ASSISTments is the mastery learning problem set, which simplifies the notion of skill mastery to three consecutive correct responses with the number of attempted problems before students achieve mastery. The current workflow of ARRS is relatively simple: after classroom teaching of a certain skill, teachers use ASSISTments to assign a mastery learning problem set of that skill to students, and students are required to first master the skill by completing the Mastery learning problem set; ARRS will then automatically reassess students on the same skill 7 days later with a retention test (also called the reassessment test in ASSISTments) built from the same sets of problems the student already mastered. If students answer the problem correctly, we treat them as if they are still retaining this skill, and ARRS will test them 14 days later, 28 days later, and then finally 56 days after that. If a student fails the retention test, ARRS will give him an opportunity to relearn the skill.

Cognitive domains usually have a model that represents the relationship between knowledge components. Each of these knowledge components is a major skill in the domain that students are expected to have. The relationship between these knowledge components or skills is either prerequisite or post-requisite. A prerequisite skill of a skill A is a skill that students are expected to have to be able to succeed in assessments of requiring skill A. Without knowledge of the prerequisite skill(s) of a given skill, a student is not expected to respond correctly to questions from that given skill. The map in Figure 1 is representation of a subset of the prerequisite skill model used by a number of features in ASSISTments. The ovals represent the skills and the arrows linking the ovals show the prerequisite and post-requisite relationships between the skills. The codes are the Massachusetts Common Core State standards for the Math skills [2]. ASSISTments started adopting the Common Core standards since fall 2013.

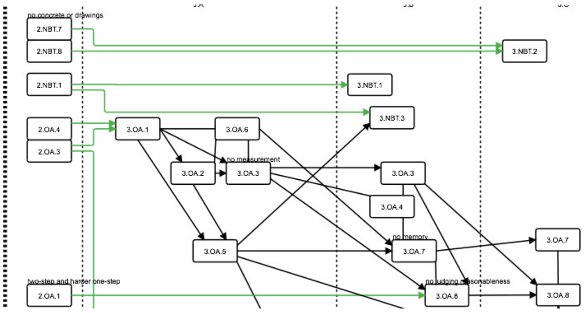


Figure 1. A subset of the Common Core skills

Cognitive models, together with their skills maps, have been used to determine students' cognitive levels in a given domain. For example, when a student answers a problem from a given skill incorrectly, problems are presented from the prerequisite skill to determine how well they know the prerequisite skills.

2. MODELING PREREQUISITE SKILL EFFECTS

Consider a situation where a student has very high performance in general but performed poorly in prerequisite skills to a particular skill. When this student encounters the postrequisite skill, we would not expect him to have robust mastery; therefore, his performance on retention tests to that postrequisite skill could be poor. However, most models have only focused student's general performance on their most recent performance. Hence we formed a hypothesis that the prerequisite skill performance can be independent from student local performance and can be used to enhance our models of predicting retention performance. We initially noticed [4] that the number of problems required to achieve mastery has great influence on the delayed performance. We refer to this number as the Mastery Speed. We first employed the mastery speed, as well as two other basic features to establish a baseline for our modeling work. These features relate to item and skill information, including: (1) *problem easiness* and (2) *skill ID*. Note that because we are not using the identifier of students in the modelling work, thus our models can test our ability of generalizing to new students. To test our hypothesis, the next step was to gather a set of prerequisite skill features and identify which features can be used as predictors. Towards this end, we selected the following three features to capture different prerequisite skill information:

(1) *prerequisite skill ID*: the unique identifier of each prerequisite skill. By modeling skill ID as a factor, we are estimating an overall effect of these skills;

(2) *student prerequisite skill performance*: this is a measure of a student's performance on a direct prerequisite skill of the retention test skill. This number is presented by the percentage of correctness of all the problems that are answered by the students for this prerequisite skill;

(3) *prerequisite skill easiness*: the percentage of correctness for this prerequisite skill across all answers and all students.

We experimented with using *prerequisite skill ID* as a factor, as well as *student prerequisite skill performance* and *prerequisite skill easiness* as covariates; hence there are three models to be calculated besides the baseline model. Table 1 provides the results for each of these models, the prediction performance were measured in terms of R^2 on the testing set.

Table 1. Prerequisite skill model performance

Model	R^2
base model + *student prerequisite skill performance*	0.189
base model + *prerequisite skill ID*	0.185
base model + *prerequisite skill easiness*	0.182
base model	0.182

From the results in Table 1, we can see that improved models were obtained both on *prerequisite skill ID* and *student prerequisite skill performance*. The results from using student prerequisite skill performance clearly indicate that a student's performance on prerequisite skills is helpful for improving predictions. The predictive power of *prerequisite skill ID* may suggest that there seems to be an overall skill effect, which is different from the average performance of prerequisite skills, which is modeled by prerequisite skill easiness. Furthermore, a model using both *prerequisite skill ID* and *student prerequisite skill performance* achieved an R^2 value of 0.192 and the result is statistically reliable ($p \approx 4.5 \times 10^{-4}$). This led us to believe that these two features are largely independent predictors and whatever *prerequisite skill ID* represents, it is relatively distinct from *student prerequisite skill performance* as the R^2 increases noticeably when both are modeled. The Beta coefficient values and p-values for each covariate are shown in Table 2.

Table 2. Parameter table of covariates

Covariate	Beta	p-value
problem easiness	6.306	.00
prerequisite skill performance	2.24	.00

The positive Beta values indicate that the larger the covariate is, the more likely the student responded to this problem correctly. So we see that the easiness of retention test problem is still more likely to affect students' performance compared to their *prerequisite skill performance*.

3. CONCLUSIONS AND FUTURE WORK

In this work we attempted to model prerequisite skill features to better predict student retention performance in an intelligent tutoring system on a small dataset. We need to further investigate our model with larger datasets and other data sources.

In this paper we only investigated the direct prerequisite skill of test skills. We have not yet looked into the skill system as a hierarchy of complete knowledge components. For future work we will consider the notion of the student's performance in all prerequisite skills prior to the skills we are investigating. For example, we could measure how well a student did on the retention tests of prerequisite skills. Also, it is possible that skill interference is also affecting the retention performance. Exploring these avenues to discover prerequisite skill impacts on performance is an interesting future direction.

4. ACKNOWLEDGMENTS

We want to acknowledge the funding on NSF grant DRL-1109483 as well as funding of ASSISTments. See here (http://www.webcitation.org/67MTL3EIs) for the funding sources for ASSISTments.

5. REFERENCES

[1] Baker, R. S., Gowda, S. M., Corbett, A. T., & Ocumpaugh, J. 2012. Towards automatically detecting whether student learning is shallow. In Intelligent Tutoring Systems (pp. 444-453). Springer Berlin Heidelberg.

[2] Massachusetts Common Core State standards for Mathematics. http://www.corestandards.org/Math Accessed: February 28, 2014

[3] Wang, Y., Beck, J.E. 2012. Incorporating Factors Influencing Knowledge Retention into a Student Model, In *Proceedings of the 5th International Conference on Educational Data Mining*, 201-203

[4] Xiong, X., Li, S., Beck, J. 2013.Will you get it right next week: Predict delayed performance in enhanced ITS mastery cycle. In *Proceedings of the 26th International FLAIRS Conference*

Indicator Visualisation for Adaptive Exploratory Learning Environments

Sergio Gutierrez-Santos, Manolis Mavrikis, Alex Poulovassilis, Zheng Zhu
London Knowledge Lab, University of London, UK
{sergut,m.mavrikis,ap,zheng}@dcs.bbk.ac.uk

ABSTRACT

This paper presents our approach to identifying areas of improvement in the intelligent components of adaptive Exploratory Learning Environments. Students' interaction data from an online operational database are first transformed into a data warehouse in order to allow visualisation and exploration using online analytical processing (OLAP) tools. Using a microworld for secondary school algebra as a case study, we also present some more targeted visualisations of the students' interaction data. We demonstrate the possibilities that these visualisations provide for exploratory data analysis, enabling confirmation or contradiction of expectations that pedagogical experts may have about the system and ultimately providing both empirical evidence and insights for its further development.

Keywords

exploratory learning environments, indicators, visualisation

1. INTRODUCTION

In recent years there has been much research and development work focusing on open-ended interactive educational applications that encourage students' experimentation within a domain. These applications range from simple games to complex simulators and microworlds [1]. Although not new, they are becoming more common due to the new forms of interaction afforded by tablets and increasing ease of creation through related authoring tools. In parallel, the appreciation that in order for students to benefit from interaction with such Exploratory Learning Environments (ELEs) there is a need for explicit pedagogical support [2] has led to the development of adaptive support components [1].

The design and improvement of such adaptive exploratory environments is not a trivial task. Following a principled, evidence-based approach needs to rely on data gathered from students' interactions, which can help educationalists to understand how students are interacting with the system and technical experts to prioritise the development of enhanced or new support features. However, log files from ELEs contain large quantities of data that render their interpretation for researchers, teachers and systems designers quite a difficult and expensive task (cf. [3]). In addition, one does not always know in advance what data are required for analytical purposes and therefore an exploratory analysis may be needed. Lastly, logging of students' interaction data typically takes place in a manner that is optimal for recording and supporting students' interaction but not necessarily for subsequent analysis and decision-making.

In this paper, our case study is the MiGen system, which provides an intelligent environment to support 11-14 year old students' learning of algebra concepts In MiGen, students undertake tasks in a microworld called eXpresser. These tasks ask students to create models consisting of 2-dimensional tiled and coloured patterns — firstly specific instances of such models and then generalised versions in which one or more of the numbers in their construction are replaced by so-called "unlocked" numbers (i.e. variables). In parallel, students are asked to create rules specifying the number of tiles of each colour that are needed to fully colour their models (for more details see www.migen.org).

As students are interacting with the system, MiGen's intelligent support component [1] applies rule-based and case-based reasoning techniques to infer the occurrence of a wide range of significant task-independent and task-dependent indicators from the students' actions. These inferrences are used to provide both unsolicited and on-demand feedback to students. In addition, the indicators are stored in the operational online MiGen database, leading to large volumes of such data.

The question we address in this paper is: how might this data be visualised and explored in order to determine the effectiveness of the intelligent support provided by the system and to improve it? We have investigated several possible visualisations, including (i) multi-dimensional data visualisation and exploration using online analytical processing (OLAP) tools, and (ii) more targeted visualisations of the frequency of occurrence of different types of indicators and the transitions between them.

2. INDICATOR VISUALISATION

We first transformed data from the online MiGen database into a data warehouse that categorizes indicator occurrences according to several dimensions (e.g. when in occurred, the student and task it relates to, what kind of indicator it is). Multi-dimensional visualisation of this warehouse data using standard OLAP tools allowed the MiGen team and other experts to see what kinds of positive, neutral and negative behaviours are occurring as students are undertaking a task.

2.1 Frequency of indicator type occurrences

The visualisation in Figure 1 illustrates the conditional relative frequencies of different types of indicators (indicators of Status -1, 0, 1, 2) in three successive classroom sessions (Sessions 1, 2, 3). The widths of the bars correspond to the relative frequencies of indicator occurrences between the sessions. We can see that the number of indicator occurrences grows with each successive session and that the frequency of occurrence of negative indicators is decreasing with each successive session. This may be because students are becoming more familiar with using the system — a hypothesis that could warrant further investigation.

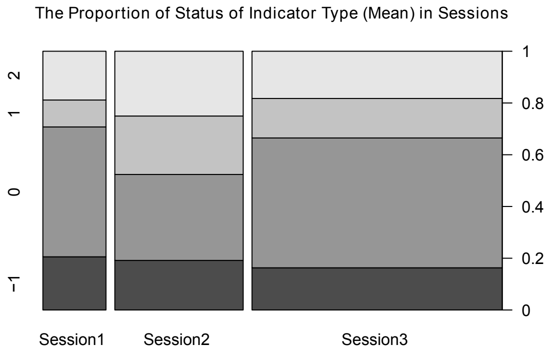


Figure 1: Per session proportion of negative (-1), neutral (0), positive (1) or feedback (2) indicators.

2.2 Transition of indicator type occurrences

Sequences of indicator types may be presenting patterns that can provide insight. Standard sequence analysis, however, provides patterns that are difficult to inspect. In order to facilitate the involvement of domain experts, we therefore investigated transition matrices, which are used to describe the transitions of a Markov chain.

Given a finite space of indicator types, $P_{ij} = P(j|i)$ is the probability of moving from indicator i to indicator j in one time step. Transition matrices can be normalised to quantify the transition probability from indicator i to any other indicator. We can also normalise the matrix to measure the incoming transition probability to indicator j from other indicators. In addition, we add artificial points to the system to capture the start and end of the interactions. Accordingly, for each model, s indicates the first indicator before the student begins construction of the model and e the last indicator at the end of the model's construction.

Transition matrices can be visualised using graphs such as those in Figure 2. Indicators shown with a circle round them indicate that there are transitions in the data where this indicator occurs in succession. The thickness of each line or circle indicates the value of the transition probability: the thicker the line, the higher the probability. The red (light grey) lines are associated with a probability less than 0.2 and the black lines a probability greater than or equal to 0.2.

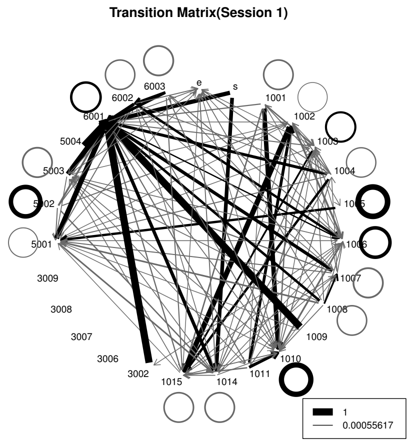


Figure 2: Incoming Transition Matrix (Session 1)

Figure 2 shows an example transition matrix from Session 1 that leads to interesting insights. For example, consider the transition from indicator 3002, corresponding to numerical answer being provided by the student, to indicator 6001, corresponding to an intervention being generated by the system. In the visualisation of Session 2 (not shown here) there is no occurrence of this transition. This indicates that feedback received from the system in Session 1 was carried over to students' interactions in Session 2. Such an observation helps us raise a hypothesis for more detailed analysis or subsequent experimentation (e.g. "are students internalising the system's feedback and thus avoiding the same error in subsequent sessions or is this simply an artifact of their increasing familiarity with the system?").

3. CONCLUSIONS

We have developed several visualisations of learners' interaction data from an exploratory environment. We have discussed some insights derived from these and how they can inform decisions with respect to further research and design of the intelligent support provided by the system. Currently, our visualisations require the support of a technical expert in order to create them, using either standard OLAP tools or ad-hoc visualisations (mostly generated using R scripts). We plan to improve both their interactivity and their ease of use, in order to allow stakeholders with less technical expertise to be able to create such visualisations for themselves, to explore the data from their perspective, and to derive hypotheses worth further investigation.

Acknowledgements. The MiGen project (migen.org) and the data collection was funded from the ESRC-EPSRC TEL Programme. The work here has been co-funded by the FP7 EU projects iTalk2Learn (#318051 italk2learn.eu) and M C Squared (#610467 mc2-project.eu)

4. REFERENCES

[1] S. Gutierrez-Santos, M. Mavrikis, and G. D. Magoulas. A Separation of Concerns for Engineering Intelligent Support for Exploratory Learning Environments. *Journal of Research and Practice in Information Technology*, 44, 2012.

[2] R. E. Mayer. Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. *American Psychologist*, 59(1):14–19, 2004.

[3] C. Romero, P. Gonzalez, S. Ventura, M. Del-Jesus, and F. Herrera. Evolutionary Algorithms for subgroup

Learning Aid Use Patterns and Their Impact on Exam Performance in Online Developmental Mathematics

Nicole Forsgren Velasquez
Utah State University
nicolefv@usu.edu

Ilya Goldin
CDDAAL, Pearson
ilya.goldin@pearson.com

Taylor Martin
Utah State University
taylormartin@usu.edu

Jason Maughan
Utah State University
jsnmaughan@gmail.com

ABSTRACT

Developmental mathematics is a college course aimed to remediate areas missed in high school mathematics. These courses are often offered online, which offers new opportunities to deliver content and learning aids to students. We utilize cluster analysis to identify learning aid use patterns, and then investigate their correlation to subsequent exam performance.

Keywords

Learning aids, online learning, cluster analysis

1. INTRODUCTION

Digital learning aids are often available to university students [7], and are used by novices as they solve novel problems and learn new material [5]. Indeed, most textbook publishers offer online or digital course content to instructors. We focus our research on digital learning aids because of their growing use, and research suggests that individuals who are engaged in learning seek information, such as learning aids, from online sources [2].

Digital learning aids can come in various formats, including text, videos, animations, and solution guides. Prior studies have shown that individuals use various learning aids to increase learning [e.g., 3], but it is still unknown what *combination* of learning aids impact student learning. That is, when several types of learning aids are available, do students display patterns of preferred learning aid use? To address this, we investigate the combinations of learning aids used by students, and the correlations of these combinations with student exam performance.

2. BACKGROUND

There is a great deal of prior work on how various learning aids may affect learning. For instance, it may be faster to study worked examples than to solve problems on the same skill, and studying worked examples enables faster subsequent skill application [8]. However, what is needed is a learning-analytic approach that examines the use of learning aids *in vivo*. Moreover, real-world use of learning aids reflects student preferences, habits and beliefs, distinct from so-called learning styles [6]. Because student preferences may differ and these preferences may change over time, this calls for a model of how students use learning aids. Based on sourcing theory [3], we investigate the learning aid use patterns seen in an online learning environment, and examine the correlations these patterns have on subsequent exam performance.

1. What learning aid use patterns are seen among college students in a developmental math course using online resources?

2. Do these learning aid use patterns correlate with subsequent exam performance?

3. METHODS

We used existing log data on a single course of 160 students. The subject was Developmental Mathematics, an introductory course for students who enrolled in college but lacked prerequisites for further study. We did not separate students into conditions. Demographic information was not available. No compensation was associated with the study. Instructor and students participated with the course management site as normal.

The online course management site gave instructors the ability to select assignments for their students. We captured learning aids available in our study context: animation, calculator, sample problem, textbook, and video. Each learning aid was not available for all homework problems, so the percentage of use was reported (calculated as number of learning aids used divided by number of learning aids available). These and other variables are described.

Animation: short animations of movement and graphics.

Calculator: provided through the web interface; included even though a student could use their own calculator.

Sample problem: worked example for the problem at hand with values that are different from the current problem. Each step is demonstrated and explained.

Textbook: content from the corresponding section in the textbook.

Video: distinguished from animations in that they are longer, include more explanation, and include audio.

Exam performance: performance that follows the use of learning aids; calculated as the number of problems answered correctly divided by the number of problems attempted. We do not include unanswered exam questions, because a student may know how to correctly answer questions they never see or questions they skip (e.g., by employing a test-taking strategy wherein they skip problems with the intention of returning to them later).

IRT difficulty: average for all problems on the exam across all occurrences of a problem (i.e., across all courses).

Exam number: sequence number of each exam.

Pretest Performance: the student's score on questions from the first exam covering whole numbers.

4. ANALYSIS AND RESULTS

What learning aid use patterns are seen among college students taking a developmental math course using online resources?

To investigate learning aid use combinations among college developmental math students, we develop a classification with cluster analysis [4]. We conduct our cluster analysis in three steps. First, we compile the learning aids used by each student between exams. Because we are interested in learning aid combinations regardless of when the combination appears, we look at all time periods between exams for all students. For example, if one student takes four exams, we include four learning aid use combinations, corresponding to the four time periods that the student could have used learning aids. We excluded all exams where the student completed less than ten percent of the exam questions. In total, the 160 students took 2,989 exams (average 18.68 exams), resulting in 2,989 learning aid use combinations.

Second, we clustered the learning aid usages. Hierarchical cluster analysis was selected for this analysis, because there was no theoretical reason for a priori specification of the number of combinations used by students [4]. Clustering was conducted using five methods: Ward's [10], centroid, median, between-groups linkage, and within-groups linkage.

Third, we evaluated the cluster solutions. We examined possible solutions including three to seven clusters by examining the change in agglomeration coefficients, cluster membership (i.e., excluding any solutions with very small membership), and significance of univariate F-tests [9]. Based on these analyses, the solution using Ward's method with three clusters performed best. These three clusters were significantly different ($p < 0.001$). This solution included one large cluster (n=2,351), and two smaller clusters (n=228 and n=410), which we describe next.

To better understand the clusters, we conducted post hoc comparisons of the means of each learning aid using Games-Howell test, because there are more than two clusters and equal variances are not assumed [4]. This test conducts pairwise comparisons for each learning aid across clusters, and significant differences are identified (at a predefined level, $p < 0.10$ in this exploratory study). The test sorts these means into groups, where the means of learning aid use within a group are not significantly different from others within the same group, but are significantly different from those in other groups.

Table 1. Comparison of Learning Aid Clusters

	F-values[a]		Cluster 1 Low Use		Cluster 2 Moderate Use		Cluster 3 High Use	
Animations	47.678	*	0.00%	L[b]	0.00%	L	2.00%	H
Calculators	3413.733	*	0.10%	L	24.80%	H	1.40%	M
Sample problems	3087.162	*	0.00%	L	6.50%	M	27.00 %	H
Textbook	109.529	*	0.10%	L	2.50%	H	1.30%	M
Videos	103.285	*	0.00%	L	0.00%	L	3.00%	H

[a] Significant at the $p < 0.001$ level.

[b] H, M and L indicate that the mean for the cluster was high, medium or low, respectively, based on Games-Howell Test.

Cluster 1: Low Use. This learning aid use combination represents a minimalist approach to learning aid use and exhibited significantly low levels of all learning aids investigated.

Cluster 2: Moderate Use. This learning aid use combination exhibited greater variability in learning aid use, and exhibited very traditional resources when learning.

Cluster 3: High Use. This learning aid combination exhibited the highest overall use of learning aids, with a preference for media.

We also report the distribution of learning aid combinations used by students. 13% of students do not change learning aid use combinations throughout the course (all Low). 45% of students use two learning aid use combinations during their coursework. (Approximately half utilize Low and High Use combinations, but not Moderate Use, and half utilize Low and Moderate Use combinations, but not High Use; no students use Moderate and High Use combinations without Low Use.) 42% of students use all three combinations.

Do these learning aid use patterns correlate with subsequent exam performance?

Armed with these learning aid patterns, we investigate the relationship between learning aid cluster and exam performance, and conduct a random effects generalized least squares regression analysis using cluster (recoded into dummy variables), exam number, IRT difficulty, and pre-test score as independent variables. The results indicated the predictors accounted for 30.2% of the variance in exam performance ($R^2 = .302$, F(5,2989)=749.67, p = 0.00). Exam performance was significantly influenced by IRT difficulty (ß = -1.50, $p = 0.00$), pretest score (ß = 0.17, $p = 0.001$), and cluster 3 (High Use) (ß = -0.04, $p = 0.00$); cluster 2 (Moderate Use) and test number were not significant. That is, we find that the high use learning aid combination correlates with low exam performance, just as high hint use correlates with low proficiency [1]. It is likely that students who make the most use of learning aids are weaker students, which explains their lower exam scores. The results imply that we need to separate the study of learning aid use in low-ability students from high-ability students.

5. REFERENCES

[1] Goldin, I. M., Koedinger, K. R., & Aleven, V. (2012). Learner Differences in Hint Processing. *Int Ed Data Mining Soc.*

[2] Gray, P. H., & Durcikova, A. (2006). The role of knowledge repositories in technical support environments: Speed versus learning in user performance. *J Man Info Sys*, 22(3), 159-190.

[3] Gray, P.H., & Meister, D.B. (2004) Knowledge sourcing effectiveness. *Manage Sci*, 50(6), 821–834.

[4] Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E., & Tatham, R.L. (2006). *Multivariate Data Analysis* (2nd ed.). New Jersey: Pearson Prentice Hall.

[5] Markus, M. L. (2001). Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. *J Manage Inform Syst*, 18(1), 57-94.

[6] Pashler, H., McDaniel, M., Rohrer, D., & Bjork, R. (2008). Learning styles concepts and evidence. *Psych Science Public Interest*, 9(3), 105-119.

[7] Rowley, J. (2000). Is higher education ready for knowledge management? *Int J of Educ Manage*, 14(7), 325-333.

[8] Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition Instruct*, 2(1), 59-89.

[9] Ulrich, D., & McKelvey, B. (1990). General Organizational Classification: An Empirical Test Using the United States and Japanese Electronic Industry. *Organ Sci*, 1(1), 99-118.

[10] Ward, J.H. (1963). Hierarchical Grouping to Optimize an Objective Function. *J Am Stat Assoc*, 58, 236-244.

Learning to Teach like a Bandit

Mykola Pechenizkiy
Department of Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven
The Netherlands
m.pechenizkiy@tue.nl

Pedro A. Toledo
Department of Computer Science
Universidad de La Laguna
Fac. Física. Francisco Sánchez SN
Santa Cruz de Tenerife, Spain
pedro@isaatc.ull.es

ABSTRACT

Designing a good course curriculum is a non-trivial task many teachers have to deal with on a regular basis. There are multiple learning methodologies available, but some of the basics are common; thus, one of the important steps is to identify key concepts and knowledge or skills prerequisites for mastering them. If this can be done properly, a teacher acting as a course designer can think how to sequence the material. After the first edition of the course the teacher takes into account what went well and what adjustments to the course curriculum would be appropriate. With the growing popularity of ITS and recently MOOCs there are more opportunities for data-driven decisions on how to sequence learning materials and activities to optimize the learning process. Personalizing curriculum to different students is also becoming possible based on how well students learn or are expected to learn. Finding the best possible curriculum for all, a group or an individual student is a nontrivial problem that has an explore-exploit nature. We can use ideas of reinforcement learning and consider course design and learning activities sequencing as a kind of multi-armed bandit problem. We illustrate how to sequence these activities iteratively by employing the genetic process mining framework for generating a population of curriculum candidates from historical data and how to choose these candidates using the Bandit strategy to address the exploration-exploitation trade-off.

Keywords

Iterative course design, reinforcement learning, process mining.

1. INTRODUCTION

The design of a course curriculum or more broadly instructional design has been traditionally an important challenge to educators or teachers responsible for construction of a course [2,3]. And it is a key part of getting satisfactory results in any teaching and learning process. Therefore, teachers have to invest significant time and effort in it.

The design process usually takes a long period of time and it is done using what is called a **curriculum development and implementation cycle** [1] as illustrated in Figure 1.

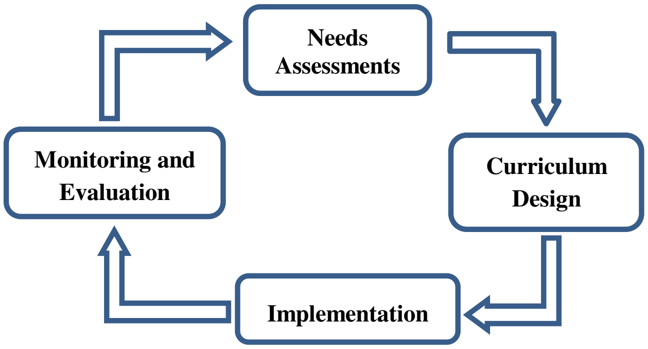


Figure 1 Course Curriculum Design Cycle

The design process may result with a high number of course curriculum alternatives with different task sequencing [4, 5], which would be difficult to evaluate and compare to each other accurately. We propose a new strategy for the curriculum design to construct a data-driven process [8] which may automatically justify changes in the curriculum design.

2. LEARNING TO TEACH LIKE A BANDIT APPROACH

Our course curriculum model describes the relations among the set of activities that may take place during a course, and the resources used with them. To build it, we follow the classical curriculum design cycle. In the first phase (Needs Assessments), a set of curriculum designs are originally proposed by the course teachers based on the course assessments. In the second phase (Curriculum Design), the population is considered the starting population of solutions for the Genetic Process Mining Algorithm. Genetic Process Mining is the technique designed by applying genetic algorithms to perform process mining. In the third phase (Implementation), once new evolved curricula have been obtained, one of them is selected using the Bandit Algorithm. The Multi-Armed Bandit Problem can be modeled as a Single-State Markov Decision Process [6, 7]. To choose, it considers the expected performance distribution of each of the curriculums from the population. The selected curriculum is implemented, and new information is gathered. Finally in the fourth phase (Monitoring and Evaluation), the new data is used to update the curriculum performance information. Back to the first stage of the cycle, the curriculum population is pruned using the updated expected performance distribution. A graphical representation of the approach can be seen in Figure 2.

As a consequence of the running cycle, a multi-objective optimization of the model takes place. Each curriculum selection stage has the consequence of gathering new data from the high performance curriculum implementations. In the long term, the obtained data log will mainly correspond to high performance curriculum implementations. Additionally, each time the Genetic Process Mining algorithm is executed, the obtained models are optimized to fit the gathered data. Consequently, the populations of models obtained in the long-term execution of the cycle correspond to high performance curriculum models that accurately describe the real process implemented for the course.

The described method present serious advantages with respect to the straightforward process mining approach:

Firstly, the inclusion of Multi-Armed Bandit Problem Strategy solves the curriculum selection problem in the exploration exploitation scenario.

Secondly, a continuous strategy of curriculum improvement is defined. This is so because the evolutionary algorithm will use the gathered performance information to transform the population of curriculum models.

Finally, the combination of the Multi-Armed Bandit Problem Strategy and Genetic Process Mining solves the multi-objective problem of curriculum mining. This problem consists on maximizing the model fitting to the data, and the expected performance.

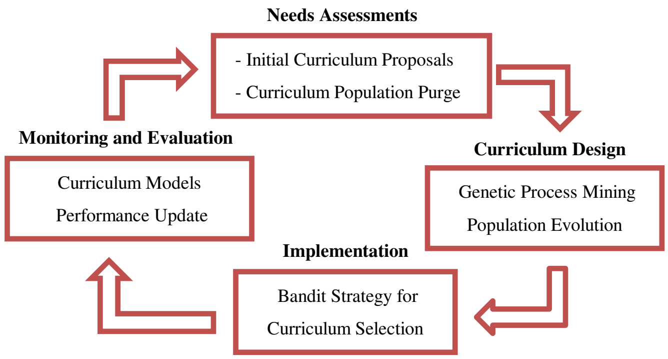


Figure 2 Learning to Teach as a Bandit Approach

2.1 Application Settings

Different scenarios are suitable for the application of the 'Learning to Tech as a Bandit' strategy.

On the one hand, when a new course is designed, multiple possibilities for the curriculum definition are considered. In this case the successive iterations of the course implementation could be used to improve the curriculum design. At each iteration, the whole classroom would follow the selected curriculum until the end. Only after the course is finished, the obtained data from the students' results is analyzed. The information about the evaluation of the implemented curriculum is updated and, based on it, the next curriculum for the following course is implemented. Obviously, it is possible to plan simultaneous courses implementing each one a different curriculum. In that case the curriculum evolution is faster in terms of time spent to discover better solutions. However, higher costs in terms of implementation of poor performance curriculum are also expected.

On the other hand, the scenario of student curriculum personalization in MOOCs is also a plausible setting for the approach. In this case the purpose is not to get the better curriculum design for the course. Furthermore, it is assumed that there is not a singular curriculum to use in a one-fits-all mode. Usually, the group of students of a MOOC is heterogeneous, and therefore the use of personalized curriculum for each user is advisable. In this case, the strategy would consist on obtaining the best fitting curriculum for each.

3. CONCLUSIONS AND FUTURE WORK

This paper proposes a novel approach for curriculum design and adaptation. The hard problem of curriculum design has been formalized in terms of Curriculum Mining. Furthermore, the problem of searching the best design has been stated, pointing out the exploration-exploitation trade off. The approach tackles this issue, with the formulation of a Multi-Armed Bandit Problem. Consequently, the cost of implementation of suboptimal curriculum is minimized. The approach is valid for different settings. It may be either used in the context of new course curriculum design or for student curriculum personalization.

The proposal would need empirical validation to contrast its efficiency in comparison with the traditional curriculum design and personalization methods. Nevertheless, all the ingredients necessary for the implementation are already available: not only the Genetic Process Mining algorithms but also the Multi-Armed Bandit Problem solvers. Therefore, implementing the approach results a straightforward task.

Learning to Teach as a Bandit, is a first attempt to build a data-driven approach to curriculum design, incorporating the scientific method to a traditionally experience based task. The whole educational community would benefit from the approach advantages. Teachers would have a solid guide for curriculum design and a clear method to improve their courses. Additionally, students could have not only courses with a further adaptation to their expectations, but also they could minimize the experience of poorly designed curricula.

4. REFERENCES

[1] Frase, L.E., English, F. W., Poston Jr., W. K. (2000) The Curriculum Management Audit. Improving School Quality. Rowman & Littlefield Education.

[2] Gagné, R. M., & Driscoll, M. P. (1988). Essentials of learning for instruction. Englewood Cliffs, NJ: Prentice-Hall

[3] Gagne, R., Briggs, L. & Wager, W. (1992). Principles of instructional design (4th Ed.). Fort Worth, TX: HBJ College.

[4] Morales, R., Agüera, S. (2002) Dynamic Sequencing of Learning Objects. IEEE International Conference of Advanced Learning Technologies (ICALT)

[5] Wiley, D.A. (2000) Learning Object Design and Sequencing Theory. PhD Dissertation. Birgham Young University.

[6] Mitchell, C.M., Boyer, K. E., Lester, J. C. (2013) A Markov Decision Process Model of Tutorial Intervention in Task-Oriented Dialogue. Artificial Intelligence in Education. LNCS. Vol. 7926, pp 828-831.

[7] Almond, R.G. (2007) An Illustration of the Use of Markov Decision Processes to Represent Student Growth (Learning). Reseach Report. ETS, Princeton, NJ

[8] Koedinger, K. R., Brunskill, E., Baker, R., McLaughlin, E. A., Stamper, J. (2013). New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization. AI Magazine. Vol 34, No 3,3, 27-41.

Matching Hypothesis Text in Diagrams and Essays

Collin F. Lynch
Center for Educational Informatics
North Carolina State University
Raleigh, North Carolina, U.S.A.
collinl@cs.pitt.edu

Mohammad Falakmasir
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, Pennsylvania, U.S.A.
falakmasir@cs.pitt.edu

Kevin D. Ashley
Learning Research & Development Center
University of Pittsburgh
Pittsburgh, Pennsylvania, U.S.A.
ashley@pitt.edu

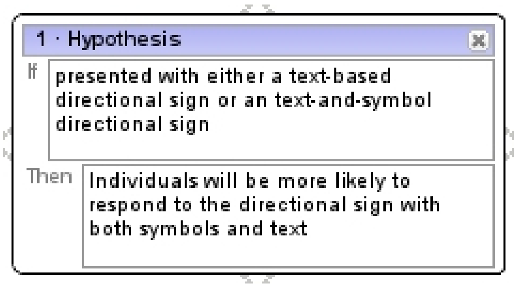
Keywords
Argument Diagramming, Ill-Defined Domains, Intelligent Tutoring Systems, Text-Mining, Multiple-representations

1. INTRODUCTION

We have previously shown that argument diagrams can help students both to read existing arguments [8] and to plan new ones [4]. We have also shown that student-produced diagrams can be graded reliably and evaluated, both by human graders and automatic analysis, to predict subsequent essay grades [4, 6, 5]. Argument diagrams are advantageous for tutoring as they focus students' attention on key structural features of otherwise implicit or opaque arguments as well as supporting empirically-valid automatic assessment and feedback [4]. It has not yet been shown that the content of the argument diagrams closely matches the essay text or that the two can be automatically aligned. Here we show that automatic alignment of hypothesis statements and hypothesis nodes is possible.

A sample hypothesis node is shown in Fig. 1. The ontology used here included nodes representing hypotheses, citations, claims, and the current study. Students added these nodes to a flexible workspace and connected them using supporting, opposing, undefined, and comparison arcs. Hypothesis nodes frame the discussion via a simple *if-then* format. The hypothesis statement tagged in the associated essay is:

> When presented with text-based signs versus signs with text and images or symbols, individuals will be more likely to respond to those signs with both images and text.



1 · Hypothesis

If presented with either a text-based directional sign or an text-and-symbol directional sign

Then Individuals will be more likely to respond to the directional sign with both symbols and text

Figure 1: A sample hypothesis node drawn from a student-produced LASAD diagram.

2. ANALYSIS

Data for this study was drawn from work on argument planning for writing described in [4]. In that study we collected a set of 105 paired diagrams and essays collected in a course on Research Methods at the University of Pittsburgh. Students in this course conducted a group research project. As part of the assignment students produced an argument diagram when planning their essay. The diagrams and essays were graded independently by an expert grader who also annotated the hypothesis statements within the text. The reliability of the grading and annotation was evaluated via a separate inter-grader reliability study where we found 70% agreement on hypothesis tags. 85 of the pairs contained one or more hypothesis nodes in the diagram and one or more tagged hypothesis statements in the essay.

We assessed our primary hypothesis via two types of analyses. In the first analysis we focused on sentence *classification* with the goal of determining whether the textual information from the hypothesis nodes can be used to train efficient classifiers or to improve upon existing techniques. We split the papers into individual sentences using the grader tagging to annotate hypothesis statements. We then extracted two feature vectors for each sentence. We extracted a *static* feature vector for each sentence that reflects the 6 most frequent keyword stems: 'would,' 'likely,' 'hypothe,' 'study,' 'expect,' and 'predict.' Ironically the most predictive single feature was 'predict.'

We then matched each of the candidate sentences with the text drawn from the hypothesis node in the associated diagram and calculated a *similarity* vector. This vector contained five features each of which reflected the output of an existing sentence similarity metric. The features were: Levenshtein distance [3, 11], Jaro-Winkler distance [12, 10], Ratcliff & Obershelp score [9], and two semantic metrics based upon WordNet [1]: Path [2], and Wu & Palmer [13]. For these latter measures, the similarity scores were calculated using only the first sense of the words in each sentence. For diagrams with more than one hypothesis node we computed one similarity vector per node and chose the best result based upon the Ratcliff & Obershelp score.

We trained two sets of classifiers via 10-fold cross-validation using these features. One set of classifiers was trained solely on the *static* vectors and reflected the predictiveness of the individual cue terms while the second *combined* both the static and similarity vectors for each sentence. We chose five standard classification algorithms for this purpose: Naïve Bayes, Nearest Neighbor, Maximum Entropy, Support Vector Machines, and Linear Regression. All of the classifiers were trained and evaluated using the RapidMiner toolkit [7]. For the Linear Regression model we tagged the sentences with a binary output variable and made predictions based upon a fixed cutoff of $\frac{1}{2}$. RapidMiner performs some mechanical filtering of collinear terms.

The most precise classifier was a maximum-entropy model based upon the static features which had a precision of 0.7, a recall of 0.48, and an F1 score of 0.56. The best overall classifier was an SVN model based upon the combined features which also achieved the highest recall and F1 Scores. The precision, recall, and F1 scores for this model were 0.65, 0.65, and 0.65 respectively.

In our second analysis we implemented a second linear ranking function that estimates the likelihood of each sentence being a hypothesis statement based upon the aforementioned features. The weights in this model were trained using leave-one-out cross-validation. For each essay we then selected the sentence or sentences with the highest likelihood of being a hypothesis statement. For this analysis we compared predictions based on the static features alone, similarity alone, and the combined set. These algorithms were trained via leave-one-out cross-validation and were designed to select the best sentence on a per-paper basis. As in the classification study we found that the combined model outperformed the static and similarity models with precision scores of 73%, 66%, and 55% respectively.

3. CONCLUSIONS

We found that combined models which used both the static static features and the similarity measures were better at classifying hypothesis statements and ranking candidate statements within a written essay than either the static or similarity features alone. These results lead us to conclude that it is possible to use this similarity information to link the hypothesis nodes and hypothesis statements most of the time. In future work we plan to test automatic alignment of other diagram components and to investigate other linking mechanisms. We believe that a hybrid model which incorporates information from multiple nodes can be more robust than any individual comparison.

Acknowledgments

This work was supported by NSF award 1122504, “DIP: Teaching Writing and Argumentation with AI-Supported Diagramming and Peer Review,” Kevin D. Ashley PI with Chris Schunn and Diane Litman, co-PIs.

4. REFERENCES

[1] C. Fellbaum. *WordNet: An electronic lexical database.* MIT Press, 1998.

[2] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In Fellbaum [1], pages 265–283.

[3] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet physics doklady*, 10, 1966.

[4] Collin F. Lynch. The Diagnosticity of Argument Diagrams, 2014. (defended January 30th 2014).

[5] Collin F. Lynch and Kevin D. Ashley. Empirically valid rules for ill-defined domains. In John Stamper and Zachary Pardos, editors, *Proceedings of The 7th International Conference on Educational Data Mining (EDM 2014)*. International Educational Datamining Society IEDMS, 2014. (In Press).

[6] Collin F. Lynch, Kevin D. Ashley, and Min Chi. Can diagrams predict essays? In Stefan Trausen-Matu and Kristy Boyer, editors, *Intelligent Tutoring Systems, 12th International Conference, ITS 2014, Honolulu, Hawai'i, USA*, Lecture Notes in Computer Science. Springer, 2014. (In Press).

[7] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1–6, 2006.

[8] Niels Pinkwart, Kevin D. Ashley, Collin F. Lynch, and Vincent Aleven. Evaluating an intelligent tutoring system for making legal arguments with hypotheticals. *International Journal of Artificial Intelligence in Education*, 19(4):401–424, 2009.

[9] J. W. Ratcliff and David Metzener. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, 7(46), 1988.

[10] Wikipedia. Jaro-winkler distance — wikipedia, the free encyclopedia, 2014. [Online; accessed 28-March-2014].

[11] Wikipedia. Levenshtein distance — wikipedia, the free encyclopedia, 2014. [Online; accessed 1-March-2014].

[12] W. E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods (American Statistical Association)*, pages 354–359, 1990.

[13] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico*, 1994.

Matrix Factorization Feasibility for Sequencing and Adaptive Support in ITS

Carlotta Schatten, Ruth Janning, Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
schatten, janning, schmidt-thieme@ismll.uni-hildesheim.de

Manolis Mavrikis
London Knowledge Lab
University of London, UK
m.mavrikis@lkl.ac.uk

ABSTRACT

Performance prediction has the potential of ameliorate the student model of an Intelligent Tutoring System by predicting whether a student mastered or not a specific set of skills. Recently, it has been shown, by means of a simulated learning process, how performance prediction methods based on Matrix Factorization can be used for continuous score prediction and for sequencing contents through a policy inspired by Vygotsky's concept of the Zone of Proximal Development. In this paper we discuss the feasibility of the approach analysing a commercial system dataset. We evaluate performances of the score predictor and feasibility of the Vygotsky policy for sequencing tasks and providing adaptive support.

Keywords

Matrix Factorization, Sequencing, Adaptive Support

1. INTRODUCTION & BACKGROUND

In Intelligent Tutoring Systems (ITS), adaptive sequencers can take past student performance into account to select the next task which best fits the student's learning needs. Simple sequencing policies rely on assumptions such as that a student will be able to solve an exercise of the achieved difficulty level but not the more difficult ones without having completed ones of the previous level. This can be problematic as it requires students to go through all the topics in the current level even if they can answer them successfully with the first attempt. Although the power-law-of-practice [4] would suggest that students should be provided with several opportunities to practice, unnecessary repetition can be detrimental in that it can lead to student frustration and influence their perception of the reliability of the system. One way to approach the problem is based on assessing the student skills and matching them to the required skills and difficulties of the available tasks. For example, in [2] the less known skills by the students are selected to be practiced in the next session. In this scenario two problems arise: 1. Tagging tasks with required skills necessitates experts and thus is a time-consuming, costly process, and, especially for fine-grained skill levels, also potentially subjective. 2. Learning adaptive sequencing models requires online experiments with students and specific data collection policies, that consists, at the beginning, in many randomly proposed tasks. Problem 1. extends also to common performance prediction methods and their extensions: Bayesian Knowledge Tracing (BKT) [1] and Performance Factors Analysis (PFA)[5].

On the contrary, Matrix Factorization (MF), the algorithm we use for performance prediction, is domain agnostic. Its most common use is for Recommender Systems and in previous work [6] we showed how a score prediction method and a simple policy, inspired by Vygotsky's concept of Proximal Development, could be used for ameliorating sequencing in a simulated environment. Despite its plausibility, applying this sequencer in a real use case of an already established ITS and real students requires design decisions that are not well documented. In this paper we show promising preliminary results and work in progress toward the use of the sequencer in a multi-topic commercial ITS. Moreover, we discuss how the performance prediction indications could be used to help hint provision, where Machine Learning was also applied [3]. Our main goal is to present first results towards the integration of the sequencer presented in [6] into an open architecture while discussing its feasibility.

2. FEASIBILITY DISCUSSION

In this section we discuss how the MF can be applied to a commercial system which has over 1000 lessons in 20 topics and was adapted to be used in several countries like United Kingdom, USA, and Russia. We performed a practical feasibility study using a dataset that is composed by data collected from children from five to fourteen years using the ITS in classrooms and homes. A lesson is composed of test and exercise sessions. The exercise session consists of approximately 10 exercises on a topic and specific learning objectives. While trying to solve those exercises a student can consult several hints, one of those is the bottom-out hint, which displays the solution. In order to pass the exercise a student must achieve a score of 7 out of 10 (7/10) that allows them to pass to the test session. There students have to show what they learned answering 5 questions with a score greater than 6/10. The lesson sequencing policy relies on the assumption that a student will be able to solve the exercises of the achieved difficulty level but not the more difficult ones without having completed all the lessons of the previous level. In contrast to state-of-the-art performance prediction, where the main task is to predict the student's correct at first attempt answer, the commercial system uses the score as student's performance measurement. The data granularity level is low if compared with benchmark systems, since we possess a single score record for the 10 questions of the exercises and one record for the five test questions.

2.1 Performance Prediction Feasibility

In this paper we use Matrix Factorization (MF) as score predictor since we do not possess Knowledge Component

Table 1: Performance Prediction Error

Experiments, score range [0,1]	**RMSE**, ± SD over five experiments
Global average	0.3032796
Biased User-Item Exercise	$0.2639167 \pm 3.6989\ 10^{-5}$
Exercise Preprocessing	$0.26061115 \pm 5.97504\ 10^{-5}$

Table 2: Dataset Statistics

Number of Items (Exercise/Topic)	9091/4169
Number of Students	258391
Total Student-Item Interactions	30813070
Total Exercise sessions	17512972
Exercise passed (Score 70-99)	9520278 i.e. 54
Gaming the system (Score 100 + Bottom-out hint)	3988891 i.e. 23%
Total Test sessions	13300098
Test session passed (Score 60-99)	4378461 i.e. 33%
Average score obtained	8.1

information. The matrix $Y \in \mathbb{R}^{n_s \times n_c}$ can be seen as a table of n_c total tasks and n_s students used to learn the students' model, where for some tasks and students performance measures are given. MF decomposes the matrix Y in two other ones $\Psi \in \mathbb{R}^{n_c \times P}$ and $\Phi \in \mathbb{R}^{n_s \times P}$, so that $Y \approx \hat{Y} = \Psi\Phi^T$. Ψ and Φ are matrices of latent features. Their elements are learned with gradient descend from the given performances. This allows computing the missing elements of Y for each student i in each task j of a dataset D. The optimization function is represented by: $\min_{\psi_j,\varphi_i} \sum_{i,j\in D} (y_{ij} - \hat{y}_{ij})^2 + \lambda(\|\Psi\|^2 + \|\Phi\|^2)$,where one wants to minimize the regularized squared error on the set of known scores. MF prediction is computed as:

$$\hat{y}_{ij} = \mu + \mu_{cj} + \mu_{si} + \sum_{p=0}^{P} \varphi_{ip}\psi_{jp} \tag{1}$$

where μ, μ_c and μ_s are respectively the average performance of all tasks of all students, the learned average performance of a content, and learned average performance of a student. The two last mentioned parameters are also learned with the gradient descend algorithm. We followed the standard approach in the field to divide the dataset temporally in two thirds for training and one third for testing, evaluating the performances with the Root Mean Square Error (RMSE). The score, as in [6], is represented in a continuous interval which goes from zero to one. In Table 1 we present Global Average, i.e. a worst case predictor that assumes students will always perform equally to the global score average computed on the training dataset. The Biased User-Item predictor, instead, uses only the biases μ, μ_s, and μ_c of Eq. 1, i.e. the latent features number P is set to zero. Consequently, out of Table 1 one can see the contribution of the single components of Eq. 1 in ameliorating the prediction. According to the results, the dataset is suitable for the task and MF is able to predict a continuous interval performance in a multiple-topic scenario.

2.2 Sequencing and Hinting Policy Feasibility

The Vygotsky Policy based Sequencer (VPS) in [6] is composed of two components: the Vygotsky Policy (VP) and a Performance Predictor. Given MF as predictor, we want to sequence tasks in a way that attempts to keep students in the so-called zone of proximal development (ZPD), which, in our context, we associate with tasks that are neither too easy nor too difficult to accomplish without much help. This concept is formalized by the following formula: $c^{t*} = \text{argmin}_c \left|y_{th} - \hat{y}^t(c)\right|$, where y_{th} is a threshold score that will challenge the students and keep them in the ZPD. The policy will select at each time step the content c^{t*} with the predicted score $\hat{y}^t$ at time t most similar to y_{th}.

Considering our use case with a score range for passing of 6-10/10, y_{th} should be set in the middle of the interval, so that the most exercise selected are predicted with a score of 8. This avoids that, in case of no available tasks predicted with exactly y_{th}, the policy does not select exercises which are out of the score range for passing and consequently minimizing the risk of MF incorrect prediction. With an RMSE of ±2.6 (Table 1), the selected lessons are approximately always in the aforementioned range.

Another use of the performance prediction is to enhance feedback provision to students as it provides the possibility of developing 'task-independent' adaptive support, i.e. hints that relate to students' interaction overall rather than the specific problem solving steps. At least in the case of the commercial ITS under investigation, problem solving steps are dealt by different components and in fact operate as individual learning objects. Examples of such feedback include the provision of support at the beginning or end of the exercise but also during an exercise if, for example, there is no task-specific help to provide. Accordingly, when students start their experience, it is helpful to provide suggestions about which topic(s) to study based on the MF prediction. The topic having the most tasks in the ZPD, should be proposed. During an exercise, if students attempt to ask for help but the prediction above indicates that they do not seem to need it, the system can restrict help depending on the current answers and attempts on the exercise as in [3].

3. FUTURE WORK

In this paper we discussed the feasibility of employing a Matrix Factorization prediction to sequencing and providing adaptive support. Our plan is to apply the sequencer as task and hint sequencer. However, a still open issue is how to evaluate the contribution of the VPS. Considering the number of exercises passed and failed reveals only a part of the incorrectly sequenced tasks, i.e. the too difficult/failed ones. We believe there are possible improvements on other aspects of the interaction such as a reduction of the 'gaming the system' behaviour (see Table 2) as indicated by the tasks that are achieved with 100%, having accessed the bottom-out-hint and not spending enough time to reflect on it.

Acknowledgements. This research was funded from the FP7 STREP iTalk2Learn (www.iTalk2Learn.eu).

4. REFERENCES

[1] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMAI*, 1994.

[2] K. Koedinger, P. Pavlik, J. Stamper, T. Nixon, and S. Ritter. Avoiding problem selection thrashing with conjunctive knowledge tracing. In *EDM*, 2011.

[3] M. Mavrikis. Data-driven modelling of students' interactions in an ILE. In *EDM*, 2008.

[4] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition.*

[5] C. H. Pavlik, P. and K. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *AIED*, 2009.

[6] C. Schatten and L. Schmidt-Thieme. Adaptive content sequencing without domain information. In *CSEDU*, 2014.

Microgenetic Designs for Educational Data Mining Research

Taylor Martin[1]
Ani Aghababyan[3]
Utah State University
Instructional Technology and Learning Sciences
2830 Old Main Hill
Logan, UT 84322-2830
taylor.martin@usu.edu

Nicole Forsgren Velasquez[2]
Jason Maughan[4]
Utah State University
Management of Information Systems
3515 Old Main Hill
Logan, UT 84322-3515
nicolefv@gmail.com

Philip Janisiewicz[5]
University of Texas at Austin
Curriculum and Instruction
1 University Station D5700
Austin, TX 78712
pjanisiewicz@gmail.com

ABSTRACT

Educational Data Mining (EDM) methods can expand the reach of microgenetic research. This paper presents an example of our pilot work using microgenetic analysis in the context of fraction game data, where we characterize student activity based on clustered sequences of actions. We cluster sequences by the similarity between them, calculated using optimal matching techniques.

1. INTRODUCTION

Microgenetic research investigates processes of learning ([8]; [11]), rather than simply focusing on products of learning. Three main elements distinguish microgenetic research designs: 1) studies occur when the topic is likely to learned, 2) observations of learning behavior are dense, and 3) analysis is conducted on an instance by instance basis [6]. To date, the grain size for these studies has been fairly large, and the number of time points has been relatively small. These elements can be greatly improved using EDM methods. A few researchers have begun expanding microgenetic methodology using EDM methods (e.g., [2]; [4]; [5]) but this work is still in early stages. In addition, many EDM researchers use methods and conduct analyses that could be productive for microgenetic research (even if they do not place their work within the microgenetic paradigm). Some of these approaches include process and sequence mining, some uses of hidden markov models, and dynamic bayesian networks.

2. REFRACTION

Third grade students (approximately 8–9 years old) played Refraction (http://play.centerforgamescience.org/refraction/site/), an online game based on fraction learning through splitting. In the game level used for this study, students create laser beams of 1/6 and 1/9 using a combination of 1/2 and 1/3 splitters. Students played the level twice: once at the start of gameplay (the prelevel) and again after playing the series of game levels (the postlevel). As students could stop play at any time, we had uneven numbers who completed the prelevel (N = 3,258) and the postlevel (N = 1,127).

3. ANALYSIS

Our unit of analysis is a "board state," or the configuration of the mathematical pieces of the game after a student makes a change. The two attributes of a board state we included in our analysis were initial splitter used (1/2 splitter or 1/3 splitter) and node depth. Solving the level requires starting with a 1/3 splitter, so the initial splitter variable indicates the quality of the board state. Node depth is the number of nodes, or levels of splitting, there are on a board state.

We employed the Needleman-Wunsch algorithm for optimal matching in R using the package TraMineR version 1.8-8, ([7]; [10]). This algorithm computes the "cost" of transforming one sequence into another based on insertions, deletions or substitutions. We set all costs equally at 1 as our events are all of the same type. To account for the discrepancies in our sequence lengths (prelevel range 1-82; postlevel range 1- 140), we used Abbott's normalization approach to standardize optimal matching distances ([1]; [7]).

We then used the distance matrix generated with optimal matching in a hierarchical cluster analysis [9] using Ward's, single linkage, and weighted average methods. We evaluated the number of clusters using dendrograms, and referenced group membership to ensure no clusters were too small. Finally, we inspected a visualization of each cluster solution for interpretation. The solution using Ward's analysis with seven clusters performed best (See Figures 1-6).

⬜	0_0	🟥	2_2	🟩	3_3	🟩	5_2	🟧	6_3
🟦	1_2	🟥	2_3	🟪	4_2	🟦	5_3	🟩	7_3
🟦	1_3	🟩	3_2	🟪	4_3	🟧	6_2	🟨	8_3

The first number in the pair is the node depth. The second number is for the 1/2 or 1/3 initial splitter.

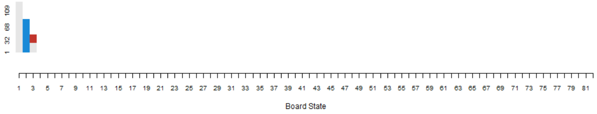


Figure 1. Minimal.

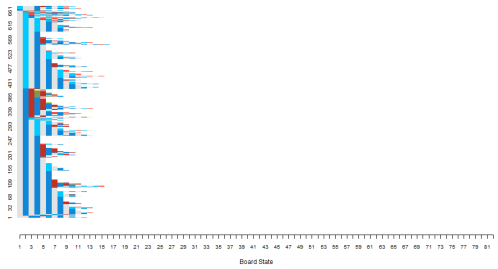


Figure 2. Halves.

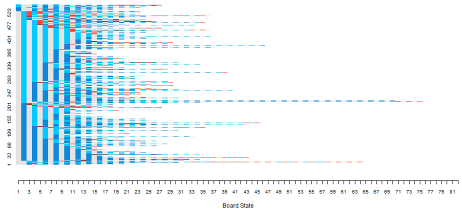


Figure 3. Exploring Halves.

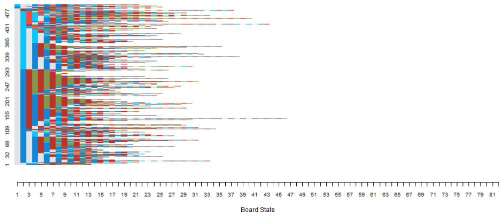


Figure 4. Exploring.

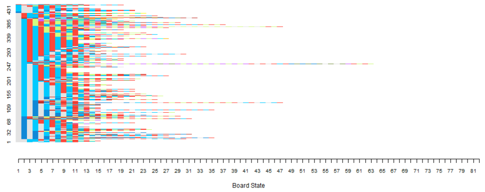


Figure 5. Exploring Thirds.

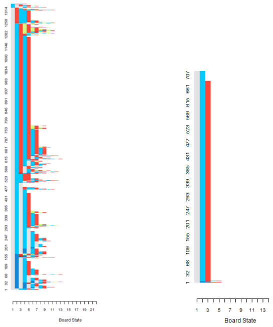


Figure 6. Thirds (left) and Efficient (right).

4. RESULTS

a. *Minimal* (N prelevel = 129; N postlevel = 4): very short sequences, very low node depths, and all 1/2 initial splitters.

b. *Halves* (N prelevel = 651; N postlevel = 26): medium sequences, shift from low node depth to higher, shift from mostly 1/2 initial splitter to some 1/3 initial splitters, show "reset" pattern, or clearing laser and starting over.

c. *Exploring Halves* (N prelevel = 536; N postlevel = 14): very similar to the Halves cluster, except longer sequences.

d. *Exploring* (N prelevel = 341; N postlevel = 165): greater exploring: many long sequences, try higher node depths, use both the 1/2 and 1/3 initial splitters.

e. *Exploring Thirds* (N prelevel = 359; N postlevel = 90): very similar to the Exploring cluster, except mostly 1/3 initial splitters.

f. *Thirds* (N prelevel = 859; N postlevel = 490): relatively short sequences, mostly node depth of 1 or 2, mostly 1/3 splitter.

g. *Efficient* (N prelevel = 383; N postlevel = 387): very short sequences, nearly all sequences identical: the start state, a state with node depth of 1 and 1/3 initial splitter, and a state with node depth of 2 and 1/3 initial splitter.

Most students, regardless of cluster membership on the prelevel, were in the Thirds or Efficient clusters on the postlevel (see Table 1). The marginal homogeneity nonparametric test for related samples of ordinal data [3] showed that this change was significant (p < .001).

Table 1. *Change in Cluster Membership From Pre- to Postlevel: Percentage of Students*

				Postlevel			
Prelevel	Minimal	Halve s	Exploring Halves	Exploring	Exploring Thirds	Thirds	Efficient
Minimal	0%	3%	0%	23%	3%	40%	31%
Halves	0%	6%	3%	18%	9%	45%	18%
Exploring Halves	1%	3%	1%	21%	13%	42%	20%
Exploring	0%	1%	2%	12%	10%	50%	26%
Exploring Thirds	0%	0%	2%	21%	10%	43%	24%
Thirds	1%	1%	1%	12%	7%	44%	35%
Efficient	1%	2%	0%	6%	4%	40%	47%

Students in the Thirds and Efficient clusters were more likely to succeed on the both the pre- and postlevels than those in the other clusters; prelevel $\chi^2(1,6) = 1353.39$; $p < .001$; postlevel $\chi^2(1,6) = 605.32$; $p < .001$.

5. CONCLUSIONS

While this case demonstrates the utility of this approach as a microgenetic method in EDM, our next steps will be to extend this method to examine change over days or weeks of a learning event, to further test the utility of this method.

6. REFERENCES

[1] A. Abbott and A. Hrycak. Measuring resemblance in sequence data: An optimal matching analysis of musicians' careers. American journal of sociology, 96(1), 1990.

[2] R.S.J.d. Baker, A. Hershkovitz, L.M. Rossi, A.B. Goldstein, S.M. Gowda. Predicting Robust Learning With the Visual Form of the Moment-by-Moment Learning Curve. Journal of the Learning Sciences, 22 (4), 639-666, 2013.

[3] W. Barlow. Modeling of categorical agreement. In P. Armitage & T. Colton (Eds.), The encyclopedia of biostatistics (pp. 541-545). New York, NY: Wiley, 1998.

[4] M. Berland, T. Martin, T. Benton, C. Petrick Smith, & D. Davis. Using Learning Analytics to Understand the Learning Pathways of Novice Programmers. Journal of the Learning Sciences, 22(4), 564–599, 2013.

[5] P. Blikstein, M. Worsley, C. Piech, A. Gibbons, M. Sahami, & S. Cooper. Programming Pluralism: Using Learning Analytics to Detect Patterns in Novices' Learning of Computer Programming. International Journal of the Learning Sciences Special Issue on Learning Analytics, 2013.

[6] L. K. Fazio and R. S. Siegler. Microgenetic learning analysis: A distinction without a difference. Human Development, 56(1): 52-58, 2013.

[7] A. Gabadinho, G. Ritschard, M. Studer, and N. S. Müller. Mining sequence data in r with the traminer package: A users guide for version 1.2. Geneva: University of Geneva, 2009.

[8] D. Kuhn. Metacognitive development. Current directions in psychological science, 9(5): 178-181, 2000.

[9] M. Lorr. Cluster analysis for social scientists. Jossey-Bass San Francisco, 1983.

[10] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, 48(3): 443-453, 1970.

[11] R. S. Siegler and K. Crowley. The microgenetic method: A direct means for studying cognitive. American Psychologist, 46(6), 606, 1991.

Mining and Identifying Relationships Among Sequential Patterns in Multi-Feature, Hierarchical Learning Activity Data

Cheng Ye, John S. Kinnebrew, Gautam Biswas
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
{cheng.ye, john.s.kinnebrew, gautam.biswas}@vanderbilt.edu

***Abstract*—Computer-based learning environments can produce a wealth of information on each student action, which can often be represented at multiple levels of abstraction and with a variety of features. This paper extends an exploratory sequence mining methodology for assessing and comparing students' learning behaviors by autonomously identifying abstraction levels in a hierarchical taxonomy of actions and their potential features. We apply this methodology to action data gathered from the Betty's Brain learning environment. The results illustrate the potential of this methodology in identifying and comparing learning behavior patterns across groups of students with complex, hierarchical action and action feature definitions.**

I. Introduction

In order to more effectively teach and promote skills required in the modern world, computer-based learning environments (CBLEs) have become more complex and open-ended. In CBLEs, individual student actions can often be represented at multiple levels of abstraction and with a variety of features describing different aspects, contexts, and results of the action.

Sequence mining is widely used in extracting knowledge from databases of human-generated activity data. Further, researchers have applied sequence mining techniques to a variety of educational data in order to better understand and scaffold learning behaviors. In previous work, we have compared sequential patterns derived from student activity sequences to identify ones that differ in usage between two or more groups of students [1], [2] and over time [3].

In this paper, our approach integrates and goes beyond work in differentiating student groups by sequential patterns of behavior [1], [2], as well as work in employing multiple, hierarchically-defined features/dimensions of information in identifying frequent sequential patterns [5], [6]. In particular, the previous work has focused on identifying the *most specific*, detailed frequent sequential patterns, which we extend by identifying a level of specificity (or conversely, generality) that is *most appropriate* for representing sequential patterns that differentiate student groups.

We present example results from the application of this data mining methodology to learning interaction trace data gathered during a middle school class study with the Betty's Brain learning environment. These results illustrate the potential of this methodology in identifying and comparing learning behavior patterns across groups of students with complex, hierarchical action and action feature definitions.

II. Multi-Feature, Hierarchical, Differential Sequence Mining Methodology

Our approach to effectively mining important patterns in Multi-Feature, Hierarchical (MFH) learning activity sequences employs five primary steps:

1) Define MFH action representation to extract MFH action sequences from student activity traces.
2) Flatten action representation to obtain the most specific action definitions (within frequency constraints) for use with sequence mining methods.
3) Employ DSM to identify differentially-frequent (flattened) activity patterns that distinguish the student groups.
4) Identify hierarchical relationships among mined patterns in the form of directed, acyclic graphs of patterns incorporating different features and levels of detail.
5) Identify the best pattern representations by collapsing more specific pattern nodes into the more general ones that provide a similar degree of differentiation between student groups.

In the final step of this methodology, we iteratively identify and collapse the link for which the parent and child patterns are most similar in terms of their differentiation of the student groups. This similarity is calculated as the difference in effect sizes (by pattern occurrence across the two student groups) between the parent and child patterns with a consideration of the *direction* of the effect (i.e., if the parent pattern occurs more frequently in one student group, but the child parent occurs more frequently in the other student group, then the difference is calculated by summing the effect sizes instead of subtracting them).

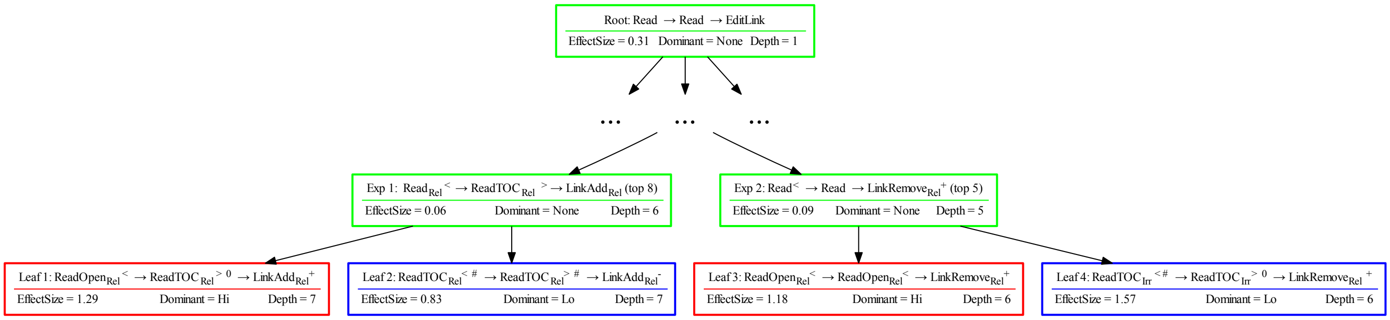

Root: Read → Read → EditLink
EffectSize = 0.31 Dominant = None Depth = 1
...
...
...
Exp 1: $\text{Read}_{\text{Rel}}{}^{<}$ → $\text{ReadTOC}_{\text{Rel}}{}^{>}$ → $\text{LinkAdd}_{\text{Rel}}$ (top 8)
EffectSize = 0.06 Dominant = None Depth = 6
Exp 2: $\text{Read}^{<}$ → Read → $\text{LinkRemove}_{\text{Rel}}{}^{+}$ (top 5)
EffectSize = 0.09 Dominant = None Depth = 5
Leaf 1: $\text{ReadOpen}_{\text{Rel}}{}^{<}$ → $\text{ReadTOC}_{\text{Rel}}{}^{>\,0}$ → $\text{LinkAdd}_{\text{Rel}}{}^{+}$
EffectSize = 1.29 Dominant = Hi Depth = 7
Leaf 2: $\text{ReadTOC}_{\text{Rel}}{}^{<\,\#}$ → $\text{ReadTOC}_{\text{Rel}}{}^{>\,\#}$ → $\text{LinkAdd}_{\text{Rel}}{}^{-}$
EffectSize = 0.83 Dominant = Lo Depth = 7
Leaf 3: $\text{ReadOpen}_{\text{Rel}}{}^{<}$ → $\text{ReadOpen}_{\text{Rel}}{}^{<}$ → $\text{LinkRemove}_{\text{Rel}}{}^{+}$
EffectSize = 1.18 Dominant = Hi Depth = 6
Leaf 4: $\text{ReadTOC}_{\text{Irr}}{}^{<\#}$ → $\text{ReadTOC}_{\text{Irr}}{}^{>\,0}$ → $\text{LinkRemove}_{\text{Rel}}{}^{+}$
EffectSize = 1.57 Dominant = Lo Depth = 6

Fig. 1. Pattern tree illustrating some Hi/Lo behavior differences

III. Results and Conclusion

The data employed for this analysis consists of student interaction traces from the Betty's Brain [4] learning environment. In Betty's Brain, students learn about a science process using a set of hypermedia resources organized into sub-topics by scientific processes and teach a virtual agent, Betty, about what they have learned by building a causal map. In this analysis, we considered the additional action features listed in Table I to analyze the behavior of 8th-grade students from a recent middle Tennessee classroom study in experimental conditions receiving support for identifying causal relationships in the resources. For the differential aspect of the analysis, we focus on the difference between the 16 high-performing (Hi) and 8 low-performing (Lo) students as determined by the quality of their final causal maps.

TABLE I. Action Feature Dimensions

Actions	Dimension	Value	Symbol
[All except Quiz & Explain]	Relevance	Yes	Rel
[All except Quiz & Explain]	Relevance	No	Irr
Read	Previous (Full) Read	Yes	♯
Read	Previous (Full) Read	No	0
Read	Length	Full	>
Read	Length	Short	<
EditLink	Map Score Change	Increase	+
EditLink	Map Score Change	Decrease	-
EditLink	Map Score Change	No Change	=

With an effect size cutoff of 0.8 to only consider relatively large differences between groups, we identified 312 differential activity patterns, which resulted in 175 pattern trees. In total, there were 913 hierarchical links in the resulting pattern trees and 350 intermediate pattern nodes (i.e., those that are not a leaf pattern identified from the application of DSM nor a root pattern representing the most general form of a set of related leaf patterns). With these pattern trees, we employed the link collapsing described in Section II to identify the most important pattern nodes and hierarchical relationships.

Figure 1 illustrates part of the pattern tree created for a sequence of two reads followed by a map edit, which occurred frequently in both the Hi and Lo group. However, various features of the reading and editing actions allow us to clearly distinguish the Hi group from the Lo group in more specific versions of this pattern illustrated by the lower layer of nodes in Figure 1. In addition to better understanding differences in the skills and approaches, these pattern nodes that are not collapsed into more general versions represent a minimal level of detail that can be used to predict and scaffold students during learning with respect to their likely group characterization.

Conversely, nodes collapsed into their parents represent additional detail that is not particularly important for distinguishing the groups. For link editing followed by taking a quiz and getting an explanation from Betty. Although the initial DSM analysis identified three patterns for link editing followed by taking a quiz and getting an explanation of a quiz question. These leaf pattern nodes, which differed by whether the edit was adding a (correct or incorrect) link or removing an (incorrect) link, were collapsed up to the root pattern early in the search for the best representation level. This indicates that simply following a link edit by a quiz and explanation is characteristic of the Hi group, regardless of the specific details of the link edit, including whether it was correct or not. Thus, this approach to collapsing links in the pattern trees, not only allows the researcher to focus on a smaller subset of important patterns, but also contributes to more accurate interpretation and student characterization during learning by identifying the features and level of specificity necessary for differentiating student groups.

Acknowledgments

This work was supported by IES CASL grant # R305A120186.

References

[1] J. S. Kinnebrew and G. Biswas. Identifying learning behaviors by contextualizing differential sequence mining with action features and performance evolution. In *Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012)*, Chania, Greece, June 2012.

[2] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5(1):190–219, 2013.

[3] J. S. Kinnebrew, D. L. Mack, and G. Biswas. Mining temporally-interesting learning behavior patterns. In S. K. D'Mello, R. A. Calvo, and A. Olney, editors, *Proceedings of the 6th International Conference on Educational Data Mining*, pages 252–255. Memphis, TN, USA, 2013.

[4] K. Leelawong and G. Biswas. Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208, 2008.

[5] M. Plantevit, A. Laurent, D. Laurent, M. Teisseire, and Y. W. Choong. Mining multidimensional and multilevel sequential patterns. *ACM Transactions on Knowledge Discovery from Data*, 4(1):4:1–4:37, Jan. 2010.

[6] M. Plantevit, A. Laurent, and M. Teisseire. Hype: mining hierarchical sequential patterns. In *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, pages 19–26. ACM, 2006.

Mining Coherent Evolution Patterns in Education through Biclustering

André Vale
Instituto Superior Técnico
Av. Rovisco Pais 1
1049-001 Lisboa, Portugal
andre.vale@tecnico.ulisboa.pt

Sara C. Madeira
Instituto Superior Técnico
Av. Rovisco Pais 1
1049-001 Lisboa, Portugal
sara.madeira@tecnico.ulisboa.pt

Cláudia Antunes
Instituto Superior Técnico
Av. Rovisco Pais 1
1049-001 Lisboa, Portugal
claudia.antunes@tecnico.ulisboa.pt

ABSTRACT

With the spread of information systems and the increased interest in education, the quantity of data about education has exploded along with a new field - Educational Data Mining. Predicting students' performance has been approached by several techniques, but the combination of supervised and non-supervised techniques appeared as a new tool for improving the results. Biclustering algorithms have been successfully applied in areas such as gene expression data and information retrieval, but not used in the educational context. In this paper, we show how to apply biclustering techniques to educational data and to use its results as features to improve the prediction of student's performance.

Keywords

Educational Data Mining, Biclustering, Student's Performance, Coherent Evolution Patterns

1. INTRODUCTION

The prediction of students' performance has deserved a significant attention in Educational Data Mining (EDM) research, with several distinct approaches being proposed, mostly using classification and regression techniques. With the advances and stabilization of these techniques, it is easy to accept that the accuracy results do not depend on the technique used, but on data themselves, both on the training data and on the target variable [8].

While classification tries to find a model to predict an outcome, non-supervised techniques, as pattern mining and clustering, are able to explore the data for identifying frequent behaviors. Previous studies [1, 2] have shown that sequential pattern mining is suited to discover patterns able to model students behaviors, which in turn can be used to enrich training data, improving global classification accuracy on more than 10% [2].

Clustering is perhaps one of the most important tools for both exploratory and confirmatory analysis. Indeed, it is a technique to discern meaningful patterns in unlabeled data by grouping together data points that are similar. Biclustering algorithms [5] are a recent alternative to traditional clustering methods that allows the discovery of local patterns rather than global ones. Besides discovering sequential patterns identified by pattern mining algorithms, biclustering is able to discover other sequential patterns that reveal coherent evolutions [3, 6].

Although both the literature on EDM and biclustering topics are vast, and the results are positive, the combination of these topics is almost nonexistent. Only recently Trivedi et al. [9] applied this technique to education. In their work, they used the idea of co-clustering (namely biclustering) students and their tutor interaction features and interleave it with a bagging strategy which they used previously with clustering [10] for prediction of out-of-tutor performance of students. The results obtained were better than the baseline and also indicated that the dynamic assessment condition returns in a much better prediction of student test scores when compared to the static condition. However, they used one of the most basic techniques of biclustering, using k-means clustering algorithm to cluster students and features (rows and columns). Separately and then combining both clustering results to derive biclusters. This clustering combination is probably the reason why they obtained modest improvements compared with their previous clustering works.

In this paper, we propose to explore biclustering to discover new patterns in educational data and make use of these patterns to enrich training data in order to improve the prediction of students' performance.

2. BICLUSTERING FOR EDM

Biclustering can be applied whenever the data to analyze has the form of a real-valued or symbolic matrix A, where the value a_{ij} represents the relation between row i and column j, and the goal is to identify subsets of rows with certain coherence properties in a subset of the columns. The goal of biclustering algorithms is to identify a set of biclusters. Let A be a matrix defined by its set of rows, R, and its set of columns, C. Then we can define a bicluster $B = (I, J)$ as a submatrix A_{IJ} defined by $I \subseteq R$, a subset of rows, and $J \subseteq C$, a subset of columns [5]. This set of biclusters $B_k = (I_k, J_k)$ satisfies specific characteristics of homogeneity, that can be grouped in four categories: a) Biclusters with constant values; b) Biclusters with constant values on rows or columns; c) Biclusters with coherent values; and d) Biclusters with coherent evolutions. The first three classes analyze directly the numeric values in the data matrix and try to find subsets of rows and subsets of columns with similar behaviors. The fourth class aims to find coherent behaviors regardless of the exact numeric values in the data matrix. The type of patterns in a) and c) can be found using pattern mining, but the ones in b) and d) are not. The work by Madeira and Oliveira [5] presents a deep survey on this topic, describing the most important algorithms.

Studies have demonstrated that sequential pattern mining can be successfully applied for mining students [1] and teachers' frequent behaviours [2], which in turn may enrich training data for improving classification. As explained, biclustering is able to identify more patterns than pattern mining, in particular, patterns that reveal coherent evolutions. In this manner, we propose to explore biclustering algorithms for identifying patterns that may improve the classification task, as previously performed with sequential pattern mining [2].

In the educational context, matrix A targeted by biclustering algorithms can be any matrix relating two distinct entities, whose relation can be measured, and expresses some result. For example, a matrix relating students and subjects through achieved marks, where we might be interested on finding a group of students that shows the same evolution in a particular subset of subjects, but also a matrix for subjects and time, reporting the average performance of students enrolled on the subject in a particular term.

There are a large number of existing approaches for biclustering that finds different types of biclusters and thus different types of frequent patterns. In our case we are interested in biclusters with coherent evolutions, since existing pattern mining approaches are not able to identify them. In this work, we use the algorithms most cited in the literature to find these types of patterns, namely *Bimax* [7], *xMOTIFs* [6], *ISA* [4] and *OPSM* [3].

Table 1 provides an example of a mark matrix with 10 students and 7 subjects where we draw examples of the types of biclusters these algorithms can find. Bicluster ***B₁*** presents students who had the same marks on all the subjects - a bicluster with constant values that the *Bimax* can get; ***B₂*** has students who have constant marks on different subjects – a bicluster with coherent values on the rows found by *xMOTIFs*; ***B₃*** has students that have all the same notes on the same subjects – a bicluster with coherent values on the columns found by ISA; and finally ***B₄*** shows students that have a coherent evolution between subjects, in this particular case, students' marks satisfy the following evolution pattern: *Subject 3 < Subject 2 < Subject 1 < Subject 4* - a bicluster with coherent evolutions that *OPSM* can find.

Table 1. Example of different types of biclusters in a matrix with marks of ten students at seven subjects.

	Subject 1	*Subject 2*	*Subject 3*	*Subject 4*	*Subject 5*	*Subject 6*	*Subject 7*
Student 1	18 ***B₂***	12	18	10	12	18	12
Student 2	17	16	17	13	15	17	16
Student 3	19	16	19	17	16	19	15
Student 4	17	18	17	***B₁*** 19	19	16	17
Student 5	***B₄*** 18	16	14	19	19	19	18
Student 6	17	15	13	19	14	17	16
Student 7	19	18	16	20	***B₃*** 17	18	19
Student 8	12	16	13	14	17	18	19
Student 9	16	18	15	16	17	18	19
Student 10	13	14	10	11	13	12	10

3. CASE STUDY AND CONCLUSIONS

The data used was gathered from a graduation program (LEIC) at Instituto Superior Técnico (IST), Universidade de Lisboa, considering the student records between 1997 and 2012. The program has the duration of three years (6 semesters) with 30 subjects, and after it, students usually follow to the master program (MEIC). The task was to predict the marks of LEIC students when finishing MEIC. In order to achieve our goal, we analyzed a matrix (*students x subjects*) with 443 students and 20 subjects from LEIC, with students' marks in the cells - numbers between 10 and 20. By applying biclustering algorithms mentioned before to the matrix, we obtained 16 biclusters with *OPSM*, 975 with *xMotifs*, 308 with *ISA* and 39 with *Bimax*. In addition to the data in the matrix, we appended a class label (the mark obtained at the end of the master's program - Fair, Good and Very Good) and obtained our training dataset (baseline). As in [2], a new dataset can be obtained from the previous one, enlarged by *k* Boolean attributes, one for each bicluster. Each bicluster attribute is then filled with the true value whenever the bicluster has the student instance and false otherwise. Classification was performed by *Weka*, with decision trees, using cross-validation with 10 folds for the performance evaluation of decision trees. We then use a feature selection (FS) method, wrapper, to obtain the best attributes. Without FS and biclusters, we got 52.9% of correctly classified instances. If we add the biclusters, the precision go down to 51.8%, this happens because the model is starting to overfitting. Using FS without biclusters we had a precision of 60.1%, and if we add the biclusters we obtained 65.8% (**Figure 1**).

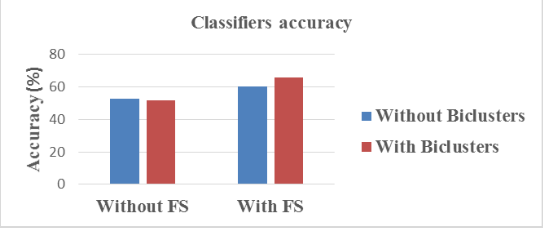


Figure 1. Precision of classifiers accuracy.

With this study we demonstrated that we can improve the accuracy of the decision tree model by more than 5% not having any strict condition to choose the biclusters that are more interesting to use. As such, we believe that after applying more effective metrics to choose the biclusters according to their quality, we can achieve an even better model accuracy. For future work we will develop metrics to apply automated techniques to choose the best biclusters, so we can distinguish the biclusters of interest regardless of what we have in the rows and columns of the matrix.

ACKNOWLEDGMENTS

This work is partially supported by Fundação para a Ciência e Tecnologia under research project *educare* (PTDC/EIA-EIA/110058/2009).

REFERENCES

[1] Antunes, C. 2008. Acquiring Background Knowledge for Intelligent Tutoring Systems. *EDM*. (2008).

[2] Barracosa, J. and Antunes, C. 2011. Anticipating teachers' performance. *KDD 2011 Workshop*. (2011).

[3] Ben-Dor, A. and Chor, B. 2003. Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of computational*. (Jan. 2003).

[4] Bergmann, S. et al. 2003. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical review E*. (2003).

[5] Madeira, S. and Oliveira, A. 2004. Biclustering algorithms for biological data analysis: a survey. *Biology and Bioinformatics, IEEE*. (2004).

[6] Murali, T. and Kasif, S. 2003. Extracting conserved gene expression motifs from gene expression data. *Pacific Symposium on Biocomputing*. (Jan. 2003).

[7] Prelić, A. et al. 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics (Oxford, England)*. (2006).

[8] Romero, C. and Ventura, S. 2010. Educational Data Mining: A Review of the State of the Art. *Trans. Sys. Man Cyber Part C*. (2010).

[9] Trivedi, S. et al. 2012. Co-Clustering by Bipartite Spectral Graph Partitioning for Out-of-Tutor Prediction. *International Educational Data Mining Society*. (2012).

[10] Trivedi, S. et al. 2011. Spectral Clustering in Educational Data Mining. *Educational Data Mining* (2011).

Mining Multi-dimensional Patterns for Student Modelling

Andreia Silva
Instituto Superior Técnico - Universidade de Lisboa
Av. Rovisco Pais 1, Lisbon, Portugal
andreia.silva@tecnico.ulisboa.pt

Cláudia Antunes
Instituto Superior Técnico - Universidade de Lisboa
Av. Rovisco Pais 1, Lisbon, Portugal
claudia.antunes@tecnico.ulisboa.pt

ABSTRACT

A careful analysis of educational data reveals their multi-dimensional nature, with several orthogonal dimensions from students to teachers, courses, evaluation items, topics, etc. In addition, their historical nature translates into large data warehouses, which are modeled through inter-connected huge tables that encompass data from several distinct perspectives. Despite the recent advances in big data research for this educational domain, the ability to consider these very large multi-dimensional datasets remains unexplored. In this paper, we explore a multi-dimensional algorithm in order to find multi-dimensional patterns in education, which in turn will be used to model student behaviors. Experimental results in a real case study show a significant improvement on the prediction of student results, when compared with the same classifiers trained without those patterns.

1. INTRODUCTION

The long history of education as an institution lead to huge amounts of data, requiring automatic means for exploring them. Educational data mining (EDM) [1] gives a first opportunity for exploring these data, providing the adequate tools to predict students performance and dropouts, but also for understanding student behaviors [4].

Despite the encouraging results, few approaches were dedicated to explore the multi-dimensionality of data. Definitely, the educational process encompasses a set of different entities, characterized by distinct sets of attributes. Each kind of entity is usually known as a *dimension* (e.g. students, teachers, courses). In the intersection of these dimensions occurs the educational process, with the materialization of its *events* (e.g. the marks obtained by students). Multi-dimensional models, such as *star schemas*, are recognized as the most usual schemas to model these kinds of data. They consist in a central table containing the occurring events, and a set of surrounding tables, comprising the specific data about each dimension. Figure 1 shows an example in the educational domain with 2 star schemas: one modeling student enrollments and another teachers quality assurance surveys.

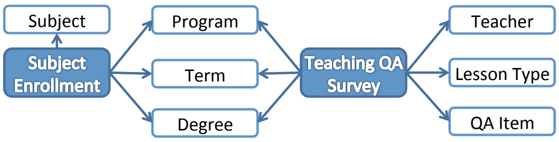


Figure 1: Example of an educational data schema.

In this work we propose a multi-dimensional methodology for analyzing educational data and improve prediction.

2. MULTI-DIMENSIONAL DATA MINING FOR EDUCATION

The prediction of future outcomes is a task mostly addressed by classification. However, results are far from being satisfactory, and one of the reasons may be the fact that the multi-dimensional relations between attributes are not being considered. Thus, we propose to use a multi-relational data mining (MRDM) algorithm to find patterns that are able to characterize different entities and their behaviors, and use the discovered information to enrich the data used for classification training, similar to what was proposed in [2].

MRDM [3] is an area that aims for the discovery of frequent relations that involve multiple tables, without joining all the tables before mining. Pattern mining, in particular, aims for enumerating all frequent patterns that conceptually represent relations among entities. These patterns can be *intra-dimensional* or *inter-dimensional*, if they contain items from the same or more than one dimension, respectively; or *aggregated*, if they result from the aggregation of events of the central table. The works on MRDM has increased, but they do not often scale with the number of facts. To overcome this, the algorithm *StarFP-Stream* was proposed [5], combining MRDM with data streaming techniques, and it is able to mine both large and growing star schemas.

The methodology proposed has four main steps: *multi-dimensional pattern mining*, *pattern filtering*, *data enrichment* and *classification*. The first consists on running an algorithm for multi-dimensional pattern mining over each star schema. After finding all the patterns, the next step is to filter the inter-dimensional and aggregate ones and choose the N best. We define a set of filters that try to capture the interestingness of a pattern: (1) *support* – The higher the support, the more events share the same characteristics represented in this pattern. However, the smaller the relations modeled; (2) *size* – The largest patterns model more relations than smaller ones. However, they tend to have the smallest supports; (3) *closed* patterns – A pattern is closed if none of its immediate supersets have the same support. Thus, a set of closed patterns (non-redundant) is more likely to be more interesting. (4) *rough independence* – If two events are independent, the occurrences of one do not influence the probability of the other, and therefore they are not interesting. Thus, $RInd(\{A_{1..n}\}) = \frac{P(A_1 \cap A_2 \cap ... \cap A_n)}{P(A_1)P(A_2)...P(A_n)}$. (5) *rough chi-square* – Chi^2 evaluates the correlation between variables. And the more correlated, the more interesting are the relations: $RChi^2(\{A_{1..n}\}) = \frac{(support(A_1 \cap ... \cap A_n) - P(A_1)...P(A_n))^2}{P(A_1)...P(A_n)}$.

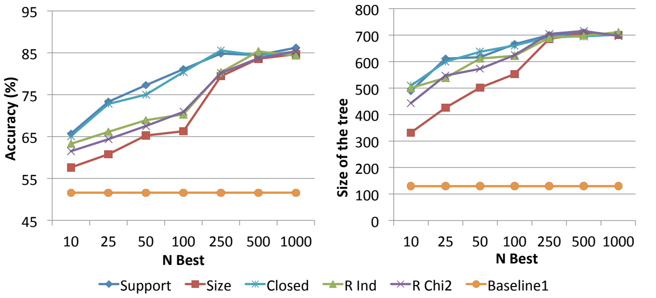


Figure 2: Accuracy and size of the model for B1.

Once we have the best patterns, we can use them as features for classification training by extending individual records with the multi-dimensional patterns, represented as boolean attributes (true or false) whenever an entity satisfies (or not) the particular pattern. We can then finally run classification algorithms on these enriched data and observe the results.

3. AN EDUCATIONAL CASE STUDY

In this case study we used the data from the *Information Systems and Computer Engineering* program, offered in *Instituto Superior Técnico – Universidade de Lisboa*, in Portugal. From the data warehouse, we have chosen the 2 stars in Figure 1, modeling student performances in their enrollments and teacher evaluation for their lectures. Our main goal is to test our multi-dimensional methodology for predicting student results on the 10 most representative courses of more advanced years (3rd-5th), based on the frequent behaviors found in the first 2 years. There were more than 650 students enrolled in some of those courses and 36 teachers lecturing them. There were 1830 enrollments to predict. We tested our enriched data with two baselines (without patterns). The first (B1) consists in the joining of the student, course and teacher dimensions, plus the student average grade, and the second (B2) contains also the specific grades on the most representative courses of the 1st and 2nd years (23 courses). Student grades were categorized and classification results are the average of several 10-cross fold validations, given by *C4.5* (available in *Weka*).

For finding student behaviors, there were more than 17 thousand enrollments that were used for pattern mining. We used an implementation of *StarFP-Stream* [5] made in Java (JVM version 1.6.0 37), and data in the fact table were aggregated per each pair student–term, so that we could find frequent sets of courses attended per term. We found, e.g. that it is frequent to succeed to both SIBD, PLD, AM3 and AN in the same term, and to fail to AN course in the 2nd season. For finding teacher behaviors, we used the surveys of the courses we were predicting, in previous years (1088 survey questions). Data in this star schema was aggregated per survey id, in order to find frequent sets of evaluations given by students to their teachers.

Figures 2 and 3 (left) show the accuracy of the classification step, over the B1 and B2, and corresponding datasets enriched with patterns from student behaviors (i.e. patterns of *Enrollments Star*). As expected, since B2 has more information about the background of the student, it achieves better accuracy than B1 (a 35% improvement). It is interesting to see that we can predict 50% of the grades of students based solely on their characteristics and average grade from years 1 and 2 (B1). When we add the patterns, we can see that the accuracy improves in both cases. In B1, the improvement is huge, of about 35%, because we are adding the behavior information about students, that was not present before. In B2, it allows classification to achieve an accuracy of 90%. Although only 4%, this improvement indicates that patterns are chosen instead of specific courses, and this may result in models with less over fitting, and therefore more accurate when predicting new instances. Also, results show that the more N best patterns are chosen, the better the accuracy, in general. When analyzing the different filters, both the *size* and *closed* filters achieved better results.

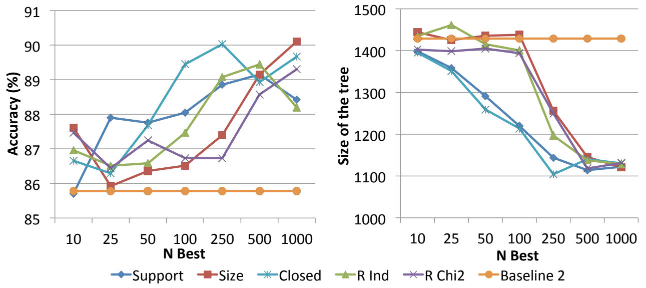


Figure 3: Accuracy and size of the model for B2.

Figures 2 and 3 (right) analyze the size of the trees created by the classifier (i.e. the size of the model). We can see that for B2, also as expected, the trees resulting from classifying the enriched datasets are smaller than the base tree. In the B1 case, the models of the enriched datasets are larger because the baseline does not have much information, and when we add patterns, they are chosen for building the tree. Nevertheless, for similar values of accuracy (85%), the tree for B1 is much smaller than for B2.

4. CONCLUSIONS

In this paper we proposed a multi-dimensional methodology for mining educational data. It is general, and may be applied to different domains and with different algorithms. Experiments on a real case study show that we can take into account the multi-dimensionality of the educational data to discover frequent behaviors, and also to improve prediction. This work is partially supported by FCT – Fundação para a Ciência e a Tecnologia, under project educare (PTDC/EIA-EIA/110058/2009) and PhD grant SFRH/BD/ 64108/2009.

5. REFERENCES

[1] R. Baker, T. Barnes, and B. J. Educational data mining 2008. In *EDM 2008: Proc. of 1st Intern. Conf. on Educational Data Mining*, page 2, 2008.

[2] J. Barracosa and C. Antunes. Anticipating teachers performance. In *Proc. of Int. W. on Knowl. Discovery on Educational Data (KDDinED@KDD)*. ACM, 2011.

[3] S. Džeroski. Multi-relational data mining: an introduction. *SIGKDD Explor. Newsl.*, 5(1):1–16, 2003.

[4] C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, 6(40):601–618, 2010.

[5] A. Silva and C. Antunes. Finding patterns in large star schemas at the right aggregation level. In *Proc. of the 9th Intern. Conf. on Modeling Decisions for AI*, pages 329–340. Springer, 2012.

Mining Reading Comprehension Within Educational Objective Frameworks

Terry Peckham
Department of Computer Science
University of Saskatchewan
110 Science Place
Saskatoon, SK., Canada
tep578@mail.usask.ca

Gordon McCalla
Department of Computer Science
University of Saskatchewan
110 Science Place
Saskatoon, SK., Canada
mccalla@cs.usask.ca

ABSTRACT

In this paper we explore patterns in student behavior as they answer questions about documents they are reading. In earlier work [4] we showed that as students answer a question online, they can be categorized into one of 4 different clusters of "reading-scanning-scrolling" behaviors. Further, their reading-scanning-scrolling behavior category predicts the quality of their answer to that particular question based on the level of that question in Bloom's Taxonomy. We have performed a second experiment that confirms these earlier results. In a third exploratory experiment we also show how the reading-scanning-scrolling clusters already discovered can be refined for use with another taxonomy, the Marzano Taxonomy. We are currently exploring whether other clusters can be found to help understand student behavior in terms of the Marzano Taxonomy.

Keywords

K-means clustering, Bloom's Taxonomy, Marzano's Taxonomy

1. INTRODUCTION

Educational objective taxonomies form a pedagogical framework for understanding student learning. Within the classroom environment, these taxonomies are utilized to challenge the teachers and instructors to move beyond simple low level learning.

Bloom's Taxonomy of Educational Objectives and its subsequent revision by Anderson [1] is a widely used taxonomy within the classroom. It is comprised of three major domains, the cognitive, affective and psychomotor. The cognitive domain is comprised of six hierarchical categories ranged from the easiest cognitive tasks to the most difficult cognitive tasks. The categories, from lowest to highest, are knowledge, comprehension, application, analysis, synthesis and evaluation (as revised by Anderson et. al. [1]).

In response to shortfalls found within Bloom's Taxonomy [2], Marzano and Kendall [3] in 2007 introduced their taxonomy of educational objectives. Marzano's premise is that knowledge use is affected by three systems: the cognitive system, the metacognitive system and the self-system [3]. When an individual is faced with some new situation, the self-system must determine if it is better to continue with the current behavior or to adapt some new behavior. The metacognitive system then tries to set the goals that are needed to achieve the desired outcome and then monitor those goals. The cognitive system processes all the necessary information required to complete the task that is obtained from the knowledge system [3].

Each of Bloom's categories for the cognitive domain can map over to one of the categories for Marzano's cognitive domain. However, there is no one-to-one mapping possible between these domains [2]. So in practice we will find that a problem categorized as Bloom level 2 (understanding) may equate to Marzano's level 2 (comprehension) or to Marzano's level 1 (knowledge) depending on the context of the problem.

This paper extends our earlier work [4]. In particular, we wanted to confirm the results of our first experiment. Additionally, we wanted to see if we could move from the Bloom taxonomy to the Marzano taxonomy, and whether this would lead to a more refined predictive capability.

2. METHODOLOGY AND RESULTS

Our initial and confirmatory experiment was performed to determine if there were useful patterns of student usage that could be found within a simple learning content management system [4]. Students were given multiple documents that contained novel information and then were asked multiple questions to determine how they had learned the material presented. The students were allowed to freely move between the various articles and questions presented to them and could freely interact with the content they were expected to learn. Following the trace methodological approach, all of the interactions/events in the system were captured and time-stamped. The events captured included mouse clicks, mouse wheel movements, button clicks, typing, and so on. Over the two experiments, a total of 50 participants were tested generating over 63,738 events.

Based on the timestamps of these events, we were able to measure when students were reading (slowest), scanning, or scrolling (fastest) through the document. The time cutoffs used to differentiate between the reading, scanning and scrolling categories were consistent with other document navigation literature,, as discussed in [4]. In the first experiment we found no significant differences between the clusters until the level of knowledge needed to answer a question in terms of Bloom's Taxonomy was factored in [4]. Then, over many k-mean clustering iterations we discovered 6 clusters that allowed us to predict the quality of the students' answers to questions based on their Bloom level, with the 4 most predictive as follows: Light Reading Cluster (50% reading, 30% scanning, 20% scrolling) (50:30:20), Light Medium Reading Cluster (60:30:10), Heavy Medium Reading Cluster (70:30:10), Heavy Reading Cluster (80:10:10). In experiment 2, we used the clusters found in [4] predictively as metrics and checked to see if we would still obtain significant differences between the clusters.

Table 1 shows that for experiment 2 all of the levels tested have significant differences. This shows that the clusters from experiment 1 hold up well in predicting students' answers to questions in experiment 2, thus confirming the results of the first experiment.

Bloom Level	F	P	F-Critical
1	23.137	1.04E-6	3.09
2	33.245	2.47E-7	3.19
3	21.237	.005796	6.60
4	50.535	.000854	6.60
5	25.128	1.18E-6	3.15

Table 1 One way ANOVA for Bloom Level Experiment 2

Again as in[4], the clustering does not predict an exact grade on a question but provides a more coarse grained prediction of a student's performance. For example, question 2, experiment 2 asked for a student to recollect two pieces of information. The heavy reading cluster almost always involved the student achieving a failing grade while those students who performed more scanning obtaining a grade greater than 75%. Those students who performed more scanning and who did not receive higher grades did so because they misinterpreted the question.

	50,30,20	60,30,10	70,20,10	80,10,10
50,30,20	-	0.19626	0.15202	0.13407
60,30,10	0.25348*	-	0.1896	0.17554
70,20,10	0.3588*	0.10529	-	0.12412
80,10,10	0.4651*	0.21159*	0.1063	-

Table 2 Tukey-Kramer Analysis Bloom Level 2 Experiment 2

Table 2 demonstrates the differences between the clusters for experiment 2. Again we see that there are significant differences but those differences tend to be between the 50:30:20 and the 80:10:10 clusters. In the second experiment the participants consisted primarily of individuals that are heavy computer users. This contrasts with the participants in experiment 1 that were primarily novice computer users. The participants from the second experiment tended to either perform heavy reading or the other extreme with the highest scanning and scrolling ratios. The middle two clusters were under-represented in the second experiment. The reason may be that the more advanced computer users have found strategies which allow for successful information processing in online environments.

Recently, Marzano's Taxonomy [3] has become popular, partly because it has finer grained sub-categories. This left us wondering if we could make predictions using Marzano's Taxonomy similar to those we did using Bloom. We decided to look at this in a third exploratory experiment where we recast the data from the first two experiments in terms of Marzano's categorizations..

To this end, questions used in experiment 1 and experiment 2 were re-categorized in terms of Marzano's Taxonomy. .Since in Marzano the cognitive domain only contains 4 main levels, there was a slight generalization from Bloom to Marzano. Table 3 shows that statistically significant predictions could be made about students' performance on questions at the first 3 levels of Marzano using the questions from experiment 2 Level 4 of Marzano did not show up as statistically significant. As with the first experiment, there weren't sufficient numbers of students to obtain significant values. However, when we combined the questions from both experiments 1 and 2, we can even make significant predictions at Marzano's level 4 (F = 43.86, F-Critical = 3.00, p = 6.77E-10).

Marzano's cognitive domain contains 4 main levels that can, in turn, be subdivided into 14 sublevels (see Table 3). These sublevels offer a much more fine-grained level of detail compared to Bloom. We found that our questions from the earlier experiments covered 8 of the 14 subcategories of Marzano's Taxonomy. In particular, all three sublevels within Marzano level 1 were represented. We could predict the quality of student answers to sublevel 1 questions with statistical significance, but could not do so with the other two sublevels of Marzano level 1. Nevertheless, this does give hope that we can make predictions at this more fine-grained levels offered by the Marzano Taxonomy, although likely larger studies will be needed, with more students, to discover the relevant clusters.

Marzano Level	F	F-Critical
1 (Sublevel 1, 2, 3)	120.98	2.73
2 (1, 2)	62.31	3.07
3 (1, 2, 3, 4, 5)	52.71	3.91
4 (1, 2, 3, 4)	0.60	3.58
All Levels Combined	1.40	2.67

Table 3 Tukey-Kramer Analysis Marzano Experiment 2

CONCLUSIONS

Our three experiments lead to the following general conclusions. First, the patterns discovered in the first experiment seem to hold well for the second experiment. This provides more confidence that they actually represent real behavioral differences, and that it is worthwhile to look at student activities in terms of the Bloom level of the tasks they are trying to accomplish. Second, the patterns to some degree survived when the questions were relabeled in terms of the Marzano taxonomy. This points out that there are strong correspondences between Bloom and Marzano, and even opens up the idea of perhaps formally exploring these correspondences in future through mining actual student behavior as they solve problems at various levels of the two taxonomies. Third, the promise of Marzano's taxonomy, with its more refined categorizations, to explain why the reading-scanning-scrolling behaviors lead to the various outcomes that they do has yet to be fully validated

REFERENCES

[1] Anderson, L.W., Krathwohl, D.R. and Bloom, B.S., (Eds), 2000, Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. New York: Longman.

[2] Aworuwa, B. & Nkoge, B. (2007). The New Taxonomy of Educational Objectives and Implications for Designing Instruction for Distance Learning Delivery. *In T. Bastiaens & S. Carliner (Eds.), E-Learn* 2007 (pp. 1394-1398).

[3] Marzano, R. J., & Kendall, J. S. (Eds.). (2007). *The new taxonomy of educational objectives*. Corwin Press.

[4] Peckham, T., & McCalla, G. (2012). Mining Student Behavior Patterns in Reading Comprehension Tasks. *Proc. EDM 2012, Crete, Greece (pp. 87-94).*

Mining students' strategies to enable collaborative learning

Sergio Gutierrez-Santos
London Knowledge Lab
Birkbeck,
University of London, UK
sergut@dcs.bbk.ac.uk

Manolis Mavrikis
London Knowledge Lab
Institute of Education,
University of London, UK
m.mavrikis@lkl.ac.uk

Alex Poulovassilis
London Knowledge Lab
Birkbeck,
University of London, UK
ap@dcs.bbk.ac.uk

ABSTRACT

Despite the benefits that collaborative discussion has on learning, one difficult problem is the formation of pairs or groups that enable appropriate discussion. This problem is even more challenging in the case of unstructured interaction with exploratory learning environments. Building on previous work on supporting individual learners in such environments, this paper reports on a tool that generates groups of students by mining what they have done in the context of an exploratory activity and then calculating similarities between their strategies.

1. INTRODUCTION

Exploratory Learning Environments (ELEs) are educational applications that provide direct access to a domain or to some alternative representation and offer a context and appropriate tools to scaffold the learning experience. They are generally aligned with theories of learning that emphasise the role of learners in constructing their own learning. In parallel to their recent upsurge, there has been a lot of work in the field of Computer-Supported Collaborative Learning (CSCL) with technological advances that are making collaborative problem solving and co-construction of knowledge possible even for remote participants. Research in both areas (ELEs and CSCL) has demonstrated that working in groups has the potential to enhance learning, but that careful planning and structuring of collaborative tasks and strategic formation of collaboration groups is a necessary prerequisite [1, 4].

Although the advantages of encouraging students to examine different approaches to a problem, discuss the benefits and drawbacks of each, build on each other's ideas, and benefit from the reflection that results from interaction with others, have been widely identified [4, 5]), it can be difficult to form *potentially productive* groups i.e. groups that will provide opportunities for students to engage in fruitful discussions, enabling them reflect on their approaches to the problem, to justify and critique their solutions, and thus lead to deeper learning. This is even more so in the case of courses with a very large number of students such as Massive Open Online Courses (MOOCs), where it is infeasible for humans to participate in the creation of the groups and any effectiveness beyond haphazard pairing must be the result of analyzing students' work. Once again, exploratory activities can offer more opportunities due to their richer interaction possibilities.

2. FORMING GROUPS BASED ON EXPLORATORY LEARNING STRATEGIES

This paper reports on a tool that aims at helping to overcome these challenge. Our tool generates groups of students by mining what they have done in an exploratory activity and then calculating similarities between their strategies. The aim is to alleviate teachers/lecturers from the task of grouping students into meaningful pairs; by *meaningful* we mean pairs that maximise the probability that students will have complementary approaches or strategies to a given task or tasks, and therefore will have more opportunities for discussion, reflection, and ultimately learning. Building on former work aimed at supporting individual students [2], this tool was first created in the context of the eXpresser microworld for the learning of algebra [3] but the general principles are valid for any exploratory learning environment that is intended to be used with very large groups. Although we omit the details of the original microworld for the sake of space, it suffices to say that the microworld allows students to create pictorial tile patterns in a square grid, that patterns can be created in many different ways, and that they are used as a scaffold towards different kinds of algebraic and pseudo-algebraic formulations of mathematical problems, thus helping young learners to strengthen their algebraic generalisation skills.

Figure 1 shows three different ways of creating the same pattern and how the same algebraic formula can be expressed in different ways. Typically, in the context of a module several tasks will be tried; for any given task, students will find one solution from the set of possible solutions (advanced students may find several) and its corresponding formula. In a classroom scenario, these individual tasks are usually followed by a collaborative task in which pairs of students must explain to each other how "their" solution is the "right" one and whether their two solutions are equivalent.

It is evident that, in order for this discussion to be meaningful, students in each pair must have found solutions that are quite different; this maximises the cognitive conflict and requires deeper reflection to see the equivalences. Unfortunately, differences in real students' approaches are rarely as evident as in the three paradigmatic solutions shown in Figure 1; even in classrooms with relatively low numbers of students, teachers find themselves in a situation in which they do not have the time to make groups effectively, taking into account all details, and they resort to haphazard cre-

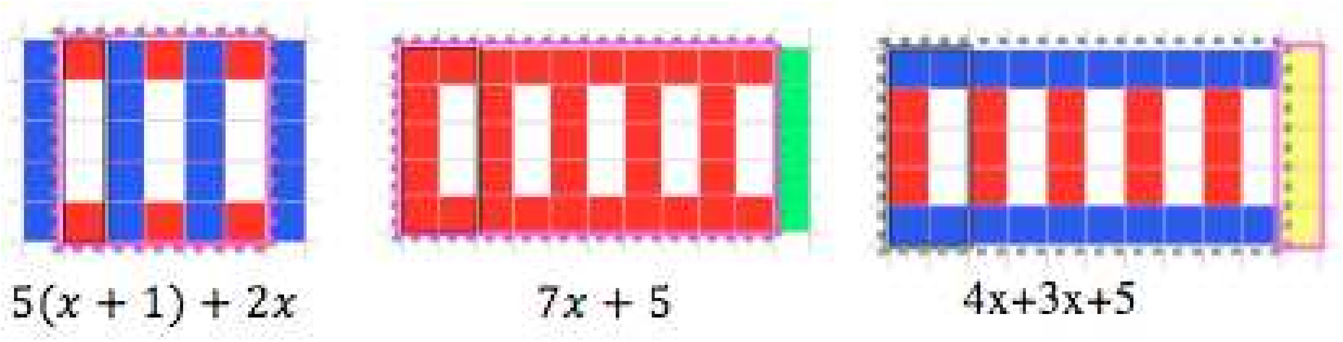


Figure 1: Example of different task solutions. Different constructions of the pattern lead to different (but equivalent) expressions.

ation of groups (e.g. based on the order of task completion or based on student choice). Our tool, on the other hand, analyses the students' actions and then suggests pairs that minimise the similarity among the approaches taken by the two students in each group.

Different strategies. In the first stages of the design of this grouping tool we tried to clarify the limits of the task, namely what were the characteristics of the best group and the worst group in our context. Although it is obviously hard to reach an agreement about these general ideas, all teachers and educators agreed that grouping together two students who have created exactly the same construction (i.e. used the same approach for the task) would not lead to much discussion as there is nothing to compare. Therefore, the first step was the determination of the definition of equality of two constructions. In collaboration with the pedagogical team, we agreed on the following definition: "Two constructions are equal from the point of view of collaborative discussion if they have the same number of patterns, the patterns have the same building blocks, the building blocks are displaced horizontally and vertically by the same amount on each iteration, and any expressions used in their attributes are related using variables in the same way".

The value of starting the design process by defining equality between exploratory strategies is twofold. First, the definition allows us to know when two students should not be put together in the same group. More importantly, it also clarifies the factors that determine when two constructions are different (and how). Our tool represents each student's strategy as the combination of three vectors in three different spaces with different metrics: building block related, numerical, and relationship. Then the overall similarity, s, is calculated as a linear combination of the inverse of the distances between the vectors of one student's strategy and those of the other:

$$s = K \times \left(w_{bb} \cdot \frac{1}{1 + bbd} + w_n \cdot \frac{1}{1 + nd} + w_r \cdot \frac{1}{1 + rd} \right)$$

where bbd, nd, and rd are the total building block, numerical, and relationship distances between pairs of patterns in the two constructions, and the w_x are weights. K is a scale factor related to the number of patterns.

Fine-tuning with experts. Weights w_{bb}, w_n, and w_r were initially set to 0.4, 0.3, and 0.3, following discussion with teachers, but were later modified and fine-tuned to ensure that the calculations made by the tool were in line with the perceptions of teachers about similarity between different students' constructions. We evaluated the validity of the suggestions of the tool by a process of gold-standard validation. This consisted of an iterative process in which our team of pedagogy experts were presented with several scenarios, each of them containing different microworld constructions, and the experts were asked to assess their similarity. At the end of this process, the average agreement between the tool's recommendation and the experts' was higher than 80%.

Acknowledgements. The eXpresser microworld and initial prototypes of the grouping tool were created in the MiGen project (migen.org) funded by the ESRC-EPSRC TEL Programme. The work here has been co-funded by the EU under the FP7 projects iTalk2Learn (#318051) and M C Squared (#610467).

3. REFERENCES

[1] P. Blatchford, P. Kutnick, E. Baines, and M. Galton. Toward a social pedagogy of classroom group work. *International Journal of Educational Research*, 39(1-2):153–172, 2003.

[2] S. Gutierrez-Santos, M. Cocea, and G. Magoulas. A case-based reasoning approach to provide adaptive feedback in microworlds. In *Intelligent Tutoring Systems*, pages 330–333, 2010.

[3] R. Noss, A. Poulovassilis, E. G. an Sergio Gutierrez-Santos, C. Hoyles, K. Kahn, G. D. Magoulas, and M. Mavrikis. The design of a system to support exploratory learning of algebraic generalisation. *Computers and Education*, 59(1):63–81, August 2012.

[4] M. Swan. *Collaborative Learning in Mathematics: A Challenge to Our Beliefs and Practices*. National Institute for Advanced and Continuing Education (NIACE, London, 2006.

[5] P. Vahey, D. Tatar, and J. Roschelle. Using handheld technology to move between private and public interactions in the classroom. In *Ubiquitous computing in education: Invisible technology, visible impact*, pages 187–210. Lawrence Erlbaum, 2007.

Modeling Student Socioaffective Responses to Group Interactions in a Collaborative Online Chat Environment

Whitney Cade, Nia Dowell,
Art Graesser
Department of Psychology
Institute for Intelligent Systems
University of Memphis
365 Innovation Drive
Memphis, TN 38152
+1 901-678-5102
{wlcade, ndowell, a-graesser}
@memphis.edu

Yla Tausczik
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
ylataus@cs.cmu.edu

James Pennebaker
Department of Psychology
University of Texas
1 University Station
Austin, TX 78712
+1 512-232-2781
pennebaker@mail.utexas.edu

ABSTRACT

Being able to monitor collaborative learning environments using unobtrusive measures is crucial to maximizing students' socioaffective experiences with a system. This analysis uses the cohesion of student responses to model students' feelings of power and connectedness to the group, two factors which emerge from a principal component analysis of a motivational survey.

Keywords

CSCL, educational group chat, group dynamics, power, connectedness, cohesion, Coh-Metrix, learning

1. INTRODUCTION

Understanding the dynamics of computer-supported collaborative learning (CSCL) environments is crucial to providing adaptive enhancements or supports to groups of students who are not receiving the full benefits of technology-based collaboration [5]. One important facet of any learning environment is the student's affect during the interaction, which may have a positive impact on motivation [4] or may lead to tension and competition if the group is experiencing negative emotions due to conflict [1]. Previous work on group dynamics and its impact on learning has made a sharp division between the social and informational processing parts of group discussion [6, 9], but in collaborative learning environments, these aspects may be difficult to tease apart, as there may be a cognitive component to social side of CSCL and vice versa. This may be particularly true when examining cues which assess the group's socioaffective state without interrupting the flow of conversation to ask for a self-report. Linguistic cues of a cognitive nature may be able to detect various socioaffective components of CSCL conversations, with the added advantages of being performed automatically and covertly based on the flow of conversation.

This work uses Coh-Metrix [7] to assess how discoursive deep cohesion predicts socioaffective components found in a motivational survey [8] administered to students engaged in a group chat environment. Deep cohesion is defined as the extent to which ideas in the text are cohesively connected at a deeper conceptual level that signifies causality or intentionality. Therefore, deep cohesion may be one way of exploring how cognitive aspects of language predict socioaffective outcomes in group conversation. By understanding group dynamics in CSCL environments, we may be able to intervene where group conversation stagnates or goes awry to maximize the learning experience.

2. METHODS

Seven hundred forty-eight students in two introductory-level psychology classes at the University of Texas at Austin used an online educational platform to chat with group members about assigned readings. Once logged on, students were randomly placed in groups of up to five members, given a 10-item pretest about the readings, then allowed to chat for exactly 20 minutes about the readings. Discussion questions were given to groups to facilitate discussion, but no restrictions were placed on what could be said. After the chat session, students were given a 10 item posttest and filled out a motivation questionnaire which asked about the students' perceptions of the interaction, group members, and their own role in the group. More details about this survey are given by Niederhoffer and Pennebaker [8]. All data was logged for analysis, then cleaned, parsed, and extracted from these logs. The chat contributions of each individual were processed using Coh-Metrix and then Winsorized.

3. RESULTS AND DISCUSSION

The first set of analyses conducted sought to find out the relationship between perceptions about the group interaction and learning. We conducted a principal component analysis (PCA) to create meaningful, broader variables with which to describe the students' socioaffective experience in the group. The data fit all the standard criteria for factorability (all variables intercorrelated with one of variable above .3, a Kaiser-Meyer-Olkin measure of sampling adequecy above .6, and a significant Bartlett's test of sphericity ($\chi^2(21) = 1640.31$, $p < .001$). Two components with eigenvalues greater than 1 were found, which collectively explained 66.6% of the variance. All items from the test loaded strongly onto only one component (>.4) with low cross-loadings (<.3). The items that loaded onto the first component was concerned with the student's perceptions of how well the chat went and how much the student "clicked" with the other members; this component was therefore labeled "Connectedness". The second component was composed of items about the social status of the student and the control they exerted over the group, and thus has been labeled "Power." Feelings of power and connectedness have both been linked to qualitatively and quantitatively better performance in collaborative learning environments [2, 3]. These components were then correlated with each students' proportionalized learning gains ([Posttest - Pretest] / [1 - Pretest]). Connectedness was significantly correlated with learning $r(743) = .164$, $p < .001$, but power was not, $r(743) = .007$, $p > .05$. McGrath [6] has posited that a positive social relationship leads to better group performance, so connectedness and learning ought to be somewhat linked even in a single discussion session, but authority has also been linked to learning [3], which was not found here. However, it is possible that power component, which is based on self-reports, is not sensitive enough to pick up this relationship in a single learning session.

The second set of analyses examined how the linguistic cue deep cohesion predicted feelings of power and connectedness to the group by using mixed-effects modeling. Mixed-effects modeling was used to account for the nested structure of the data, where students are embedded within group. This random factor can therefore be controlled for in mixed-effects modeling while measuring the effects of the fixed factors. Two models were constructed for these analyses: one to examine deep cohesion's ability to predict power and one to predict connectedness. Deep cohesion was the independent variable, while student (748 levels) nested within group (183 levels) was the random factor. Deep cohesion was found to positively and significantly predict feelings of connectedness to the group, $F(1, 744.137) = 12.25$, $SE = .021$, $p < .001$, so that as a student felt more connected to the group, the deep cohesion in their language increased. The same was also true for predicting feelings of power in the group, $F(1, 746) = 11.909$, $SE = .021$, $p = .001$; as a student's feelings of power in the group increased, so did the deep cohesion in their language. This demonstrates not only that linguistic cues are a viable source of predicting socioaffective outcomes, but that the cues for detecting such outcomes need not be restricted to typical emotive cues; a person's feelings about their group experience may also emerge in the cohesion of their language, a subtler cue than, for instance, their use of emotive language.

These analyses demonstrate that cognitive linguistic cues may be of use in detecting students' socioaffective attitudes towards fellow students in CSCL environments, which may have long-term consequences for their motivation and continued use of such systems. Being able to covertly detect these attitudes may mean that interventions are possible.

4. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (REC 0106965, ITR 0325428, HCC 0834847) and (BCS 0904909, DRK-12-0918409), the Institute of Education Sciences (R305G020018, R305A080589), The Gates Foundation, U.S. Department of Homeland Security (Z934002/UTAA08-063), and the Army Research Institute (W5J9CQ12C0043). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

5. REFERENCES

[1] Darnon, C., Muller, D., Schrager, S., Pannuzzo, N., and Butera, F. 2006. Mastery and performance goals predict epistemic and relational conflict regulation. *J. Educ. Psychol*. 98, 4 (Nov. 2006), 766-776. DOI= http://dx.doi.org/10.1037/0022-0663.98.4.766.

[2] Eales, R. T. J., Hall, T., and Bannon, L. J. 2002. The motivation is the message: Comparing CSCL in different settings. In *Proceedings of Computer-Supported Collaborative Learning* (Boulder, Colorado, 2002). CSCL '02. International Society of the Learning Sciences, 310–317.

[3] Howley, I., Mayfield, E., and Rosé, C. P. 2011. Missing something? Authority in collaborative learning. In *Proceedings of the 9th International Computer Supported Collaborative Learning Conference, Volume 1: Long Papers* (Hong Kong, China, July. 04 - 08, 2011). CSCL '11. International Society of the Learning Sciences, 336-373.

[4] Keller, J. M. 1987. Strategies for stimulating the motivation to learn. *Performance and Instruction*, 26, 8 (Oct. 1987), 1-7. DOI= http://dx.doi.org/10.1002/pfi.4160260802.

[5] Lou, Y., Abrami, P. C., and d'Apollonia, S. 2001. Small group and individual learning with technology: A meta-analysis. *Rev. Educ. Res*. 71, 3 (Fall 2001), 449-521. DOI= http://dx.doi.org/10.3102/00346543071003449.

[6] McGrath, J. E. 1997. Small group research, that once and future field: An interpretation of the past with an eye to the future. *Group Dyn.-Theory Res*. 1, 1 (Mar. 1997), 7-27. DOI= http://dx.doi.org/10.1037/1089-2699.1.1.7.

[7] McNamara, D. S. and Graesser, A. C. 2012. Coh-Metrix: An automated tool for theoretical and applied natural language processing. In *Applied natural language processing: Identification, investigation, and resolution*, P. M. McCarthy and C. Boonthum, Eds. IGI Global, Hershey, Pennsylvania. DOI= http://dx.doi.org/10.4018/978-1-60960-741-8.ch011.

[8] Niederhoffer, K. G. and Pennebaker, J. W. 2002. Linguistic style matching in social interaction. *J. Lang. Soc. Psychol.* 21, 4 (Dec. 2002), 337-360. DOI= http://dx.doi.org/10.1177/026192702237953.

[9] Tausczik, Y. R. and Pennebaker, J. W. 2013. Improving teamwork using real-time language feedback. In *Proceedings of Human Factors in Computing Systems* (Paris, France, April 27 – 2 May, 2013). CHI '13. ACM, New York, New York, 459-468. DOI= http://doi.acm.org/10.1145/2470654.2470720.

Now We're Talkin':
Leveraging the Power of Natural Language Processing to Inform ITS Development

Laura K. Allen
Tempe, AZ, USA
Arizona State University
LauraKAllen@asu.edu

Erica L. Snow
Tempe, AZ, USA
Arizona State University
Erica.L.Snow@asu.edu

Danielle S. McNamara
Tempe, AZ, USA
Arizona State University
Danielle.McNamara@asu.edu

ABSTRACT

In the current study, we utilize natural language processing techniques to examine relations between the linguistic properties of students' self-explanations and their reading comprehension skills. Linguistic features of students' aggregated self-explanations were analyzed using the Linguistic Inquiry and Word Count (LIWC) software. Results indicated that linguistic properties of self-explanations were predictive of reading comprehension ability. The results suggest that natural language processing techniques can serve as stealth assessments of abilities within intelligent tutoring systems.

Keywords

Intelligent Tutoring Systems, natural language processing, stealth assessment, student modeling, reading comprehension

1. INTRODUCTION

In the field of intelligent tutoring systems (ITSs), there exists some debate to determine when it is most optimal to assess students' performance, skills, and affect during learning tasks. System developers aim to avoid repeatedly questioning and testing students, as it may disrupt their learning flow [1]. However, it is crucial to gather student information because such variables can affect the adaptability and sophistication of these systems. One way to collect student information (without directly testing students) is through the use of stealth assessments [1]. A stealth assessment is a measure of student information (e.g., engagement, affect, skills, etc.) that is embedded within a particular task and seemingly "invisible" to users [2].

Stealth assessments can serve to inform student models within adaptive environments and, accordingly, improve system feedback and instruction. By modeling the behavioral and cognitive states of students without explicit surveys or tests, ITSs can improve student models without disrupting the learning flow of the users. This information can then be used to guide the pedagogical content that is presented to each student [3].

1.2 iSTART

The Interactive Strategy Training for Active Reading and Thinking (iSTART) tutor is an ITS that was developed to teach reading comprehension strategies to high school and college students [4]. The primary focus of the system is on the strategy of self-explanation, which has been shown to benefit students on a number of higher-level tasks [5]. Within this ITS, there are introduction, demonstration, and practice modules that explain the purpose and demonstrate the use of these strategies.

2. STUDY

The goal of the current study is to examine the extent to which the linguistic and semantic properties of students' natural language input can be used as a stealth assessment of their reading comprehension skills. To accomplish these goals, we collected students' self-explanations from the iSTART system and aggregated the individual, sentence-level self-explanations across each text that was read. Students' aggregated self-explanations were then analyzed using the Linguistic Inquiry and Word Count (LIWC) software. We utilized this tool in the current study so that we could investigate relations between students' reading comprehension ability and the semantic properties of their natural language input.

Participants were 126 high-school students from a mid-south urban environment who participated in iSTART training. Students' reading comprehension skills were measured using the Gates-MacGinitie (4th ed.) reading skill test (form S) level 10/12.

2.1 Text Analyses

The linguistic features of students' aggregated self-explanations were calculated using LIWC. LIWC is a text analysis tool that uses categorical word dictionaries to provide information about texts that corresponds to thematic and rhetorical language use [6].

To extract linguistic and semantic information from students' self-explanations, individual (sentence-level) self-explanations were combined for each text read during training. Thus, each student was left with one aggregated self-explanation file for each text that they read during their time in the iSTART system. This aggregation method is discussed in greater detail in previously published work [7].

LIWC indices were then calculated for each of the aggregated self-explanation files. For each student, this LIWC output was averaged across texts to create an average score on each of the linguistic measures. These scores provide a measure of students' aggregated self-explanations at multiple linguistic levels.

3. RESULTS

To examine the relations between the LIWC linguistic scores and students' reading comprehension performance, a correlation was calculated between students' reading comprehension scores and their LIWC scores. A stepwise regression model was then calculated to assess which properties were most predictive of students' comprehension skills. A training and test set approach was used for both regressions (67% for the training set and 33% for the test set) to validate the analyses.

There were 13 LIWC variables that significantly correlated with reading comprehension scores (see Table 1). We tested for multi-collinearity among these variables; however, no variables were correlated with each other above *r* = .90.

Table 1. Correlations between reading comprehension scores and LIWC linguistic scores

LIWC variable/category	*r*	*p*
Word count	.350	<.001
Words per sentence	-.316	<.001
Number words	.266	<.001
Past words	.224	<.010
Certainty words	.222	<.050
Filler words	-.212	<.050
Second person pronouns	-.211	<.050
Quantitative words	.201	<.050
Third person pronouns	.197	>.050
Ingestion words	-.195	>.050
Home words	.194	>.050
Social words	-.182	>.050
Vision words	.177	>.050

A stepwise regression analysis was conducted on the 90 self-explanations files with the 13 LIWC variables as predictors of reading scores (see Table 2) and yielded a significant model, *F*(4, 85) = 9.865, *p* < .001, *r* = .563, R^2 = .317 with four predictors: word count [β =.38, t(4, 85)=4.383, *p* < .001], words per sentence [β =-.29, t(4, 85)=-3.129, *p* = .002], second person pronouns [β =-.24, t(4, 85)=-2.58, *p* = .012], and ingestion words [β =-.22, t(4, 85)=-2.393, *p* = .019]. The test set yielded *r* = .490, R^2 = .240.

Table 2. LIWC regression analysis prediction comprehension scores

Entry	Variable added	R^2	Δ R^2
Entry 1	Word count	.120	.120
Entry 2	Words per sentence	.228	.090
Entry 3	Second person pronouns	.271	.043
Entry 4	Ingestion words	.317	.046

4. DISCUSSION

We leveraged NLP to develop stealth assessments of students' reading comprehension skills. A subset of LIWC indices were related to reading comprehension scores – namely, high reading ability students were more likely to have longer self-explanations (with shorter individual sentences) with an emphasis on numbers, words related to the past, and words related to home and vision. With regards to writing style, these students self-explained more confidently (certainty words), using a greater number of third person pronouns and fewer second person pronouns. Follow-up regression analyses indicated that word count, words per sentence, second person pronouns (e.g., you), and ingestion words (e.g., dish, eat, taste) provided the most predictive power in this model, accounting for 32% of the variance. Importantly, most of these indices were basic indices, rather than semantic categories. Thus, while many semantic lexical categories were significantly related to students' comprehension scores, they provided less predictive power than basic indices. The ingestion words index was the only semantic LIWC variable that was retained in the final model. This is likely an effect of the specific content presented within the iSTART passages; perhaps better readers provided more specific, on-topic information in their self-explanations. This question will be investigated more thoroughly in future, qualitative analyses.

These results are important, as they suggest that students' abilities manifest in the way that they explain concepts in texts. Therefore, linguistic and semantic properties of self-explanations may provide crucial information about students' cognitive processes during text comprehension. Here, we only analyzed pretest reading ability. However, these methods could be applied to model a number of relevant student features, such as their affective states and prior knowledge. Overall, the results of this study (and similar studies) can be used to help researchers develop assessments and models that provide more nuanced information about students for the purpose of increasing personalized instruction and adaptability.

5. ACKNOWLEDGMENTS

This research was supported in part by: IES R305G020018-02, IES R305G040046, IES R305A080589, and NSF REC0241144, NSF IIS-0735682.

6. REFERENCES

[1] Shute, V. J. 2011. Stealth assessment in computer-based games to support learning. In S. Tobias & J. D. Fletcher (Eds.), Computer Games and Instruction. Information Age Publishers, Charlotte, NC, 503-524.

[2] Shute, V. J., Ventura, M., Bauer, M. I., and Zapata-Rivera, D. 2009. Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In U. Ritterfeld, M. Cody & P. Vorderer (Eds.), Serious games: Mechanisms and effects. Routledge, Mahwah, NJ, 295-321.

[3] Vanlehn, K. 2006. The behavior of tutoring systems. International journal of artificial intelligence in education, 16 (2006), 227-265.

[4] McNamara, D. S., Levinstein, I. B., and Boonthum, C. 2004. iSTART: Interactive strategy trainer for active reading and thinking. Behavioral Research Methods, Instruments, & Computers, 36, (2004), 222-233.

[5] McNamara, D. S. 2004. SERT: Self-explanation reading training. Discourse Processes, 38, (2004), 1-30.

[6] Pennebaker, J. W., Booth, R. J., and Francis, M. E. 2007. Linguistic Inquiry and Word Count: LIWC [computer software]. Austin, TX.

[7] Varner, L. K., Jackson, G. T., Snow, E. L., and McNamara, D. S. 2013. Does size matter? Investigating user input at a larger bandwidth. In C. Boonthum-Denecke & G. M. Youngblood (Eds.), Proceedings of the 26th Annual Florida Artificial Intelligence Research Society Conference, AAAI, St. Petersburg, FL, 546-549.

Peer assessment in the first French MOOC : Analyzing assessors' behavior

Matthieu Cisel
ENS Cachan
61 av. du Pdt Wilson
94230 Cachan
+33 1 43 37 08 52
mcisel@ens-cachan.fr

Rémi Bachelet
Ecole Centrale de Lille
Cité Scientifique,
59651 Villeneuve d'Ascq
+33 3 20 33 53 53
remi.bachelet@ec-lille.fr

Eric Bruillard
ENS Cachan
61 av. du Pdt Wilson
94230 Cachan
+33 1 47 40 24 57
bruillard@ens-cachan.fr

ABSTRACT

Given the increasing number of students registering to MOOCs for free, course instructors who want to go beyond automated evaluation have no choice but to use peer assessment. Despite the increasing use of peer evaluation, very little is known regarding the factors that influence assessors' engagement in the process. Based on two editions of *Introduction to Project Management*, the first French xMOOC, we explored the impact of learners' background on their engagement in peer assessment. We observed that registrants that took part in peer evaluation differed significantly from other participants in regards to time constraints and demographic variables such as geographical origin.

Keywords

Peer assessment, xMOOC, engagement, demography

1. INTRODUCTION

The impact of Massive Open Online Courses (MOOCs) has considerably deepened since the foundation of edX and Coursera in 2012 [3]. Nevertheless, initial enthusiasm has been tempered by recurrent criticism over different aspects of MOOCs such as their low completion rates [1] or the unreliability of the grading process. Many courses rely on peer assessment [5] to evaluate at no cost large amounts of assignments. This grading process is easily scalable, but has faced high level of skepticism given the fact that MOOC participants are not trained examiners.

A deeper understanding of the factors influencing the peer grading process is needed in order to increase its efficiency. *Introduction to Project Management* is the first French xMOOC; it relies extensively on peer assessment and therefore represents an interesting case study in regards to those issues. How does participants' background influence their engagement in the peer assessment?

2. MATERIAL AND METHODS

2.1 Course description

ABC de la Gestion de Projet (*Introduction to Project Management*) is a MOOC organized by Centrale Lille, it was run twice in 2013, in the spring and in the fall. 1332 participants completed and obtained a certificate during the spring edition, and 3301 during the fall edition. In the second edition of the course exclusively, 579 students from Centrale Lille and several other French institutions of higher education registered. They were not taken into account in this analysis. When we speak about students, we refer to registrants still studying at university but not taking the course for credentials.

The course provided videos, quizzes, weekly assignments and a final examination. Two certificates corresponding to two different workloads were offered - a basic one and an advanced one. To obtain the basic certificate, it was required to complete successfully the quizzes and to pass the final exam. In order to obtain the advanced certificate, participants were required, in addition to the quizzes and the final exam, to submit weekly assignments that were peer assessed. In the spring edition and the fall edition, respectively 438 and 809 obtained the advanced certificate. Assignments were evaluated four times each in the first edition, and five times each in the second. Consequently, over the duration of the MOOC registrants could assess up to 16 and 25 assignments in the first and the second edition, respectively. Many registrants skipped some peer assessments in the spring edition. In the fall edition, course instructors threatened to lower the grades of the participants who had not taken part in the peer assessment process.

2.2 Available data and methods

In both editions, participants were asked to fill in a survey at the beginning of the course. It was responded to by 69% of the 3495 registrants of the spring edition and by 54% of the 10847 registrants of the fall edition. Response rate was higher among completers, with 99% and 93% in the first edition and the second one, respectively. Those surveys provided data on participants' origin, gender, employment status, and the amount of time they intended to work weekly for the MOOC. Countries were classified into three categories based on their human development index (HDI), obtained from UN data [7]. Countries with Medium and High HDI were grouped into a "Intermediate HDI" category. In order to obtain odd-ratios, we computed logistic regressions (glm procedure, family="binomial") with R. *Ref.* is the reference for such odd-ratios.

3. RESULTS

In both editions, only a fraction of registrants submitted assignments and were therefore allowed to take part in the peer assessment process. Among them, a significant proportion skipped peer assessment. The proportion of participants who skipped peer assessment at least once for the assignments they had submitted was higher for the spring edition (32.7%), than for the fall edition (8.3%).

Table 1. Identifying factors affecting engagement in the peer assessment process in the spring and the fall edition of the MOOC. Numbers represent odd-ratios of a logistic regression. For "Assignment submission", higher O.R means that more participants submitted at least an assignment for a given category, compared to the reference (Ref). For skipping P.A (Peer Assessment), higher O.R means that more participants skipped peer assessment at least once. *p-value <0.05, ** p-value <0.01, ***p-value <0.001

	Assignment submission		**Skipping P.A.**	
	Spring	Fall	Spring	Fall
Gender				
Male	*Ref*	*Ref*	*Ref*	*Ref*
Female	0.97	0.88	0.74	0.90
Employment status				
Higher management positions	1.27	**1.46***	0.81	1.11
Lower management positions	0.79	0.99	0.96	1.95
Unemployed	1.23	1.06	1.30	1.26
Students	0.73	1.01	1.36	1.15
Others	*Ref*	*Ref*	*Ref*	*Ref*
HDI				
Low	*Ref*	*Ref*	*Ref*	*Ref*
Intermediate	**1.43***	1.29	1.01	1.31
Very High	**1.98 *****	**1.50*****	**0.30 *****	**0.48*****
Weekly workload				
Below 2 h	**0.31 *****	**0.43 *****	1.44	1.62
Between 2 to 4 h	*Ref*	Ref	Ref	*Ref*
Between 4 to 6 h	**3.22 *****	**2.77*****	1.03	0.91
Bbove 6 h	**4.39 *****	**3.86*****	1.17	1.21

Through logistic regressions, we aimed at identifying the factors associated with engagement in the advanced certificate (Table 1). Only registrants who had responded to the initial survey were taken into account. We first tried to understand the background of participants who had submitted at least an assignment.

Geographical origin and time constraints were the main drivers of selection. As shown in Table 1, registrants from More Developed Countries (Very High HDI) were more likely to submit assignments and less likely to skip peer assessment than those from Least Developed Countries (Low HDI). Time constraints were also a very important driver of selection. Participants who were not able to spend more than two hours per week on the MOOC were unlikely to submit an assignment, and consequently to take part in the peer assessment process.

Given that taking part in peer assessment was encouraged but not compulsory to get the certificate, some participants skipped it. To analyze this phenomenon, we followed the same approach that we had used previously, but only registrants who had at least an assignment were taken into account in the logistic regression. Time constraints had no longer any statistically significant impact. Only geographical origin had a statistically significant impact in the spring edition. Participants from More Developed Countries were 70% less likely to skip peer assessment than participants from Least Developed Countries. This trend was also observed in the fall edition.

4. CONCLUSION

Among demographic factors, geographical origin, and to some extent employment status, were the most influencing factors regarding engagement in the peer assessment process. This trend had been detected in previous studies [2]. Time constraints were also one of the main drivers of selection, which is not surprising given that most registrants follow MOOCs during their free time. Given the amount of time required by assignments, selection based on motivation and availability occurred mostly before peer assessment itself. This may explain why no link was detected between skipping peer assessment and the number of hours participants had intended to spend on the course. Further investigations are needed to understand why participants from Least Developed Countries show lower levels of engagement than those from More Developed Countries, regarding both submission of assignments and participation in peer assessment.

Categorization of participants based on their behavior has been carried out mostly at the scale of the course [4]. Such approaches could be followed at the scale of the peer assessment process. Taking into account demographic parameters in the models might enhance the efficiency of strategies [6] aimed at increasing the precision and the efficiency of peer assessment.

5. ACKNOWLEDGMENTS

We acknowledge the work of Unow and of the team of volunteers who took part in the design of the course. We thank M. Epstein and M. Khaneboubi for their helpful remarks and feedbacks.

6. REFERENCES

[1] Breslow, L., Pritchard, D. E., Deboer, J., Stump, G. S., Ho, A. D., & Seaton, D. T. (2013). Studying Learning in the Worldwide Classroom: Research into edX's First MOOC. *Research and Practice in Assessment*, *8*, 13–25.

[2] Cisel, M. (2014). Analyzing completion rates in the First French xMOOC. *Proceedings of the European MOOC Stakeholder Summit 2014*

[3] Daniel, J. (2012). Making Sense of MOOCs: Musings in a Maze of Myth, Paradox and Possibility. *Journal of Interactive Media in Education*, *3*(0).

[4] Kizilcec, R. F., Piech C., Schneider E. (2013) Deconstructing Disengagement: Analyzing Learner Subpopulations in Massive Open Online Courses, LAK'13 *Proceedings of the Third International Conference on Learning Analytics and Knowledge*.

[5] Kulkarni et al. (2013) Peer and Self Assessment in Massive Online Classes. *ACM Transactions on Computer-Human Interaction*, *9*(39)

[6] Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., & Koller, D. (2013). Tuned Models of Peer Assessment in MOOCs. *Proceedings of the 6th International Conference of E.D.M*

[7] United Nations Development Programme. (2011). Human development report. Retrieved from http://goo.gl/WfA3Um

Peer Influence on Attrition in Massive Open Online Courses

Diyi Yang
Carnegie Mellon University
5000 Forbes, Pittsburgh, PA
diyiy@cs.cmu.edu

Miaomiao Wen
Carnegie Mellon University
5000 Forbes, Pittsburgh, PA
mwen@cs.cmu.edu

Carolyn Rose
Carnegie Mellon University
5000 Forbes, Pittsburgh, PA
cprose@cs.cmu.edu

ABSTRACT

In this work, we investigate the role of relational bonds in keeping students engaged in online courses. Specifically, we quantify the manner in which students who demonstrate similar behavior patterns influence each other's commitment to the course through their interaction with them either explicitly or implicitly. To this end, we design five alternative operationalizations of relationship bonds, which together allow us to infer a scaled measure of relationship between pairs of students. Using this, we construct three variables, namely number of significant bonds, number of significant bonds with people who have dropped out in the previous week, and number of such bonds with people who have dropped in the current week. Using a survival analysis, we are able to measure the prediction strength of these variables with respect to dropout at each time point. Results indicate that higher numbers of significant bonds predicts lower rates of dropout; while loss of significant bonds is associated with higher rates of dropout.

Keywords

Student Dropout, Peer Influence, MOOCs

1. INTRODUCTION

Massive Open Online Courses (MOOCs) such as those run through Coursera[1] have rapidly moved into a prominent place in the media. One notable problem with current MOOCs is the extremely high attrition, which inspires us to investigate what factors might affect student attrition [3, 2]. Prior work has been conducted to explore the connection between participation patterns in the discussion forum and student dropout. However, little attention has been paid specifically to the formation of relationship bonds during participation or how those relationship bonds influence the continuing commitment to the course. In this work, we investigate the connection between relational bonds and commitment to the course, which we refer to as **Peer Influence**. We leverage a statistical analysis technique referred as survival analysis to quantify the extent to which the informal relationships between students influence their dropout. First, we design five alternative operationalizations of relationship bonds based on patterns of communication and common topic focus in posts. We validate these five operationalizations as a single scale that enables us to construct three variables describing important aspects of the experience students have in the MOOC social environment.

2. PEER INFLUENCE EXPLORATION

In this section, we describe five separate operationalizations of relationship formation that we use to infer peer bonds.

- *Reply Interaction*: Who replies to whom is an explicit and direct indication of students' intention to socialize with specific other students. We generate a peer candidate set for a student based on the number of replies they have contributed to the posts of each of the other students as a reflection of their connection with them.
- *Co-occurrence Evidence*: Even though students are not talking to others directly, it is possible that they benefit from others' posts when they are exposed to them on the the threads they post to. Furthermore, participating in the same thread might also indicate that students share similar interests. Here, peer candidates are generated by ranking students based on number of common threads they have participated in.
- *Community Connection*: The participation patterns of students can be viewed as a social network graph, and we can use a graph partition method to identify subgraphs where students are located within that representation. Then we count the students within the same subgraph as more closely associated with one another than they are to others outside of the subgraph.
- *Topic Modeling*: Users who share interests usually talk about similar things. Similarity in topic focus can be treated as membership in an implicit interest defined subcommunity. To capture potential relations along this dimension, we use Latent Dirichlet Allocation [1] as a model to select students' peer candidates based on similarity of their topic distribution.
- *Cohort*: Cohort tells when the student has started their participation in the course and could be regarded

[1] https://www.coursera.org/

Aggregate Variables	Involved Original Variables
Cur	R_{cur}, C_{cur}, O_{cur}, T_{cur}, M_{cur}
$Prev$	R_{prev}, C_{prev}, O_{prev}, T_{prev}, M_{prev}
Num	R_N, C_N, O_N, M_N

Table 1: Variables organized into sets for constructing aggregate measures

as a proxy for their commitment (since students who join later tend to be less committed)[3]. Here, we generate the peer candidates for students based on their registration time.

3. VARIABLES DESIGN

Building on our five defined relationship measures, we formalize what relationship loss means by constructing three separate variables for each bond definition as follows. *Dropout in current week* (Cur), captures how many significant relations of student u dropped out in the current week; *Dropout in previous week* ($Prev$) captures how many significant relations of student u dropped out in the previous week; *Number of friends* (Num) describes how many significant bonds a student u has.

For each operationalization, we construct the three variables described above. Specifically, for reply bonds, we have R_{prev}, R_{cur}, R_N, representing the $Prev$, Cur, Num variables under the category of reply bonds; For co-occurrence bonds, O_{prev}, O_{cur}, O_N are gained; for community connection, we construct C_{prev}, C_{cur}, C_N; for topic modeling, we get T_{prev}, T_{cur} and discard T_{Num} which is the same for all students; for the motivation cohort, M_{prev}, M_{cur}, M_N are extracted. Those 14 variables are organized into three aggregate variables by simply averaging the same types of variables as shown in Table 1.

4. METHOD

The course we use to conduct the experiment is a Python programming course[2]: 'Learn to Program: The Fundamentals'. It has 3590 students who are active in the discussion forum, 24963 posts in total across the eight weeks. After aggregating the 14 variables into Cur, $Prev$, Num as described above, we then use those as input and conduct survival analysis to investigate how the three aspects influence the dropout of students. From the result presented in Table 2, we can observe that, (1) Students are around four times more likely to dropout if the number of their relation loss of close peers Cur are higher than average; (2) a student is 62% more likely to drop out if his/her relation loss $Prev$ is one standard deviation larger than average; (3) Comparably, the number of close peers Num indicates that the more close peers one student has, 74% less likely this student will drop out.

Figure 1 illustrates our result graphically. The middle solid curve shows survival with the number of Cur, $Prev$ and Num all at their mean level. The top curve above this middle one shows the survival when the number of close peers Num is one standard deviation above the mean (High), keeping the Cur and $Prev$ at their mean level. It indicates that higher Num is correlated with a longer continuing participation in the course. The bottom two curves show the survival when the dropout number of close peers in previous week or dropout number of such peers in the current week are both one standard deviation above the mean, keeping the other variables at their mean level. This reflects the influences of Cur and $Prev$ again – the more close peers a student loses, the less likely he/she will continue participating in the course forum.

Variable	Hazard Ratio	Std. Err	Z	P>|Z|
Cur	5.05	0.264	35.69	0.000
$Prev$	1.62	0.035	22.09	0.000
Num	0.26	0.014	-26.65	0.000

Table 2: Constructed Variables on Python Course

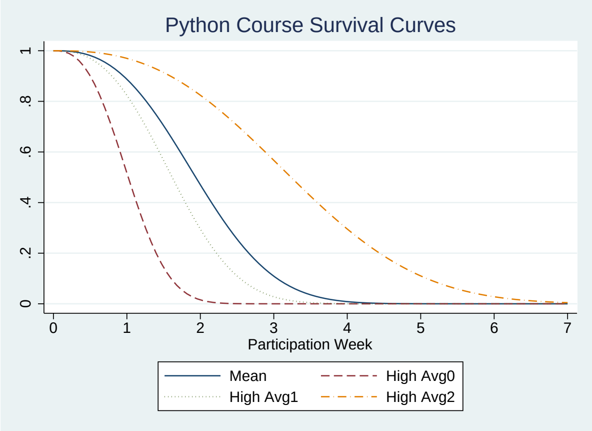


Figure 1: Survival Curves on Python Course

5. CONCLUSION

In this work, we propose to measure peer relations in the MOOC forums and explore how such relations influence student dropout. Reliable operationalizations of relations are constructed as well as variables corresponding to relationship loss. Via modeling of survival analysis , we find strong evidence that relationship loss is an important factor contributing to attrition. These results argue that attention to fostering a positive and supportive social environment could be an important direction for future MOOC development.

Acknowledgement

This work was supported in part by NSF grants IIS-1320064 and OMA-0836012.

6. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[2] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor. Modeling learner engagement in moocs using probabilistic soft logic.

[3] D. Yang, T. Sinha, D. Adamson, and C. P. Rose. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Workshop on Data Driven Education, Advances in Neural Information Processing Systems 2013*, 2013.

[2] https://www.coursera.org/course/programming1

Predicting Students' Learning Performance by Using Online Behavior Patterns in Blended Learning Environments: Comparison of Two Cases on Linear and Non-linear Model

Jeong Hyun Kim
Ewha Womans University
College of Education Bldg.
#533, Daehyun-dong,,
Seodaemun-gu, Seoul,
Korea, 120-750
+82-2-3277-3201
naralight@naver.com

Yeonjeong Park
Ewha Womans University
College of Education Bldg.
#533, 52, Daehyun-dong,
Seodaemun-gu, Seoul,
Korea, 120-750
+82-2-3277-3201
ypark78@ewha.ac.kr

Jongwoo Song
Ewha Womans University
Science Bldg. B #561,
Daehyun-dong,
Seodaemun-gu, Seoul,
Korea, 120-750
+82-2-3277-2299
josong@ewha.ac.kr

Il-Hyun Jo
Ewha Womans University
College of Education
Bldg. #533, Daehyun-dong,, Seodaemun-gu,
Seoul, Korea, 120-750
+82-2-3277-3201
ijo@ewha.ac.kr

ABSTRACT

A variety of studies using educational data traced from LMS has been conducted to predict students' performance. However, because of the complexity in its implementation, it is still challenging to predict students' learning achievement in blended learning environment. As an exploratory study, we selected two types of blended learning classes and compared their prediction models. While the first blended learning class which involves online discussion-based learning revealed a linear regression model, the second case, which was a lecture based blended learning class providing regular base online lecture notes in Moodle, did not present a linear regression model. After that, to examine the important variables of each class, RF (random forest) method was utilized. The results indicated different important variables in two cases. We concluded that the prediction models and data-mining technique should be based on the considerations of diverse pedagogical characteristics in blended learning.

Keywords

Educational Data Mining, Blended Learning, Prediction, Multiple Regression, Random Forest

1. INTRODUCTION

The use of learning management system (LMS) has grown exponentially. LMS offers a great variety of channels and workspaces to facilitate information sharing and communication among participants, to let educators distribute information to students, produce content materials, prepare assignments and tests, engage in discussions, manage distance classes and enable collaborative learning with forums, chats, file storage areas, news services, etc. [1]. Further, the large amount of students' behavioral data left in LMS can be accumulated as web-log files, extracted as valuable information, and finally utilized to improve students' learning achievement. As a result, a variety of studies using web-log data to predict students' performance has been conducted.

However, despite the abundant amount of research analyzing a massive amount of data and controlling student's academic achievement, it is still challenging to predict student's learning achievement in blended learning class which is commonly defined as an integration of traditional face-to-face and online approaches to instruction [2, 3]. In spite of applying the highly complicated and advanced data-mining technique, it is found that a single algorithm with the best classification and accuracy in all cases are not possible [4]. In higher education, there is considerable complexity in its implementation with the challenge of virtually limitless design possibilities and applicability [5]. This makes it difficult to predict student's achievement by using online learning patterns with a one-for-all prediction model.

Although there is no single framework for blended learning, it is generally assumed there are several types of blending in practice as the previous studies have attempted [6, 7]. Therefore, we intended to develop multiple prediction models to predict students' academic achievements according to the pedagogical types of blended learning. As an exploratory study, we implemented a different approach for two different types of blended learning, and tried to confirm the possibility to predict student's achievement in blended learning environments.

2. METHOD

2.1 Research Context

We analyzed the web log data of 43 college students of 'class A' and 30 college students of 'class B' opened in the regular fall semester in a large higher educational institution in 2013. While the major online activity in 'class A' was discussion forum, the second class involved a supplemental tool for submitting assignments and downloading learning materials.

2.2 Data Collection

In both cases, the data source (web-log data) was tracked from the Moodle. The independent variables for this study were computed by automatic data collection module embedded in the LMS. Total log-in time, log-in frequencies, regularities of log-in interval, visits on boards, visits on repositories were used as independent variables for both courses. Because there was no 'number of postings' variable for B class, the number of postings was used only for 'class A', This work used the Total Score as a dependent variable for each course.

2.3 Data Analysis

The procedure of data analysis consists of two phases. In phase 1, we conducted multiple linear regression analysis for both class A and B. While the class A showed a linear regression model, for the class B the linear model was neither proper nor statistically significant to predict student's achievement. Hence, as the second phase, we implemented Random Forest (RF) algorithm to increase the prediction accuracy.

3. RESULTS

3.1 Case 1: Discussion-based Learning

In class A, a blended learning which involves online discussion-based learning, linear multiple regression analysis was conducted, and this process generated a 'predictive model' of the student's final score (R^2 =.646, F=12.551, p =.000). Only two variables, log-in regularity and the number of postings in online forum, were statistically significant contributors.

3.2 Case 2: Lecture-Based Learning

In class B, a blended learning which involves offline lecture-based learning and online supplemental tool, we tried to find a model with a linear multiple regression analysis. However, only the total log-in frequency was significant, and F-test was insignificant with pretty low R^2 value (R^2 =.116, F=1.735, p =.167).

3.3 Random Forest Analysis

RF (random forest) method was tired in both cases to find important variables. As shown in Table 1, the discussion-based learning indicated the important variables as visits on board, the total log-in time, and the number of posting in forum (Pseudo R^2= 0.91), but the lecture-based learning indicated log-in regularity, the total log-in frequency, visits on board, and the total log-in time (Pseudo R^2=0.70). Here Pseudo R^2 was defined as follows.

- Pseudo R^2 = 1 – RSS/SST
- RSS = Residual sum of squares
- SST = Sum of squares of total

4. CONCLUSION

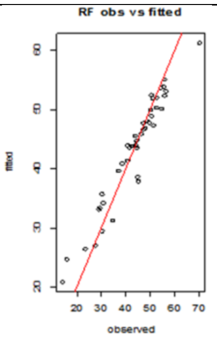
There is a variety of blended learning classes in universities and they are assumed to show different prediction models with a wide range of R^2 value. In this study, we presented that two different types of blended learning class show different models: linear and non-linear.

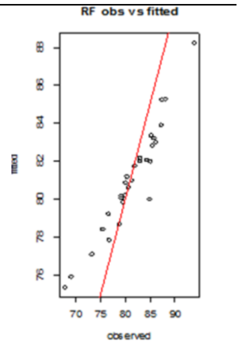
In case of the discussion-based blended learning course, which involves active learner's participations in online forum, a linear multiple regression analysis model explains the student's achievement. But in case of the lecture-based blended learning course, which involves submitting tasks or downloading materials as main online activities, linear multiple regression analysis model was not proper for prediction.

Additionally, in using a Random Forest approach, we found that two cases indicated different important variables which reflect the attributes of discussion-based learning class and lecture-based learning class, respectively. This result suggests that a future study needs to be conducted by clustering the types of blended learning classes throughout the students' online learning behavior data and predicting their learning achievement according to the clustered models. We conclude that the prediction models and data-mining technique should be based on the considerations of diverse pedagogical characteristics in blended learning.

Table 1. Comparison of important variables in two cases

Important Variable	Case 1 (Discussion-Based BL)	Case2 (Lecture-Based BL)
	N=43, Pseudo R^2= 0.91	N=29, Pseudo R^2=0.70
1	Visits on Board	Log-in Regularity
2	Total log-in time	Total log-in frequency
3	Number of Posting in forum	Visits on Board
4	Log-in Regularity	Total log-in time





5. ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2013S1A5A2A0304410

6. REFERENCES

[1] C. Romero, S. Ventura, and E. García, "Data mining in course management systems: Moodle case study and tutorial," Computers & Education, vol. 51, pp. 368-384, 2008.

[2] D. R. Garrison and N. D. Vaughan, "Institutional change and leadership associated with blended learning innovation: Two case studies," The Internet and Higher Education, vol. 18, pp. 24-28, 2013.

[3] C. R. Graham, W. Woodfield, and J. B. Harrison, "A framework for institutional adoption and implementation of blended learning in higher education," The Internet and Higher Education, vol. 18, pp. 4-14, 2013.

[4] C. Romero, P. G. Espejo, A. Zafra, J. R. Romero, and S. Ventura, "Web usage mining for predicting final marks of students that use Moodle courses," Computer Applications in Engineering Education, vol. 21, pp. 135-146, 2013.

[5] D. R. Garrison and H. Kanuka, "Blended learning: Uncovering its transformative potential in higher education," The internet and higher education, vol. 7, pp. 95-105, 2004.

[6] H. Singh, "Building effective blended learning programs," Educational Technology, vol. 43, pp. 51-54, 2003.

[7] R. Francis and J. Raftery, "Blended learning landscapes," Brookes eJournal of learning and teaching, vol. 1, pp. 1-5, 2005.

Predictive performance of prevailing approaches to skills assessment techniques: Insights from real vs. synthetic data sets

Behzad Beheshti
Polytechnique Montreal
behzad.beheshti@polymtl.ca

Michel C. Desmarais
Polytechnique Montreal
michel.desmarais@polymtl.ca

1. INTRODUCTION

A number of skills assessment models have recently emerged, and others have been around for decades. Their predictive performance have often been compared on a pairwise basis, but few studies have taken a comprehensive approach to compare them on a common basis. In this study, we apply a methodology that adopts both synthetic and real data for the purpose of this comparison. Synthetic data is generated from the underlying model of the different skills assessment techniques. The results show wide differences of performances between the skills assessment methods over synthetic data sets. They create a kind of “signature” for each specific data. If this signature is unique, it might reveal the latent structure of the skills. We discuss the potential benefits and the limits of the methodological approach that consists in exploring the performance of skills assessment methods based on the comparison of real and synthetic data.

2. SKILLS ASSESSMENT COMPARISON

This work compares a number of skills assessment techniques over real and synthetic data: the well known single skill Item Response Theory (IRT), the DINA and DINO models that rely on slip and guess factors [3], matrix factorization approaches based on conjunctive and disjunctive Q-matrices [1], and the POKS approach based on the Knowledge Space theory Falmagne [2], which does not directly attempt to model underlying skills but instead rely on observable items only. For baseline comparison, the expected value and majority class performances are also reported. The performance comparison relies solely on each approach’s ability to predict item outcome, not on the skills assessment directly which is not possible with real data.

The synthetic data sets are generated according to the each technique’s underlying model. We naturally expect to obtain the highest performance when the technique and the synthetic data underlying model are aligned, but of particular interest is the relative performance of the techniques over the different types of synthetic data. An interesting hypothesis is whether the performance patterns of the different techniques over a synthetic data set is unique and the extent to which it represents a “signature” of the underlying skills model ground truth of a data set.

3. RESULTS

Figure 1 and 2 show the performance of each technique over the synthetic and real data sets. We report the predictive accuracy of each method, along with the average success rate of the data set as a comparison point (last column), which constitutes the performance of predicting the majority class (or $(1-\text{perf})$ when perf is below 0.5). An error bar of 1 standard deviation is reported and computed over the 10 random sampling simulation runs and provides an idea of the variability of the results. Also reported is the performance of random data with a 0.75 average success rate.

4. DISCUSSION

The results do show wide differences in the performance of the techniques for different synthetic data sets. For real data sets, the differences are smaller, though still significant.

An interesting finding is that the relative performance of the different skills modeling approaches create signatures over data sets. According to these signatures, the Vomlel real data set is closest to the linear compensatory simulated data set. As could be expected, random data does have a unique signature of its own: all methods converge towards the score of the majority class. The EPCE data set is close to this signature.

Another finding is the small relative differences between the techniques for the Fraction 2/3 data set compared to the other Fraction data sets and the Vomlel data set. This data has the peculiarity that the items were chosen based on a small number of single skill per item.

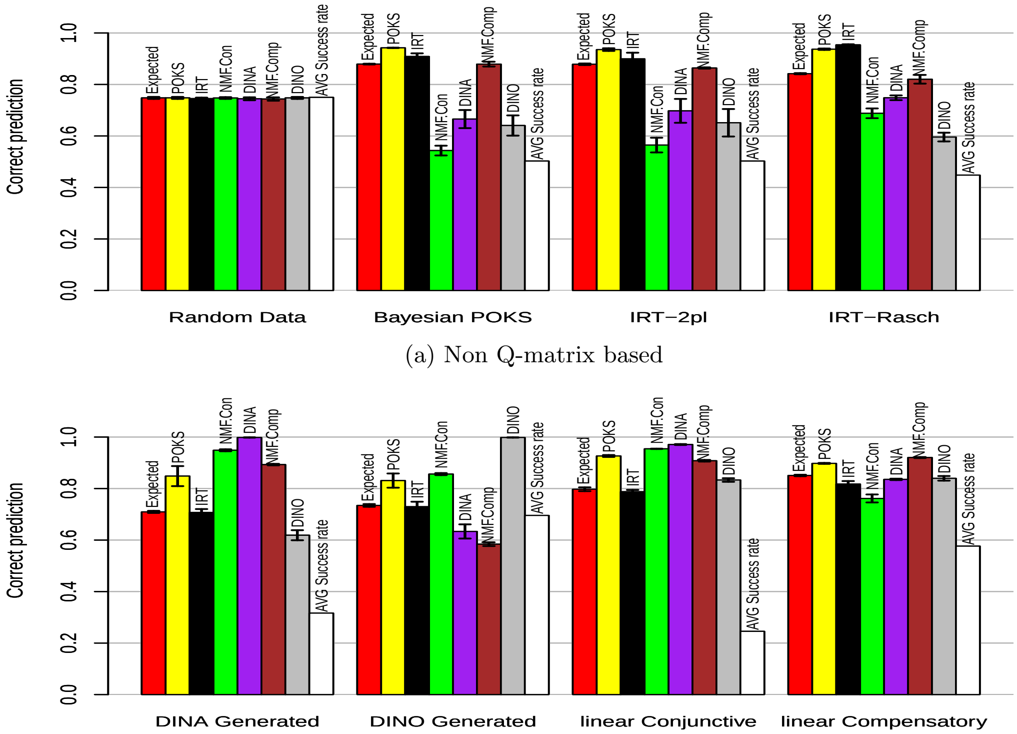
Future work will aim to establish if the findings generalize and the extent to which performance patterns generalize, but the approach of comparing these patterns of multiple models and techniques over real and synthetic data sets appears promising.

References

[1] M. Desmarais. Conditions for effectively deriving a Q-matrix from data with non-negative matrix factorization. In *4th International Conference on Educational Data Mining, EDM*, pages 41–50, 2011.

[2] M. C. Desmarais, P. Meshkinfam, and M. Gagnon. Learned student models with item to item knowledge structures. *User Modeling and User-Adapted Interaction*, 16(5):403–434, 2006.

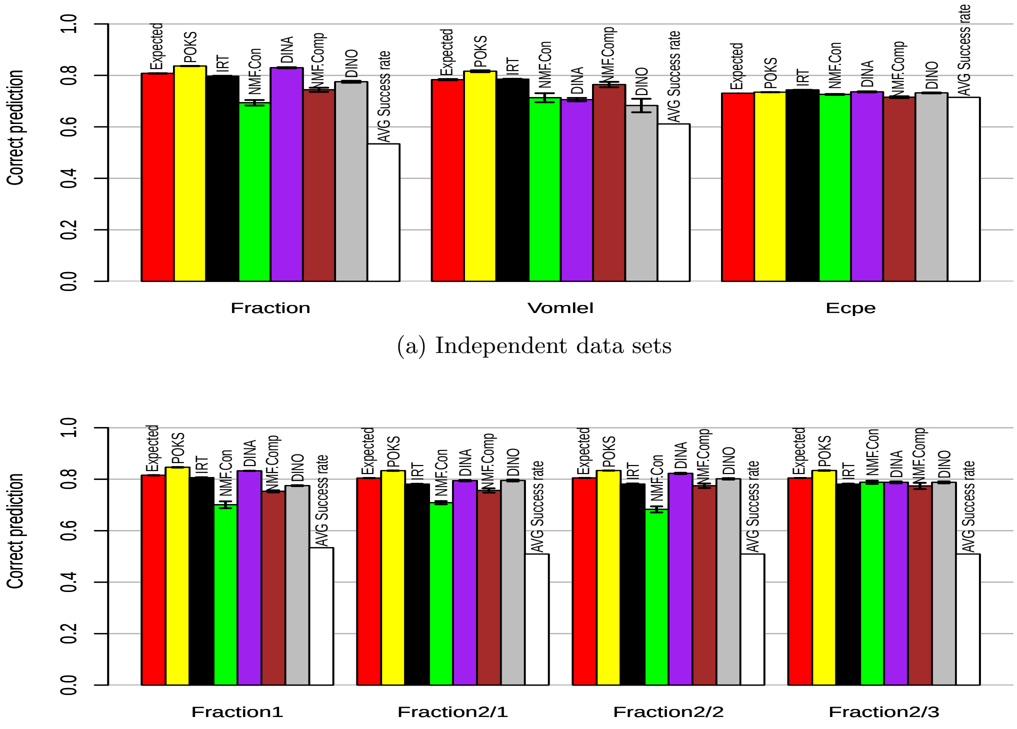
[3] B. W. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.



(a) Non Q-matrix based

(b) Q-matrix based

Figure 1: Item outcome prediction accuracy results of **synthetic data sets**



(a) Independent data sets

(b) Subsets of the Fraction data set

Figure 2: Item outcome prediction accuracy results of **real data sets**

Recent-Performance Factors Analysis

April Galyardt
University of Georgia
110 Carlton St.
Athens, GA
galyardt@uga.edu

Ilya Goldin
Center for Digital Data, Analytics, and Adaptive Learning
Pearson
ilya.goldin@pearson.com

ABSTRACT

We introduce R-PFA, a new model for predicting whether or not a student will answer an item correctly based on the student's history of practice. The key idea in R-PFA is to represent history as a recentcy-weighted proportion of correct responses. In an evaluation on a dataset from the Assistments tutoring system, we find that R-PFA improves predictive accuracy over other logistic regression model variants, including PFA and AFM.

Keywords

performance modeling, moment of learning, linear logistic test model

1. INTRODUCTION

An interactive learning environment (ILE) can adapt its behavior to what the student does and does not know. For example, an ILE may hold a domain model in terms of knowledge components (KCs) to be taught to students [5], and estimate each student's proficiency with a KC based on the student's practice with problems involving the KC. One popular model for estimating student proficiency in this setting is Performance Factors Analysis (PFA) [6]. PFA is a parameterization of the Linear Logistic Test Model [3] that predicts performance on the current item using the entire history of success and failures on previous items addressing the same KC (Eq 1).

We introduce Recent-Performance Factors Analysis (R-PFA), which extends PFA via a simple variable transformation. R-PFA includes information about whether or not a moment of learning has occurred [4]. We demonstrate that R-PFA shows improved predictive performance over PFA as well as over the Additive Factors Model (AFM) [2]. We also contend that the simplicity of R-PFA provides it with a distinct advantage over the methodology in [4] for prediction.

2. METHODS

R-PFA is based on a simple observation: If a student has already experienced a moment of learning then recent performance is likely to consist primarily of successful attempts. If a moment of learning has not yet occurred, then recent performance is likely to contain more failed attempts. In effect, recent history serves as a proxy for whether or not a moment of learning has occurred.

We compare the performance of R-PFA (Eq 2) to PFA (Eq 1), AFM, and a simplified version of PFA using success only, on the Assistments data used in the original "moment of learning" work [1]. The data contain first attempts by 4138 students on problem sets involving 54 knowledge components (KC), for a total of 187,309 first attempts. Each problem is coded with only a single KC. Table 1 lists the features in each model.

$$logit(p_{ijt}) = \theta_i + \beta_j + \alpha_j S_{ijt} + \rho_j F_{ijt} \tag{1}$$

$$logit(p_{ijt}) = \theta_i + \beta_j + \gamma_j T_{ijt} + \delta_j R_{ijt} \tag{2}$$

We use the notation:

j	KC indicator
i	student indicator
X_{ijt}	binary correct/incorrect, student i, KC j, trial t
S_{ijt}	count of previous successes, up to trial t
F_{ijt}	count of previous failures, up to trial t
T_{ijt}	count of past opportunities, $S_{ijt} + F_{ijt}$
R_{ijt}	recency-weighted proportion of past successes
p_{ijt}	$Pr(X_{ijt} = 1)$.

Table 1: Terms in predictive model variants.

	Student	KC	Success	Failure	Totals	Weighted Proportion
AFM	θ_i	β_j			$\gamma_j T_{ijt}$	
sPFA	θ_i	β_j	$\alpha_j S_{ijt}$			
PFA	θ_i	β_j	$\alpha_j S_{ijt}$	$\rho_j F_{ijt}$		
R-only	θ_i	β_j				$\delta_j R_{ijt}$
R-PFA	θ_i	β_j			$\gamma_j T_{ijt}$	$\delta_j R_{ijt}$

As a measure of 'recent' history, we introduce R_{ij}, an exponentially weighted proportion of successes.

$$R_{ijt} = \frac{\sum_{p=-2}^{t-1} b^{(t-p)} X_{ijp}}{\sum_{p=-2}^{t-1} b^{(t-p)}} \tag{3}$$

The decay factor b is a tuning parameter that controls the weights, and thus controls whether 'recent' means just the most recent trial or the entire history of practice. We compare values of {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} for b.

One potential issue with R_{ij} is that the proportion of recent successes is very noisy on the first few trials. To adjust for this noise, we stipulate ghost attempts $X_{ij-2} = X_{ij-1} = X_{ij0} = 0$. The ghost attempts are an explicit assumption that at time 0, the student has not already learned the KC. These ghost attempts affect only the value of R_{ijt}, i.e., they are not extra instances in the data set.

3. RESULTS

We fit a total of 23 models to the Assisstments data, using the `glmer` function in the R package `lme4` to fit all models. We compared models in terms of AIC and BIC. Both ranked our models in the same order for this data, so we only report AIC scores in Figure 1.

All models that include R_{ij} (R-PFA and R-only) outperform all existing models by a wide margin. As in previous work, PFA outperforms AFM [6]. Finally, the count of prior successes alone (sPFA) is a better predictor than total opportunities (AFM).

When $b = 1$, R_{ij} is the overall proportion of successes, so R-PFA and R-only use the entire history of performance. Yet, proportion of success (R-PFA and R-only) is more predictive than the number of successful attempts (PFA). For any fixed decay parameter b, R-PFA is better than R-only. The total number of practice opportunities is still informative above and beyond the recent history.

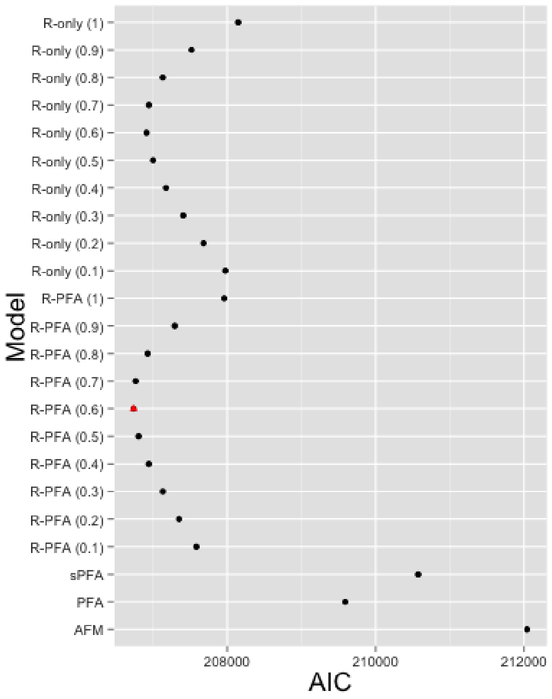


Figure 1: AIC scores for all models (lower is better). The red triangle indicates the lowest AIC.

4. DISCUSSION & CONCLUSIONS

The R-PFA model differs from PFA in two significant ways. First, unless $b = 1$, R-PFA values recent evidence more than older evidence. Second, the ghost attempts reduce the noise of predictions on the first few attempts that a student makes on any particular KC by incorporating the belief that students are unlikely to already know the KC and are unlikely to perform well on a skill on their early attempts. With these two modifications, all variants of R-PFA outperform PFA and other models in terms of predictive accuracy. Ghost attempts and decay weights matter in combination. The ghost attempts necessarily have the greatest influence on practice strings that are relatively short, and there are many such occurrences in our dataset. The ghost attempts reduce the noise that would otherwise be present in R_{ij} for these attempts. The weighting that controls the window of "recent" performance considered is also key. The difference in AIC scores between R-PFA with $b = 0.6$ and $b = 1$ is as great as the difference between sPFA and PFA. However, while R-PFA with $b = 0.6$ performed best on this dataset, that specific value of b may be due to the mastery criterion (a streak between 3-5 trials) in the Assistments software.

There are a number of tunings of R-PFA that we may explore in future work. First, there may be a relationship between the optimal number of ghosts attempts and the decay parameter b. Second, should there be different b values for different knowledge components? Third, should first-attempt hint requests be distinguished from first-attempt incorrects, by incorporating separate proportions for these prior practice outcomes? We hope that R-PFA sees widespread use in the toolset of educational data mining.

5. REFERENCES

[1] R. S. Baker, A. B. Goldstein, and N. T. Heffernan. Detecting learning moment-by-moment. In *IJAIED*, volume 21, pages 5–25, 2011.

[2] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Proceedings of 8th ITS Conference*, pages 164–175, Berlin / Heidelberg, 2006. Springer.

[3] P. de Boeck and M. Wilson, editors. *Explanatory item response models: a generalized linear and nonlinear approach.* Springer, New York, 2004.

[4] A. Hershkovitz, R. S. Baker, S. M. Gowda, and A. T. Corbett. Predicting future learning better using quantitative analysis of moment-by-moment learning. In *Proceedings 6th EDM Conference*, pages 74–81, 2013.

[5] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[6] P. I. Pavlik, H. Cen, and K. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of AIED*, pages 531–538. IOS Press, 2009.

Refining Learning Maps with Data Fitting Techniques: Searching for Better Fitting Learning Maps

Seth Adjei, Douglas Selent,
Neil Heffernan
Worcester Polytechnic Institute
{saadjei, dselent, nth}@wpi.edu

Zach Pardos
UC-Berkeley
pardos@berkeley.edu

Angela Broaddus, Neal Kingston
University of Kansas
{broaddus,nkingsto}@ku.edu

ABSTRACT

Learning sciences needs quantitative methods for comparing alternative theories of what students are learning. This study investigated the accuracy of a learning map and its utility to predict student responses. Our data included a learning map detailing a hierarchical prerequisite skill graph and student responses to questions developed specifically to assess the concepts and skills represented in the map. Each question aligned to one skill in the map, and each skill had one or more prerequisite skills. Our research goal was to seek improvements to the knowledge representation in the map using an iterative process. We applied a greedy iterative search algorithm to simplify the learning map by merging nodes together. Each successive merge resulted in a model with one skill less than the previous model. We share the results of the revised model, its reliability and reproducibility, and discuss the face validity of the most significant merges.

Keywords

Learning Maps, Iterative Search, Cognitive Modeling, Skill Graph

1. INTRODUCTION

Cognitive models are used to represent how one's knowledge may be organized. As such, they contain descriptions of component pieces of knowledge and connections among the components to indicate how understanding develops in a specified domain [4]. Different authors have described various cognitive models, including learning maps [5], learning trajectories [2], and learning hierarchies [3]. Learning maps use linear sequences of learning goals and are useful for instructional planning [5]. A learning trajectory includes a learning goal, a developmental progression defining the levels of thinking students pass through as they work toward the defined goal, and a set of learning activities or experiences that assist students in reaching the defined goal [2]. Learning hierarchies model prerequisite knowledge components in hierarchies, allowing multiple pathways to extend from one prerequisite skill to multiple learning goals [3].

In the present study we examine a small section of the learning map and investigate the effects of permuting the topology of the hierarchy. Skills and concepts are represented by latent nodes in the learning map. Directed edges represent the prerequisite relationship among latent nodes and also represent the relationship between those nodes and their associated test items. We present a simple method for improving the predictive power of the learning map by combining latent nodes.

This work connects with literature on searching for better fitting cognitive models. Several non-hierarchical cognitive models have been developed to represent the relationship between knowledge components (KCs) in the form of prerequisite skill maps. These cognitive models have been developed to help intelligent tutors, as well as experts, determine student mastery of KCs. A number of technical approaches have been developed to evaluate cognitive models developed by domain experts. One approach is Learning Factors Analysis (LFA), developed by Cen, Koedinger and Junker [1] to help the Educational Data Mining (EDM) community evaluate different cognitive models. LFA incorporates a statistical model, item difficulty and a combinatorial search to select the model. Our work is different from the flat Item Response Theory (IRT) models presented in [9] in that IRT does not deal in any way with hierarchical relationships between knowledge components.

In this work we follow the process described by Cen, Koedinger and Junker [1]. This technique can be used to analyze hypothesized learning maps and consider whether small improvements to the model result in a better fit to the data. In this method two different approaches were studied to determine the best skill map from an initial graph. Cen, Koedinger, and Junker suggested three types of operations, i.e., merges, splits, and adds [1]. However, in this study, we used only merge operations given the already highly granular quality of our initial, subject matter expert derived learning map.

2. Initial Learning Map

This study examined a section of the learning map containing 15 concepts and skills related to understanding integers. The map was developed using mathematics educational literature describing how students learn to understand and operate with integers. The set of integers includes the whole numbers and their opposites, presenting many students their first exposure to negative numbers [6].

The data for this study was gathered from responses of 2,846 students to 25 test items aligned to 15 skills. All of the test items were multiple choice questions, with four answer options per question. Each skill was assessed by one or more items. As part of the test development process, subject matter experts confirmed the alignment of each item to its associated skill, meaning that the item was judged by experts to evoke the intended skill. Furthermore, due to the hierarchical structure of the learning map, items associated with skills lower in the learning map were assumed to be more difficult, i.e., require more skills, than items associated with skills higher in the learning map.

3. Experiment

In all of the experiments our sole manipulation of the map was to merge latent nodes. A merge operation occurred when two skills adjacent to each other in the map were combined into one skill. Items from both skills that were merged were reattached to the new single skill. The prerequisites of the constituent skills became prerequisites of the merged skill and the same applied to the post-requisites.

To evaluate the models, we used per student per item cross validation with 5 student folds and 3 item folds. Our student and item folds were chosen randomly for the evaluation; however each item fold consisted of the same random partition of items. More details about how the cross-validation was done as well as other details on the algorithms used in this experiment can be found in the technical document [8]. We used the Root Mean Squared Error (RMSE) of the predictions to evaluate the results of the experiment. A lower RMSE means the model is performing with a higher accuracy.

Figure 1 shows a graph of the results from the iterative search. The search started at iteration 1, which was the initial skill map consisting of 15 skills before any merges were applied to it. The search ended at iteration 15, which is a graph consisting of just one skill with all the items attached to that one skill. The best models from each iteration are shown in Figure 1. We used RMSE to choose the best model at each iteration and to guide our search. As the number of iterations increases, the number of skills decreases since skills are merged.

Performance of the Number of Skills

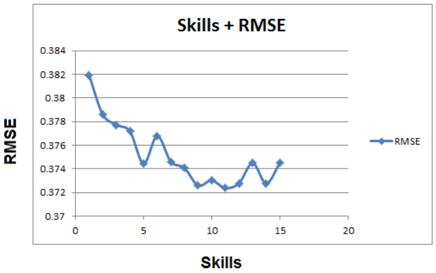


Figure 1: Performance for each number of skills. Each merge operation reduces the number of skill by 1. After iteration 1, there are a total of 15 skills. After iteration 15, there is only one skill

The results show that the best RMSE obtained was from the 11-skill map at iteration 4 with an RMSE of 0.372. This is slightly better than the original skill map with RMSE of 0.375. The 11-skill map has a small but significant improvement (p <0.01) from the original skill map. The effect size was negligible (0.01).

4. Contributions, Conclusions and Future Work

The main contribution of this paper is the provision of a greedy algorithm that simplifies learning maps. We showed that this simplification is possible without losing the predictive powers of the learning map. Even though this simplification could be done by hand, this algorithm will be useful in situations where the learning map being simplified is large.

This paper presents an initial experiment in this novel area of EDM. Instead of focusing all our attention on the flat IRT model, the community needs to pay a closer attention to and explore models that deal with hierarchical relationships between knowledge components. These studies and contributions thereof can assist domain experts to produce better fitting models which should impact student learning positively.

Since merging skills increased accuracy, these results suggest that the original skill map was too fine-grained (given the number of questions per skill and the number of students who took the test.). In some cases the test items did not adequately distinguish between the skills that were merged; hence such skills were merged. The results of algorithms like this can help the content experts who are creating skill maps and test items to either reconsider thinking of two skills as separate, or prompt them to write different test items to better distinguish between students that have mastered one of the skills but not the other skill. As future work, we intend to examine the other operations of the LFA method for refining learning maps. These include *splits* and *adds*, which were described earlier.

5. REFERENCES

[1] Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, T.-W. Chan (Eds.) Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 164-175. Berlin: Springer-Verlag.

[2] Clements, Douglas, and Julie Sarama. (2004). “Learning Trajectories in Mathematics Education.” Mathematical Thinking and Learning An International Journal 6:81-89.

[3] Gagné, Robert. 1968. “Learning Hierarchies.” Educational Psychologist 6:1-9.

[4] Gierl, Mark, Changjiang Wang, and Jiawen Zhou. 2008. “Using the Attribute Hierarchy Method to Make Diagnostic Inferences about Examinees' Cognitive Skills in Algebra on the SAT.” The Journal of Technology, Learning, and Assessment, 6(6):4-49.

[5] Popham, W. James. 2011. Transformative Assessment in Action: An Inside Look at Applying the Process. Alexandria,VA: ASCD.

[6] Van de Walle, J. A., Bay-Williams, J. M., Karp, K. S., Lovin, L. H. (2014). Teaching Student-Centered Mathematics: Developmentally Appropriate Instruction for Grades 6-8 (2nd ed.). Upper Saddle River, NJ: Pearson Inc.

[7] van der Linden, W. and Hambleton, R. (Eds.). Handbook of Modern Item Response Theory. New York, NY: Springer-Verlag (1997).

[8] Learning Maps Dataset and Evaluation Code https://sites.google.com/site/assistmentsdata/kansas-project

Last accessed on February 25, 2014

Relevancy Prediction of Micro-blog Questions in an Educational Setting

Mariheida Córdova Sánchez[*]
Information Technology
Purdue University
115 S. Grant Street
West Lafayette, IN 47907
cordovas@purdue.edu

Parameswaran Raman
Department of Computer Science
Purdue University
305 N. University Street
West Lafayette, IN 47907
params@purdue.edu

Luo Si
Department of Computer Science
Purdue University
305 N. University Street
West Lafayette, IN 47907
lsi@purdue.edu

Jason Fish
Information Technology
Purdue University
115 S. Grant Street
West Lafayette, IN 47907
jfish@purdue.edu

ABSTRACT

Micro-blogging has become increasingly popular in recent years. Using micro-blogging in a large classroom could be beneficial for learning. However, sometimes addressing the large number of posts could be cumbersome to a reader who has only limited time in a classroom. We propose a novel solution for predicting the relevancy of a question asked in a class by looking at the questions asked in previous semesters, the similarity of the question to the lecture material, as well as a set of question features such as the number of students' votes, number of replies, the length of the question, and whether it was asked anonymously. To identify similar questions asked previously, topic modeling and feature selection are used. Empirical results show that topic modeling leads to better prediction performance score as compared to feature selection. The similarity of the question and its corresponding lecture material further improves the relevancy prediction of the questions.

Keywords

Text Categorization, Micro-blogging, Topic Modeling, Feature Selection

1. INTRODUCTION

Using micro-blogging, students are able to ask questions about the material without interrupting the class, which increases student participation. However, answering these questions could be a cumbersome activity as the posts pile up and there is only limited time during the lecture.

In this paper, we propose a novel solution for predicting the relevancy of a question asked in a class by looking at the questions asked in previous semesters and the course lecture material. We also use a set of features such as the number of replies and votes the question received, the length of the question, and whether the question was asked anonymously. To identify similar questions asked previously, topic modeling and feature selection are used. The data consists eight semesters of a Personal Finance course offered at Purdue University.

Cetintas et al. [2] propose a few approaches to this by using the correlation between questions to identify the most relevant and irrelevant questions. In [1], Cetintas et al. propose a text categorization approach that uses personalization, correlation between questions themselves, and students' votes on questions. However, Cetintas et al. do not explore using topic modeling, nor did they explore using feature selection for their classification task. The use of topic modeling for microblog content has been explored by Remage et al. [3].

Empirical results show that topic modeling leads to a better prediction score as compared to feature selection. The similarity of the question and its corresponding lecture material further improves the relevancy prediction of the questions.

2. MODELS

For purposes of training and testing the models, the data were divided into two parts in time, which means that the train data corresponds to previous semesters, while the test data belongs to future semesters. Cross validation and regularized were used in all models.

2.1 Model using Post Features

[*] This author is also a student in the Department of Computer Science at Purdue University.

A model was built using features from the posts. These features are: the length of the post, the number of votes the post received, the number of replies the post received, and whether or not the post was posted anonymously. These features are referred to as *Post Features* and the model that only uses these features is called *LR_Post*.

2.2 Topic Modeling

In order to find which posts are relevant and which ones are not, we first find what topics the students are talking about. The intuition behind this is that we might find that some posts are about topics directly related to the course, while other topics are regarding projects, assignments, exams, etc. We used Latent Dirichlet Allocation, or LDA, to find a set of topics from the posts.

2.2.1 Model using LDA

The output of the Latent Dirichlet Allocation algorithm is a set of topics with the probability distribution of each post belonging to them. These probability distributions are used as prediction features for this model, along with the *Post Features* discussed in section 2.1. We call this model *LR_LDA_Post*. We experimented using different number of topics and terms and chose 10 topics with 15 terms in each since we observed the best performance with this combination.

2.2.2 Model using feature selection

Another approach to topic modeling used was to take the most popular terms of each topic and only consider those terms, disregarding all other terms belonging to the topics. For the top terms of each topic, we find the term frequency in the post. We then have a set of features, which are the frequencies of these terms in the posts. We call this model *LR_FeatSel_Post*.

2.3 Model using the lecture material

An important factor when considering the relevancy of a post is what was actually being discussed during that particular lecture. It could happen that the post is relevant to the overall course, but not relevant to the current lecture. For this matter, the similarity of the post to the lecture was calculated. The Kullback-Leibler divergence, or KL divergence, was used. For this, the lecture was divided into smaller overlapping chunks and the similarity between these chunks and the post was then calculated. We explore using different sizes of chunks and chose a size of 100 characters since we observed the best performance. This feature is referred to as *KLD*. *KLD*, together with *Post Features*, forms another model called *LR_KLD_Post*.

2.4 Model Comparison

The different models were evaluated using the F1 score. Figure 1 shows a comparison of the models. All the different models shown in this figure include the *Post Features* described in section 2.1. The *Basic* model in the figure contains only the *Post features*, i.e. model *LR_Post*. Following this model, we show the models *LR_FeatSel_Post*, *LR_LDA_Post*, and *LR_KLD_Post*. Then we show the models which include the post features together with topic modeling features and KLD features, i.e. *LR_FeatSel_KLD_Post* and *LR_LDA_KLD_Post*. From this figure we can see that having the *Post Features* alone yeilds the lowest performance score of 0.72. Adding feature selection to it gives us a performance of 0.782, while adding the LDA features to it achieves a performance of 0.795.

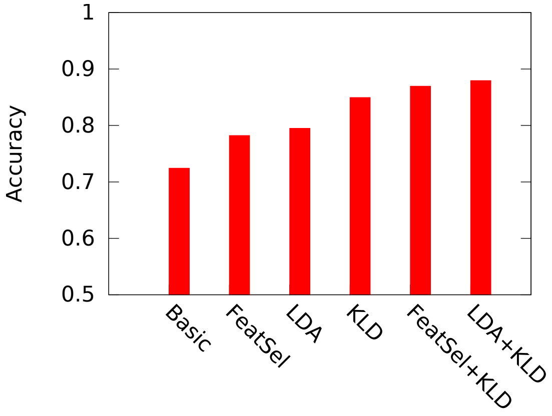


Figure 1: Model comparison

Comparing the two topic modeling approaches to the KL divergence approach, we can see that the KL divergence performs better. With the *LR_KLD_Post* model we obtain a performance score of 0.850. Including the *KLD Feature* to both topic models achieves a better performance. When the *KLD Feature* is added to the Feature Selection model, the performance goes up to 0.870. Similarly, when the *KLD Feature* is added to the LDA model, the performance of the model goes up to 0.880.

3. CONCLUSIONS

The experimental results show that all the features used in our models are helpful in predicting the relevancy of questions. LDA performs slightly better than feature selection for our application. We also show that adding the similarity of the posts to the lecture material further improves the performance of the techniques.

4. ACKNOWLEDGMENTS

This work was partially supported by NSF research grants IIS-0746830, CNS- 1012208, IIS-1017837 and CNS-1314688. This work was also partially supported by the Center for Science of Information, an NSF Science and Technology Center, under grant agreement CCF-0939370.

5. REFERENCES

[1] S. Cetintas, L. Si, H. Aagard, K. Bowen, and M. Cordova-Sanchez. Microblogging in a classroom: Classifying students' relevant and irrelevant questions in a microblogging-supported classroom. *IEEE TLT*, 4(4), 2011.

[2] S. Cetintas, L. Si, S. Chakravarty, H. Aagard, and K. Bowen. Learning to identify students' relevant and irrelevant questions in a micro-blogging supported classroom. In *ITS*, volume 2, 2010.

[3] D. Ramage, S. T. Dumais, and D. J. Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.

Singular Value Decomposition in Education: a case study on recommending courses

Fábio Carballo
Instituto Superior Técnico – Universidade de Lisboa
Av Rovisco Pais
1049-001 Lisboa
+351 218 419 407
fabio.carballo@tecnico.ulisboa.pt

Cláudia Antunes
Instituto Superior Técnico – Universidade de Lisboa
Av Rovisco Pais
1049-001 Lisboa
+351 218 419 407
claudia.antunes@tecnico.ulisboa.pt

ABSTRACT

After bachelor, many students strive to select the masters' courses that are most likely to meet their interests. Although this decision may have a big impact on students' motivation and future achievements, usually no support is offered to contest this problem. The use of recommendation systems to suggest items to users has well-known success in several domains, and some of the most successful techniques use Singular Value Decomposition (SVD) to capture hidden latent factors in reduced dimensionality and produce high quality recommendations. In this paper, we propose to use SVD, with a contextual mapping to the educational paradigm, to capture relationships between courses grades and recommend masters' courses that are suitable to students' skills given their bachelor achievements. Our results show that using SVD to predict the masters' courses marks has potential to serve as basis for the recommendation production.

Keywords

Courses recommendation, Singular Value Decomposition

1. INTRODUCTION

The decision that students have to make on which master's courses to enroll has way more impact than it looks: this choice can have a direct effect on their academic and personal goals. A bad choice of courses may demotivate a student, which can cause the student to drop out or to not take advantage of the fullness of his capabilities. Therefore, understanding students' particularities is needed, so as to recommend courses that are not only interesting to them, but also adequate to their capabilities. Current solutions have a tendency to recommend courses based on its contents or potential interest to the students, not considering how those courses can affect students' overall academic performance [1]. Therefore, we propose the creation of a system that, with the minimal user-participation, recommends masters' courses that add value to students' academic achievements, given their bachelor path. To do it, we explore Singular Value Decomposition so as to capture hidden factors in the historical students marks and then identify the best courses to recommend

With the Netflix challenge [2][3], there was a huge trend to use *latent factor models*, in order to reveal the hidden latent features that somehow explain the observed ratings. The most successful technique in these models is *Singular Value Decomposition*, due to its accuracy and scalability. This technique factors an m x n matrix R, into three matrices as in (1),

$$R = U \times S \times V' \qquad (1)$$

where U and V are two orthogonal matrices of size m x r and n x r respectively, while r represents the rank of the matrix R. Matrix S is a diagonal matrix, and its entries are stored in decreasing order of their magnitude. Each entry of matrix S represents a hidden feature and the stored value in it stands for the weight the feature has to the variance of the values on R. The sum of the values of all entries represents the total variance on matrix R. SVD has many applications of particular interest, but it is especially useful as a way to find the best rank-k approximation, R_{k}, to the matrix R, such that the Frobenius norm of R - R_{k} is minimized. The Frobenius norm (|| R - R_{k} ||F) is defined as the sum of squares of elements in R - R_{k} . To reduce the rank r to k, where $k < r$, one should only use the first k diagonal values of the matrix S (the singular values), and then reduce both U and V accordingly. The result is the closest k-rank approximation $R_{\mathrm{k}} = U_{\mathrm{k}}\mathrm{x}S_{\mathrm{k}}\mathrm{x}V'_{\mathrm{k}}$.

The usual idea, when using this technique on recommendation systems, is to use R as a users-items matrix, where m is the number of users and n the number of items. The value of each cell holds the rating that a user has given to a certain item. The idea is that after the decomposition we can calculate both the users-features and items-features spaces and use them to predict ratings. In the users-features space, $U_k\sqrt{S_k}$ – let's call it P – each row is a vector with the preference values of a user over the discovered features. On the other hand, in the items-features space, $\sqrt{S_k}V'_k$ – let's call it W - each row is a vector that represents how the item is weighted in each feature. Hence, this consists on the decomposition of the usual user-item matrix into a k-dimensional space where just the k most relevant features are taken into account: the noise in the data is reduced, and this enables the production of better quality rating predictions.

However, SVD is known for not dealing well with sparse matrices, where there are a lot of missing values. Gladly, Simon Funk found a solution to this problem [3]. He proposed to use a *gradient descent algorithm* in order to compute the best rank-k matrix approximation using only the known ratings of the user-item matrix R. This process follows the same idea than the training on *neural* networks. With the error in a prediction of user i to item j being $(R_{ij} – Rk_{ij})$, Funk's approach takes the derivative of the square of the error with respect to P_{ik} and then with respect to W_{jk}. Since R is constant, and $R_k = P$xW' (note that in this approach P and W contain the S matrix, that usually results from the matrix decomposition), the updates for the user and item spaces, P and W then become (2) and (3), respectively:

$$P_{if}(t+1) = P_{if}(t) + learning\_rate * (R - R_k)_{ij} * W_{jf}(t) \qquad (2)$$

$$W_{jf}(t+1) = W_{jf}(t) + learning\_rate * (R - R_k)_{ij} * P_{if}(t) \qquad (3)$$

In summary, the final solution of this learning problem is the combination of feature weights on both P and W such that the error in the approximation R_k is minimized. This solution is determined iteratively, as the gradient of the error function is computed at each iteration step. Note that in this approach all features vectors are initialized with the global rating average along with some introduced random noise.

2. SVD-BASED COURSES RECOMMENDATION

As we stated above, we aim for exploring SVD to recommend masters' courses to students given only their bachelor's courses marks. Hence, we must start of an historical record over triplets in the form of <*Student, Course, Mark*> into a structure that SVD can explore. As we have seen, SVD makes use of a matrix *R* that holds knowledge over the ratings that users gave to items. In usual representations, users are placed in rows and items in columns, and each cell R_{ij} in the matrix corresponds to the rating that the i^{th} user attributes to the j^{th} item.

As a first step to map our problem to the educational context we must transform our historical students' marks record into a matrix *R* that holds our knowledge over students' capabilities in each course taken. Our proposal is that matrix *R* will have students represented on rows and courses on columns, and each entry R_{ij} will be filled with the mark obtained by the i^{th} student on the j^{th} course. When students didn't enroll, the mark is the zero value. This is a natural mapping, as we want to recommend the courses with the predicted best marks, while having the constraint of recommending only a subset of the courses, the masters' courses. Our idea is to apply SVD to the matrix described above, so as to predict the marks of every student on all masters' courses and then use those predictions to recommend a specific set of courses. We will use Funk's *gradient descent* algorithm to calculate SVD, and, consequently, produce both the users and courses spaces. Applying Funk's gradient descent to the student-courses matrix *R* (with *N* number of features to discover) we get matrices *P* and *W*. Matrix *P* represents the user features dimensional space, where row *i* stands for student *i* features vector, which relates the student with each of the *N* features. Likewise, each row of matrix *W* shows how each course is related to each one of the *N* features. The product *PW'* constitutes a *N*-rank approximation of the original matrix *R*.

At this moment, we can use matrices *P* and *W* to predict the students' masters' marks. The predicted mark of a student *i* on course *j* corresponds to the dot product between the i^{th} row of *P* and the j^{th} row of *W*. This dot product represents how the student is related with the course according the several features. The predicted mark may need some bound restriction in order to be between acceptable values. Finally, we can just recommend the *N* masters' courses with the best-predicted marks.

3. EXPERIMENTAL RESULTS

We tested our approach with data from a bachelor and a masters program at Instituto Superior Técnico, Universidade de Lisboa, in Portugal. This dataset describes 9149 courses' results achieved by 251 students on both bachelor and masters. The marks scale goes from 0 to 20, where 10 is the minimum grade that a student must achieve to be approved on any course.

To evaluate our results we follow the belief that the overall quality of the recommendations, independently of the method used to produce them, depends a lot on the quality of our predictions. To have a comparison to our grades' predictions results we used two baselines. The first sets the predicted mark of each student as his average mark on bachelor. The second baseline uses the average mark achieved in each masters' course. To do our prediction experiment, we started by constructing the 251 x 94 students-courses matrix *R*. We then applied Funk's SVD to produce both the students and courses features spaces and predicted every student's marks on all masters' courses. The achieved results in terms of the Mean Absolute Error (MAE) can be seen on Figure 1, and it is clear that our SVD approach has a smaller error than any of the baselines. In average, our predictions are 1.97 points deviated from the real mark, while both baselines have error values near 2.15. Hence, our predictions sustain an above average basis from where to recommend masters' courses to students.

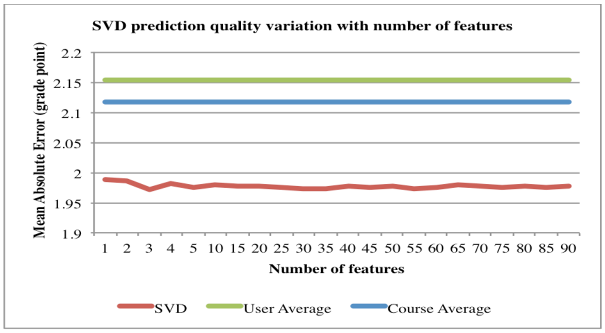


Figure 1 –MAE with the variation of the number of features and comparison with baselines

We did another experiment to see how the recommendations may affect students' masters' average mark. In this case we also used two baselines approaches: one recommends the most frequented courses in the historical training data while the other recommends the courses with best average mark on the same data. Table 1 shows the average mark of followed recommendations on the test data for all the approaches. We can see that the average mark achieved with the recommendations of our SVD approach is better than any of the baselines.

Table 1 – Average grade on followed recommendations.

Best Grades	Most Popular	SVD
13.6	13.4	14.6

ACKNOWLEDGMENTS

This work is supported by Fundação para a Ciência e Tecnologia under research project *educare* (PTDC/EIA-EIA/110058/2009).

REFERENCES

[1] Romero, C. and Ventura, S. 2010. Educational Data Mining: A Review of the State of the Art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40, 6, 601-618.

[2] Koren, Y., Bell, R., and Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42, 30-37.

[3] Funk, S. *Netflix Update: Try this at home*. http://sifter.org/~simon/journal/20061211.html, 2006.

The predictive power of SNA metrics in education

Diego García-Saiz
Department of Mathematics, Statistics and Computation
University of Cantabria
garciasad@unican.es

Camilo Palazuelos
Department of Mathematics, Statistics and Computation
University of Cantabria
camilo.palazuelos@unican.es

Marta Zorrilla
Department of Mathematics, Statistics and Computation
University of Cantabria
zorrillm@unican.es

ABSTRACT

Nowadays, centrality measures from social network analysis are being used for discovering underlying relationships among different actors or elements whose connections can be modeled as a graph. Their application in the educational domain has not been studied thoroughly, but the information that these metrics provide, as well as their predictive power, justify their use to model the students' social profile, as this paper shows.

Keywords

SNA, centrality measures, student performance

1. INTRODUCTION

Technology has provoked a great change in the way of teaching and learning. Its use in classrooms has supposed a change in the learning paradigm, from a teacher-centered model to a student-centered one, boosting students to construct their own knowledge whilst only being guided by the teacher. Also, social media is currently contributing, in an informal way, to train new skills to look for information, discuss, and elaborate new knowledge collaboratively. Therefore it is conceivable that, in a not very far future, social networks might be re-thought as a support for learning [2].

This has led us to study metrics that help us measure student social behavior in order to assess their predictive power to build student performance classifiers. In particular, we review centrality measures provided by social network analysis (SNA) and evaluate those that are suitable to build the student social profile and those that contribute to achieve better student performance models.

2. METHODOLOGY

First of all, we extracted all measurable activity variables for each student from the Moodle database. Next, we built a social network with the answers given in the forums. That means, we designed a graph in which students and instructors were defined as nodes and the answers given to questions or answers written by students were gathered as directed and weighted edges, being the weight the number of times that a student answers questions initiated by the student to whom is connected. Secondly, we utilized the ORA social network application [1] to calculate 10 node level SNA metrics [3]. Next, we interpreted the meaning of these SNA metrics in the educational context and associated each one to a social behavior. Then, we added them to previous datasets and carried out a feature selection process using Weka. This process allowed us to find out the subset of input variables of each dataset that had relevant predictive information for the pursued objective.

3. DATASETS

For this study, we chose two virtual courses taught at the University of Cantabria, entitled "Calculus" and "Gender Equality in Institutions" (GEI) with 115 and 48 students enrolled, respectively. These were selected because the activity in the forum was remarkable. In the former, the forum was used by students to ask for help and suggestions to their peers and instructor, and in the GEI course, it was used as a discussion tool.

4. RESULTS AND DISCUSSION

Table 1 shows the top 5 scoring nodes side-by-side for those centrality measures of the "Calculus" course. From its analysis, we can discover the different educational roles and the social behavior of each individual.

Node 3254 leads almost all the ranking categories. This is usually associated with the instructor of the course since it is often the one that starts the threads and makes others intervene. Node 5036, though not as prominent as 3254, can be thought as a co-instructor or a teaching assistant: it asks and responds noticeably (high indegree and outdegree), happens to be well-connected (high betweenness and low closeness) and seems to be both an authority and a hub. Node 4046 also presents an interesting behavior: somewhat in between the teacher mode of operation and the students' one. It is an active node (high degree) but its reputation is not as high as the previous nodes' (low eigenvector, hub, and authority values). It could be a not-so-active co-instructor or a highly collaborative student.

Regarding the students' behavior, node 12722 seems to have a remarkable attitude towards the course as a student. It answers a high amount of questions (high outdegree) and

Table 1: Top 5 scoring nodes for the centrality measures

Degree	Indegree	Outdegree	Eigenvector	Closeness	Information	Betweenness	Hub	Authority
3254	3254	3254	3254	3254	3254	3254	12722	3254
5036	4046	5036	12722	5748	5036	4046	6837	6826
4046	5036	12722	6837	5036	12722	5036	5036	5036
12722	6837	4046	5036	4046	4046	6685	5077	6821
6837	5630	6837	5077	6821	6837	5630	6821	6837

its responses are of great value (high hub). This permits it to reach a majority of the people through its answers (high eigenvector). It does not seem to ask too much. It would rather respond than ask. Finally, node 6837 presents the behavior of a good student, but participating in the forum in a radically different way than the previous node. This node tends to ask more than answer (high indegree) and the questions it makes are appreciated by the rest of students (it is in the top 5 authority ranking).

We used ClassifierSubSetEval and SubSetEval techniques for the feature selection process. Both were run using 10-cross fold validation. Therefore, the relevance of each attribute for the classification task is measured from 0 to 10 (no relevant to highly relevant). Table 2 shows the attributes selected by both techniques in the GEI course. The relevance of some SNA metrics is exceptional, such as degree and hub (students who answer those who receive many answers, learn, and have a higher probability to pass), as well as activity metrics, such as the number of initiated discussions in the forum and the number of visits to the course resources, though to a lesser extent.

Table 2: SubSetEval in the GEI course

	Attribute	Relevance (0-10)
SubSetEval	Degree	7
	Hub	3
ClassSubSet NaïveBayes	Hub	10
	Authority	3
	InformationCentrality	6
	ClickMembership	4
	DegreeClustering	3
	N_initiated_discussions	3
	N_views_resources	9
ClassSubSet J48	Hub	5
	ClickMembership	3
	Betweeness	3
	N_Initiated_discussions	4
	N_read_discussions	3

Table 3 displays the result obtained in "Calculus" course. It can be observed that the SNA measures have also an outstanding importance in order to build the prediction models.

Finally, we built classification models from these datasets with and without including SNA measures as attributes. Table 4 shows the accuracy (Acc.), sensitivity (Sens.) and specificity (Spec.) of the models built, as well as the improvement obtained using SNA attributes.As can be observed, the models obtained using SNA measures are more accurate in 75% of cases and present significant improvements. Due to the fact that students' involvement in the "Calculus" course is 40%, whereas in the GEI is 100%, it is to be expected that SNA measures show a higher predictive power in the latter, as it can be confirmed by our results.

Table 3: SubSetEval in the "Calculus" course

	Attribute	Relevance (0-10)
CfsSubSet	Degree	4
	N_attempts_quizzes	6
ClassSubSet NaiveBayes	Eigenvector	3
	Authority	4
	ClusteringDegree	3
	N_read_discussions	8
	N_view_resources	8
	N_attempt_quizzes	9
ClassSubSet J48	Betweeness	3
	N_view_resources	7
	N_read_discussions	7
	N_attempt_quizzes	8

Table 4: Accuracy, Sensitivity and Specificity obtained with J48 and NaïveBayes in both courses

			SNA	No SNA	Improv.
GEI	J48	Acc.	76.74%	62.79%	13.95%
		Sens.	76.9%	65.4%	11.5%
		Spec.	76.5%	58.8%	17.7%
	NB	Acc.	51.16%	48.84%	2.32%
		Sens.	57.7%	53.8%	3.9%
		Spec.	41.2%	41.2%	0.0%
"Calculus"	J48	Acc.	76.52%	70.43%	6.09%
		Sens.	86.0%	80.2%	5.8%
		Spec.	48.3%	41.4%	6.9%
	NB	Acc.	64.34%	65.21%	-0.87%
		Sens.	82.6%	83.7%	-1.1%
		Spec.	10.3%	10.3%	0.0%

These results allow us to conclude that SNA measures, extracted from the interactions of the students in forums from e-learning courses, are very informative to predict the students' performance and help to improve the classification models. Of course, the more the forum is used in a course, the more useful SNA measures are for this purpose.

5. REFERENCES

[1] K. Carley, J. Pfeffer, J. Reminga, J. Storrick, and D. Columbus. Ora user's guide 2013. Technical Report CMU-ISR-13-108, Carnegie Mellon University, 2013.

[2] C. Greenhow. Online social networks and learning. *On the Horizon*, 19(1):4–12, 2011.

[3] M. Newman. *Networks: an introduction.* Oxford University Press, 2010.

Data-Driven Curriculum Design:
Mining the Web to Make Better Teaching Decisions

Antonio Moretti[†], José P. González-Brenes[⋆], Katherine McKnight[†]
[†]Center for Educator Learning & Effectiveness
[⋆]Center for Digital Data, Analytics & Adaptive Learning
Research & Innovation Network, Pearson
{antonio.moretti, kathy.mcknight, jose.gonzalez-brenes}@pearson.com

ABSTRACT

University professors of conventional offline classes are often experts in their research fields, but have little training on educational sciences. Current educational data mining techniques offer little support to them. In this paper we propose a novel algorithm, Analyzing CurrIculum Decisions (ACID), that leverages collective intelligence to model student opinions in order to help instructors of traditional classes. ACID mines publicly available educational websites, such as student ratings of professors and course information, and learns student opinions within a statistical framework. We demonstrate ACID to discover patterns in learner feedback and factors that affect Computer Science instruction. We investigate the choice of a programming language for introductory courses and the grading criteria for all courses.

Keywords

offline teacher support, web mining, collective intelligence

1. INTRODUCTION

University professors of conventional offline classes are often experts in their research fields, but have little training on educational sciences. For example, studies have identified a lack of pedagogical training preparing research-based graduate students to teach in higher education [3]. It is not clear how existing educational data mining technologies can utilize the power of internet to learn student opinions in order to support traditional offline instructors.

We propose a novel algorithm, *Analyzing CurrIculum Decisions*(ACID), which is able to discover the effect of teaching decisions in the classroom by mining the increasing amount of information available online from educational websites. ACID develops resources and scripts to make use of collective intelligence and leverages this hierarchy of information within a statistical framework. ACID supports instructors of traditional offline courses by extracting from the web teaching syllabi data, and using crowd-sourcing to pair it up with students' course ratings and opinions to analyze the relationship between the two.

Algorithm 1 ACID pseucode

```
$n$ universities to analyze, $z$ reviews to analyze
procedure ACID
    while $|R| < z$ do
        $s \leftarrow$ sample of $n$ universities
        $s \leftarrow$ Remove non-English speaking universities
        $R \leftarrow$ Search_The_Web_For_Reviews($s$)
        $R \leftarrow$ ratings rated by more than $\epsilon$ students
    $Q \leftarrow$ CrowdSource_Questionnaire($R$)
    Analyze_Data($Q$)
```

This paper reports a case study of using the ACID methodology to answer questions that instructors of computer science courses face when designing their courses:

1. **For introductory classes, which programming language do students associate with clearer instruction?** The choice of a first programming language likely affects students' decision to continue education within the field of computer science. It is thus valuable to model data capturing learner sentiment.
2. **What grading rubric do students associate with clearer instruction?** Instructors want to optimize their grading criteria with respect to student learning and the student experience. The question of how to implement a grading rubric determines what students focus on within a course.

2. ANALYZING CURRICULUM DECISIONS

We use publicly available self-selected ratings of professors from a third-party website, *Rate My Professor* [4]. This site allows students to rate the professors and the courses they have taken. The website contains data from over 13 million ratings for 1.5 million professors. They collect ratings on a 1—5 scale (being 1 the lowest possible score, and 5 the highest) under the categories of "easiness", "helpfulness" and "clarity."' Additionally students may fill out an "interest" field in which they indicate how appealing the class was before enrolling, and a 350 character summary of their class experience. We focus on perceived clarity because of the direct link between clarity and quality of instruction.

We first select a random sample of 50 international universities that teach Computer Science from the Academic Ranking of World Universities [2]. From our sample of 50 univer-

Table 1: Programming Language Statistics

	Value	Std.Err	t-value	Pr<|t|	n
C	3.38	0.32	10.58	0.0000	109
C++	3.30	0.31	10.65	0.0000	214
Java	3.62	0.19	19.33	0.0000	353
Python	3.70	0.26	14.50	0.0000	133
Scheme	4.06	0.47	8.61	0.0000	32
Scratch	3.91	0.84	4.67	0.0000	49

Table 2: Grading Criteria Statistics

	Clarity	Std.Err	t-value	Pr<|t|	n
Exam Heavy	3.23	0.12	26.91	0.000	726
Equal Mix	3.52	0.14	26.04	0.000	484
Exam Proj	3.65	0.13	27.76	0.000	610
Exam HW	3.12	0.13	23.53	0.000	415

sities, 41 universities are English speaking. The nine non-English speaking universities are removed from our sample. We scrape and parse the reviews of the ratings website for all professors within the computer science departments of the universities in our sample. We remove the ratings from faculty that were rated by fewer than 30 students. This narrows our final sample to 10,655 different reviews of 180 different professors teaching 1,112 courses at 22 universities.

We use Amazon Mechanical Turk [1], a crowdsourcing platform, to find course features for each of the courses in our ratings sample. We do this by asking respondents to fill out a survey. The survey requests to find the online syllabus that corresponds to the course and professor from which we have ratings that is closest to the date of the student review we collected.Then, using the syllabus, respondents are asked to to provide the programming language(s) used, the textbook(s) used, and the percentage of the grade that was determined by homework, projects, quizzes, exams.

From our original sample of 1,112 courses taught by a unique professor, respondents find an online syllabus matching the professor for 342 courses (∼31%). We hypothesize three explanations for the missing syllabi: (i) the syllabi may be accessed only with a password through a course management system, such as blackboard, (ii) the syllabi may not be available only, or (iii) the respondents are not able to find the syllabi.

3. LEARNING STUDENT OPINIONS

We make use of the ratings and syllabi data collected to provide insights into which programming languages beginning students associate with clear instruction. We filter the data to only include introductory level courses (one which does not require any prerequisite coursework in computer science). Our restricted sample includes 1024 reviews. We explore the relationship between clarity ratings and programming language using general linear mixed modeling with random professor and course effects. We do not report programming languages with less than 30 student reviews. Table 1 summarizes the perceived clarity of courses by programming language (higher is better). An intercept is not modeled in order to make the results easily interpretable. The mean clarity rating for introductory courses is 3.599.

We found C and C++ had the lowest coefficients (i.e. compiled languages were less clear). Observe that Scheme and Scratch have the highest clarity ratings followed by Python and Java. We note that the standard errors are smallest for Java and Python and largest for Scheme and Scratch. There is more variation in reviews of courses using Scheme and Scratch than there is for courses using Java and Python. Students in our sample associate clearer instruction with interpreted languages rather than compiled languages.

To assess students' course ratings of clarity based on the percentage of the grade due to exams, quizzes, homework and projects, we created a factor made up of four clusters representing four ways of weighting homework, projects, exams, quizzes and miscellaneous (such as extra credit) for the students' grade. We sort the data to only include observations in which the grading criteria is available and sums to 100. There are 2225 observations with full grading criteria. We use k-means clustering to partition the 2225 observations with complete grading criteria information based on the five aforementioned variables. We optimize our number of clusters by examining how the BIC and AIC of the mixture model change based on the number of clusters selected. A four cluster solution optimizes the AIC and log-likelihood of the model. The cluster membership is modeled using random professor and course effects.

The exams and projects cluster has the highest estimate of clarity. We find that weighting projects equally with exams is associated with a clearer course experience. The equal mix cluster also is associated with higher clarity estimates. The exam heavy cluster and the exam and homework heavy clusters are associated with lower student clarity ratings. We find that a rubric that weights exams and projects evenly is correlated with clearest instruction.

4. CONCLUSIONS

We demonstrate how the Analyzing CurrIculum Decisions (ACID) methodology can be used to leverage collective intelligence and learn student opinions. In introductory computer science courses, we find that students that are taught interpreted languages find their classes clearer. We also that find students who are given an even weighting of exams and projects find their classes clearer. Our study does not necessarily suggest that teachers should change their programming language. Further research is needed before drawing causal inferences. Student evaluations often include free form text where students can describe their experience in the course. One extension is to regress text sentiment on course features. ACID is a useful tool to discover patterns in student opinions. Syllabus data and course ratings data are becoming increasingly available on the Web. This data is used by millions of students and worthy of further research.

5. REFERENCES

[1] Amazon. Mechanical turk, 2014.

[2] MultiMediaLLC. Academic ranking of world universities, 2013.

[3] T. Robinson and W. Hope. Teaching in higher education: is there a need for training in pedagogy in graduate degree programs? *Research in Higher Education Journal*, 2013.

[4] J. Swapceinski. Rate my professor, 2014.

Towards IRT-based student modeling from problem solving steps

Manuel Hernando
Universidad de Málaga
Bulevar Louis Pasteur, 35.
29071 Málaga, Spain
(+34) 952 132 863
mhernando@lcc.uma.es

Sergey Sosnovsky
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
(+49) 681 85775 5367
sergey.sosnovsky@dfki.de

Eduardo Guzmán
Universidad de Málaga
Bulevar Louis Pasteur, 35.
29071 Málaga, Spain
(+34) 952 137 146
guzman@lcc.uma.es

Eric Andres
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
(+49) 681 85775 5367
eric.andres@gmail.com

Susanne Narciss
Technische Universität Dresden
Psychologie des Lehrens und Lernens
01062 Dresden, Germany
(+49) 351 4633 6059
susanne.narciss@tu-dresden.de

ABSTRACT

In this research, we use Item Response Theory based model for computing procedural knowledge of a sample of primary school children solving fraction addition exercises. For each exercise, the model needs to automatically construct a solution graph. We have explored different strategies for building such graphs and the effects they have on the quality of the model predictions. The results obtained shed light on the applicability of Item Response Theory for the task of measuring procedural skills and provide recommendations on the choice of IRT model adjustment.

Keywords

Student Modeling, Item Response Theory, Problem Solving, Procedural Knowledge.

1. INTRODUCTION

Intelligent tutoring systems (ITS) are designed to provide individualized computer-supported learning. One of the most important characteristics of a good ITS is a high-quality student modeling component, that maintains representation of student knowledge and helps the ITS to support personalized tutoring helping each student improve her knowledge in the optimal way.

High-quality student modeling starts with accurate knowledge assessment. The classical approach to infer procedural knowledge is based on exposing students to problem solving, as it is the most natural way for a student to demonstrate procedural skills.

In the field of testing, Item Response Theory (IRT) [2]is known to provide accurate and invariant measurement of declarative knowledge. In [5], we have proposed a model that employs IRT for procedural knowledge assessment and can be used in problem solving environments. As a part of this approach, dynamic problem solution graphs are automatically constructed from student logs. Such graphs are updated and improved every time a new student interaction with a target exercise has been registered. The work presented in this paper explores different alternatives for constructing the graphs, and analyses how various evidence aggregation techniques influence the quality of the resulting IRT models and the accuracy of knowledge assessment they support.

2. PROCEDURAL ITEM RESPONSE THEORY

There are three types of IRT-based models, according to how they score student responses to the test items (questions) and update student knowledge [3]: dichotomous, polytomous, and quasipolytomous models. Dichotomous models consider only two scores per item (correct/incorrect); polytomous models assume different scores for different answers, thus, being more informative than dichotomous models, but requiring more data to calibrate [2]; quasipolytomous models [4] are halfway between dichotomous and polytomous: some possible answers have their own scores and others are clustered into aggregate options.

The process of solving a multistep learning problem can be represented as a graph that contains all the steps and actions a student could perform, where nodes correspond to the states of the solution process and the arcs to the actions of a student transitioning her from one state another. In this work, instead of using pre-constructed graphs, we data-mine individual problem solution graphs from the student activity logs.

The procedural IRT mode makes an analogy between problem solving and testing by considering a students' path through the process of solving a multistep learning problem as a testing sequence. Each node could be understood as an item and each step as an item response.

3. PROBLEM SOLVING ENVIRONMENT AND DATA USED

The data used in this study comes from the controlled experiment conducted in Spring of 2012 in Dresden (Germany) with 6th- and 7th-grade pupils. Students had to solve simple fraction problems in the computer-based learning environment ActiveMath [6][7]. The overall experiment contained several phases and covered several topics of fraction arithmetic. In this paper, we have focused on multistep problems on "*Adding Fractions with Unlike Denominators*" that students were solving during the posttest phase of the experiment. After filtering out subjects who did not manage to try the target set of problems, we have 61 students (25 males and 36 females) contributing to the final dataset.

The problems were based on an interface allowing students to construct individual solution paths by providing structured templates for intermediate steps [1]. While solving a problem, a student could choose a type of the operation to perform on the next step and then fill in the corresponding template. Only students defined the number and sequence of steps that they needed to reach the final solution.

4. EXPERIMENTAL MODELS

We have explored three different strategies to generate problem solution graph from the log data. First, we have applied our approach in a straightforward way – by generating one graph per problem without any aggregation and applying the IRT to this graph. We have called this model *Direct Application* (DA). The second model seeks to increase the supporting evidence per single steps by merging the states that represent the same semantic operation in a problem solution graph. We call this model *Semantic Operation* (SO). Finally, the *Common Graph* (CG) model logically develops the approach of the SO model by aggregating semantically equivalent operations across problems. As a result, a single graph is constructed to represent the entire subset of isomorphic problems related to "*Adding Fractions with Unlike Denominators*".

5. EVALUATION

We have used two sets of problems in this research: the *target set* consists of three multi-step problems on adding fractions with unlike denominators; the *assessment set* contains 13 one-step problems on fraction expansion, fraction reduction and adding fractions with a common denominator.

In order to evaluate the quality of each model in terms of its predictive validity, we compare the obtained estimates with the knowledge scores students achieve on the *assessment problem set*. These scores are also computed using the IRT approach. Each of the 13 *assessment set* problems is a single-step problem, therefore it corresponds to a single test item. We have looked into which model produces better predictions of student knowledge assuming that a better model will be closer to the control assessment.

We have used different quasipolytomous models depending on the supporting threshold of arcs (understanding threshold as the minimum acceptable support of steps) being a threshold = 1 a pure polytomous model and the maximum threshold a pure dichotomous.

Table 1 shows the results of our experiments, the two columns contain the maximum and the minimum values for Pearson's correlation (*r*). The values depend on the support threshold chosen for a particular quasipolytomous IRT setup as described above. Essentially, all models produce knowledge predictions that are significantly positively correlated with the controlled assessment. In all three cases, the maximum correlation effect size is rather high; however, the difference between the straightforward *DA* model and the SO/CG models semantically aggregating students' results is considerable.

Table 1. Correlations of the scores on the assessment test and the target tests produced by the experimental models

Model / Test	r_{max} (threshold)	r_{min} (threshold)
DA	.42 (9)	.27 (15)
SO	.54 (5)	.34 (21)
CG	.51 (3)	.35 (90)

6. CONCLUSION

In this paper we have studied different strategies to elicit the problem solving graph for assessing the student procedural knowledge with an IRT-based model. We have distinguished three different strategies: building a graph directly from student behavior graph, building the graph grouping states by semantic operations, and building a graph that represents more than a single problem. Results suggest that all of the strategies could be valid to infer procedural knowledge but we get better results when we group some states. However, when we use the same graph for more than a problem we have not obtained any advantage, even SO model obtains better results.

The use of IRT in a problem-solving environment for assessing procedural ensures that the results obtained are invariant and well-founded, since they are computed using data-driven statistical procedures. Results of our work are promising but we should to test them for larger student samples.

REFERENCES

[1] Andres, E., Sosnovsky, S., Schnaubert, L., Narciss, S. 2013. Using Fine-Grained Interaction Data to Improve Models of Problem Solving. In proceedings of *DAILE Workshop*, 4th STELLAR Alpine Rendez-Vous.

[2] Embretson, S.E., Reise, S.P. 2000. Item response theory for psychologists. Lawrence Erlbaum, Mahwah.

[3] Guzmán, E., Conejo, R., de-la Cruz, J.L.P. 2007.Adaptive testing for hierarchical student models. *User Model. and User-Adapt. Interact.*, 17, 119-157.

[4] Hernando, M., Guzmán, E., Conejo, R. 2013. Validating Item Response Theory Models in Simulated Environments. In *Proceedings of the AIED Workshop on Simulated Learners*, 41-50. Memphis, TN, U.S.A.

[5] Hernando, M., Guzmán, E., Conejo, R. 2013. Measuring Procedural Knowledge in Problem Solving Environments with Item Response Theory. In *Proceedings of the 16th Int. Conf. on AI in Education*, 653-656. Memphis, TN, U.S.A.

[6] Melis, E., Goguadze, G., Homik, M., Libbrecht, P., Ullrich, C., Winterstein, S. 2006. Semantic-aware components and services of ActiveMath. *Brit. J. Educ. Technol.*, 37(3), 405–423.

[7] Sosnovsky, S., Dietrich, M., Andrès, E.,Goguadze, G., Winterstein, S., Libbrecht, P., et al. 2013. Math-bridge: bridging the gaps in European remedial mathematics with technology-enhanced learning. In Using tools for learning mathematics and statistics, 437–451. Berlin: Springer.

Towards Uncovering the Mysterious World of Math Homework

Mingyu Feng
SRI International
333 Ravenswood Ave
Menlo Park, CA 94025 USA
1-650-859-2756
mingyu.feng@sri.com

ABSTRACT

Homework has been a mysterious world to educators due to the fact that it is hard to collect data with regard to homework behaviors. Little is known about when a student works on homework, how long it takes him to complete the homework, how much time he spends on a problem and whether and where he has struggled, etc. Such information not only have implications on a student's performance level on assigned skills, but also are potential indicator of his non-cognitive status, such as engagement with homework and whether he was persistent. In this paper, we present our initial effort to uncover the mysterious world through exploratory analyses of the system logs from the ASSISTments platform when 690 7th grade students in the state of Maine did their math homework in the system.

Keywords

Homework, math, online tutoring.

1. INTRODUCTION

Homework is a well-established practice in schools, despite all the controversial discussion regarding its influence on learning (Kohn, 2006), and the research knowledge base for the effectiveness of homework is also well established (Cooper et al., 2006). Yet, without explicit interventions, homework has been commonly underutilized for improving teaching and learning. Educational technologies have gained popularity in schools (e.g., Khan Academy, DreamBox, IXL.com), but not at home. Most of the computer programs for homework are for college-level populations (e.g., WebAssign, Mastering Physics, OWL), but not in K-12 settings. Homework has been a mysterious world to educators partly due to the fact that it is hard to collect data. However, information from homework, such as when a student works on homework, how long it takes him to complete the homework, how much time he spends on a problem and whether he has struggled, has not only implications on a student's performance level on assigned skills, but also is potential indicator of his non-cognitive status, such as engagement with homework and whether he was being persistent.

2. BACKGROUND

ASSISTments (www.assistments.org) is an online tutoring system that provides "formative assessments that assist." Teachers choose (or add) homework items in ASSISTments and students can complete their homework items online. As students do homework in ASSISTments, they receive feedback on the correctness of their answers. Some problem types also provide hints on how to improve their answers, or help decompose multistep problems into parts. Teachers receive reports on their students' homework and can use this information to organized more targeted homework reviews, to assign specific follow-up work to particular students, and to more generally adapt or differentiate their teaching.

Prior research also has established the promise of ASSISTments for improving student outcomes in middle school mathematics through homework support (Mendicino et al., 2009; Singh et al., 2011; Kelly et al., 2013). Building on this prior work, a large-scale efficacy study is being conducted with ASSISTments in the state of Maine where a one-to-one laptop program was well established, to evaluate the efficacy of ASSISTments for online homework support. This randomized controlled trial involves 45 middle school schools that were randomly assigned to treatment or control (i.e. "business as usual") conditions. The intervention is implemented in Grade 7 math classrooms in treatment schools over 2 consecutive years. In the treatment condition, teachers receive professional development and use ASSISTments to assign homework for their students during the school year.

3. METHOD

3.1 Data

For this study, we collected homework log of 690 7th grade students from classes of 17 teachers in 9 middle schools that participate in the efficacy study. The data set includes 779 homework assignments made by the teachers during January and February 2014. These students have been using ASSISTments to do their homework since the beginning of the school year and their teachers started using ASSISTments since September 2012. We excluded the problems that took students over 10 minutes to complete, considering students were likely to be off-task and thus the measure of completion time might not accurate. On average, each student solved 181 problems, and the number varies a lot among students (standard deviation = 163). In addition to student homework log, we also collected teacher's usage data, in particular, when they have opened a report provided by ASSISTments.

Based on the student log and teacher usage data, we calculated the following metrics

- %Correct—student's average percent correct on all problems in an assignment

- AvgAttempt—the average number of attempts [1] a student made on a problem in an assignment
- AvgFirstResponseTime—the average amount of time it took a student to respond to a problem in an assignment
- AvgTotalTime—the average total time it took a student to complete a problem in an assignment
- StartHour—the hour of the day when the student started working on an assignment
- CompletionIndicator—whether an assignment was completed on time, late or not completed.
- CompletionRate—a student's overall homework completion rate during the time period
- %ReportOpening—a teacher level metric, the percentage of assignments for which a teacher has opened related ASSISTments reports. For example, if a teacher has made 10 homework assignments to her students, but only looked at reports for 4 of the assignments, then %ReportOpening will be 40%.

3.2 Analysis and Findings

Our analysis was mostly exploratory. First, we plotted the data (see Figure 1) to see when students started working on homework, and if there is any association between when a student started and whether the assignment was completed on time or not. We observed that for the 8573 instances of assignments that were completed on time, most of the time students started around 11am, or 12pm, or early in the morning at 9am. The assignments that were not completed tended to start a bit later at 1pm or 10am.

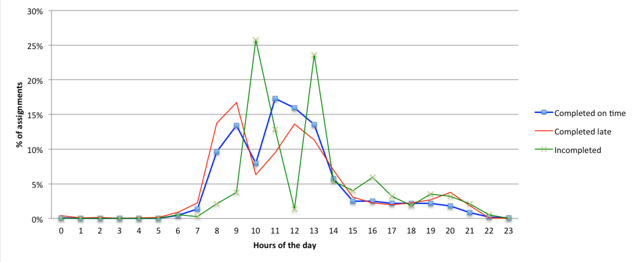


Figure 1. Time students start working on homework

Then we looked to see whether there was any difference in student's performance or behaviors when they completed homework assignments on time or not. We found that for assignments that were not completed, students were significantly (unpaired t-test, p < .01) low on %Correct metric, yet high on AvgFirstResponseTime, and AvgTotalTime, comparing to their performance on assignments that were completed on time, indicating students were struggling with the problems in those assignments. Meanwhile, students were also significantly low (upaired t-test, p < .01) on AvgAttempt, suggesting they were not as persistent when trying to solve the problems.

Teacher's review of homework performance report is a critical step in the ASSISTments logic model and teachers are encouraged to look at the reports to direct their homework review with students and adapt their instructions. During the interviews (another data collection activity of the efficacy study), teachers indicated homework review time has been largely reduced because of that the ASSISTments reports have made the review more targeted. While we don't have the classroom observation data yet, we consider %ReportOpening as an indicator of how often the homework review was done. We discretized %ReportOpening into 3 bins: low, medium and high, and aggregated other metrics across students within each bin. We found in the bin where %ReportOpening was low, students' average %Correct and CompletionRate were significantly higher yet AvgFirstResponseTime and AvgTotalTime were all significantly lower, comparing to those for the "high" bin. While this finding was against our initial instinct, it is too early to draw any conclusion regarding a casual relationship between teacher's review practices and student's homework performance from this, given that the analysis wasn't tracking changes in the same teacher's classes longitudinally, and didn't account for any incoming homework performance data of the students (e.g. homework completion rate, %Correct, etc). Teachers who knew their students had problems with completing homework may choose to look at reports more often to monitor student's progress.

4. CONCLUSION

In this paper, we presented some initial results from analyzing student homework logs and teacher's usage of an online homework support program as a part of an efficacy study. The analyses here represent the beginning of our efforts to understand the world of homework. In the future, we plan to link student's homework log data with their unit test scores (as proximal measure of their knowledge) and end of year standardized test scores to investigate the relationship between homework and learning outcomes. We also plan to analyze student homework log data, teacher's report usage data and test scores together longitudinally and triangulate the results with findings from field classroom observations to further investigate the impact of teacher's review practices on student's learning outcome and on how students do their homework.

5. ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Educational Sciences, U.S. Department of Education, through Grant R305A120125 to SRI International. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

6. REFERENCES

[1] Cooper, H., Robinson, J., & Patall, E. (2006). Does homework improve academic achievement? A synthesis of research, 1987-2003. *Review of Educational Research, 76*, 1–62.

[2] Dufresne, R., Mestre, J., Hart, D. M., & Rath, K. A. (2002). The effect of web-based homework on test performance in large enrollment introductory physics courses. *Journal of Computers in Mathematics and Science Teaching, 213*, 229–251.

[3] Kelly, Y., Heffernan, N., Heffernan, C., Goldman, S., Pellegrino, G. & Soffer, D. (2013). Estimating the effect of web-based homework. In Lane, Yacef, Motow & Pavlik (Eds) *The Artificial Intelligence in Education Conference* (pp. 824-827). Springer-Verlag.

[4] Kohn, A. (2006). *The homework myth: Why our kids get too much of a bad thing*. Cambridge, MA: Da Capo Press.

[5] Singh, R., Saleem, M., Pradhan, P., Heffernan, C., Heffernan, N., Razzaq, L., & Dailey, M. (2011). Improving K-12 homework with computers. In *Proceedings of the Artificial Intelligence in Education Conference,* Auckland, New Zealand.

[1] In ASSISTments, students are allowed to make multiple attempts at problems until they solve the problem correctly.

Towards Using Similarity Measure for Automatic Detection of Significant Behaviors from Continuous Data

Ben-Manson Toussaint
Informatics Laboratory of Grenoble
11, rue des Mathématiques
38400 St-Martin d'Hères
+33 4 76 57 48 61
ben-manson.toussaint@imag.fr

Vanda Luengo
Informatics Laboratory of Grenoble
11, rue des Mathématiques
38400 St-Martin d'Hères
+33 4 76 57 47 75
vanda.luengo@imag.fr

Jérôme Tonetti
University Hospital of Grenoble
Boulevard de la Chantourne
38700 La Tronche
+33 4 76 76 66 06
j.tonetti@chu-grenoble.fr

ABSTRACT

This paper presents our method based on similarity measure between contiguous pairs of sequences to yield automatic detection of significant behaviors from raw and continuous traces. The traces, produced by a simulation-based Intelligent Tutoring System dedicated to percutaneous orthopedic surgery, are related to perceptual-gestural behavior and ill-defined tasks involved in this domain. Preliminary qualitative evaluations have been conducted on real data from five simulation sessions and showed the relevancy of our method and adjustments that need to be realized for further experiments.

Keywords

Sequences similarity, Perceptual-gestural behavior, Ill-defined task, Simulation-based ITS, Learners modeling.

1. INTRODUCTION

The learning process of orthopedic surgery is composed of two parts: a theoretical part involving declarative knowledge and a practical part involving perceptual-gestural knowledge related to surgical gestures. This knowledge is qualified as perceptual-gestural because it is tacit and mostly accessible empirically through repeated practices. Tasks related to this knowledge are ill-defined as different strategy patterns can be applied to execute a given operation and no precise way can be defined in advance to satisfy their validation criteria. As demonstrated in [6], there is a gap in the learning process that can hardly be bridged by traditional teaching methods. TELEOS learning environment aims at providing the missed intermediate phase of apprenticeship.

For offering tutoring services in adequacy with perceptual-gestural and ill-defined knowledge, some constraints must be considered like the impossibility to define an exhaustive theoretical framework, the importance of designing an opened knowledge model and the difficulty to assess perceptual-gestural knowledge in the diagnosis process. To overcome these constraints, we want to set up a hybrid approach [2] combining a data-driven paradigm including automatic acquisition of knowledge from traces, with the existing expert-oriented paradigm. The purpose is to keep the knowledge model opened and incremental. To achieve this, we need to capture and model learners' strategies in the execution of simulated operations. That requires on first hand that we foster automatic detection of significant execution behaviors from the continuous raw traces recorded by the simulator.

2. BACKGROUND

The most recent related work reported in the literature is CanadarmTutor, a simulation-based ITS for training astronauts for the handling of an articulated robotic arm [4]. It provides a 3D simulated environment where leaners train in moving the robotic arm from an initial configuration to another predetermined one. As explained in [1] this task is complex and ill-defined.

For offering convenient tutoring services considering these constraints, a hybrid approach combining expert system, model-tracing and automatic acquisition of partial task from experts has been proposed [5]. Like in our case, this work seeks to extract parts of solution paths that are frequently applied to be reused later for supporting key tutoring services. One of the main differences between this work and ours is the importance in our case to link extracted resolution patterns with the phase in which they lie as some actions give different performance insights depending on the phase in which they were executed.

3. METHOD

To capture perceptual-gestural behavior in TELEOS learning environment, we use two complementary devices with the simulator: an eye-tracker for tracing perceptual behavior, that is, points and areas of interest gazed during the execution actions [3] and a haptic arm to capture gestures-related actions executed with the trocar [4]. (Traces from the three tools are recorded independently. They are heterogeneous regarding their content types, their content format and their time granularities. To link each sequence of action to the associated sequences from the complementary devices, we merge their parameters so that each sequence from one source contains the parameters of sequences from the two other sources at the moment it occurred. After this treatment, an action is represented by a subset of sequences that defines its continuum until the next action is executed.)

3.1 Characterizing Significant Behaviors

Our case study is centered on vertebroplasty[1]. This surgical operation is conducted in three phases: the patient preparation, the drawing of the cutaneous marks and the trocar insertion. As opposed to classical open-heart surgery, surgeons are guided all along the operation by X-rays. Validation criteria of each executed action are evaluated by visual analyses of these latter. The first phase of a vertebroplasty is validated if the X-ray appliance (the fluoroscope) is positioned as to generate both face and profile X-rays that render properly the position of the targeted vertebra. In the second phase, the cutaneous marks are validated if their drawing overhangs properly the targeted vertebra on the X-rays. The last phase is validated if the X-rays confirm the correct trajectory of insertion of the trocar.

[1] Vertebroplasty is a percutaneous orthopedic surgery that is practiced to treat fractured spine bones with cement injected with a trocar inserted through small incisions in the skin.

However, validated actions can need to be revised if not executed correctly. These corrections can take place within the same phase or can require that the intern returns to a previous phase. They can be the consequence of different behaviors. For example, the intern may not take enough time to analyze generated X-rays or not enough X-rays to guide his or her actions. On the other hand, another intern can often ask for visual guidance if his or her strategy is to progress with slight and prudent adjustments where another one can ask for very little visual guidance but take more time to analyze each generated X-ray.

Thus, we want to automatically detect, from the simulation traces, phase changes, corrections within the same phase and taken step back to decide on next action or to validate passed action. To achieve this, we need to identify the amplitude of displacements of the simulation environment tools step by step, that is, from one sequence to the next. We need also to identify the elapsed time between these sequences. In fact, similarity between sequences of the same action continuum is supposed to be high and important changes, marked by contiguous sequences with low similarity. The elapsed time between contiguous sequences gives also insights on the learners' behavior as it can point out the time taken for modifying an action, for thinking on the next action to execute or on the validation check of a passed action.

Based on observations of simulation sessions realized by interns at the university hospital of Grenoble, we made the assumption that small temporal gaps coupled with low similarities are more likely to represent phase changes; large temporal gaps coupled with low similarities are more likely to represent corrections within a phase and large temporal gaps coupled with high similarity are more likely to represent step back for deciding on next action or to check the validation criteria of passed action.

3.2 Computing similarity

We used the cosine similarity measure to calculate the similarity between pairs of contiguous sequences. This measure is given by the following formula:

$$\mathrm{sim}(A,B) = \cos(\theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \qquad (1)$$

This metric excerpt the proximity of the orientations of the vectors A and B based on the angle θ that they form and consequently, the level of similarity of the sequences that they represent. Its outcomes are bounded in [0,1]: 1, representing a perfect similarity between the elements of the two vectors and 0, a strong dissimilarity between them.

4. EXPERIMENTS

We conducted a preliminary qualitative experimentation based on traces from five simulation sessions of vertebroplasty. Each session was executed by one different intern surgeon at the University Hospital of Grenoble and was screen video recorded. We proceeded to the comparison of behaviors listed from videos with the list of automatically detected behaviors from the traces as to identify accurate, missed and false detections. As reported in Table 1, the automatic detection method demonstrated good performance for the detection of phase changes. Indeed, all of those that it reported were relevant, bringing the precision of detections for this category of behavior to 1.00. However, 60% of these changes were missed. This explains the poor recall score (0.40) for this category of behavior, as for the detections of corrections within phases for which the recall is only of 0.33. High precision (0.91) and recall (0.76) are recorded for the detections of taken step back for the five sessions.

F-Score Behaviors	Precision	Recall	F_1-score
Corrections within phases	0.60	0.33	0.42
Phase changes	1.00	0.40	0.57
Step back	0.91	0.76	0.83

Table 1. Measures of Precision and Recall of the automatic method compared to observations from video recorded simulation sessions.

5. CONCLUSION

These evaluations, specially the obtained recall scores, reveal the sensitivity of sequences similarity outcomes in presence of other factors that were not considered in this experiment like the level of experience or competence of the interns. The choice of similarity and temporal gap thresholds should be adapted regarding these factors in future evaluations. This work is the first step in achieving more fine-grained diagnosis by integrating in the process learners' simulation execution strategies along with evaluation of their single actions. The planned next step is to yield automatic recognition and categorization of significant behaviors signatures in addition to the mere detection of their occurrences.

6. REFERENCES

[1] Fournier-Viger, P., Nkambou, R., Mephu Nguifo, E. 2010. Building Intelligent Tutoring Systems for Ill-Defined Domains. In *Advances in Intelligent Tutoring Systems*, Springer, Heidelberg. 81–101

[2] Fournier-Viger, P., Nkambou, R., Mayers, A., Mephu Nguifo, E., Faghihi, U. 2011. A Hybrid Expertise Model to Support Tutoring Services in Robotic Arm Manipulations. In *Proceedings of the 10th Mexican International Conference on Artificial Intelligence*. MICAI 2011. LNAI 7094, Springer, 478–489

[3] Jambon, F., Luengo, V. 2012. Analyse oculométrique « on-line » avec zones d'intérêt dynamiques : application aux environnements d'apprentissage sur simulateur. In *Actes de la Conférence Ergo'IHM sur les Nouvelles Interactions, Créativité et Usages*, Biarritz, France

[4] Luengo, V., Larcher, A., Tonetti, J. 2011. Design and implementation of a visual and haptic simulator in a platform for a TEL system in percutaneous orthopedic surgery. In *Medecine Meets Virtual Reality 18*, Westwood J.D., Vestwood, S.W., Ed. MMVR 2011. 324–328

[5] Minh Chieu V., Luengo V., Vadcard L. 2010. Student Modeling in Orthopedic Surgery Training: Exploiting Symbiosis between Temporal Bayesian Networks and Fine-grained Didactical Analysis. In *International Journal of Artificial Intelligence in Education 20.* IJAIED 2010. IOS Press. 269-301

[6] Tonetti, J., Vadcard, L., Girard, P., Dubois, M., Merloz, P. Troccaz, J. 2009. Assessment of a percutaneous iliosacral screw insertion simulator. In *Proceedings of the Conference Clinical Orthopaedics and Related Research.* CoRR 2009. 471-477

Using Data Mining to Automate ADDIE

Fritz Ray
Eduworks Corporation
136 SW Washington Ave. STE 203
Corvallis, OR 97333
+1 (541) 753-0844 x308
fritz.ray@eduworks.com

Keith Brawner
U.S. Army Research Laboratory
HRED-STTC
Orlando, FL 32826
+1 (407) 380-4648
keith.w.brawner@us.army.mil

Robby Robson
Eduworks Corporation
136 SW Washington Ave. STE 203
Corvallis, OR 97333
+1 (541) 753-0844 x304
robby.robson@eduworks.com

Abstract:

The goal of this work is to transform informational and instructional content into adaptive and personalized training experiences. We have developed semi-automated methods to do this that parallel the traditional “ADDIE” (Analysis, Design, Development, Implementation, and Evaluation) process. The source content can include documents, presentations and manuals and existing courseware. The techniques use artificial intelligence (AI), data mining, and natural language processing and generally belong to the discipline of “educational data mining.” This poster/demo demonstrates the processes and discusses the algorithms used.

1. PROBLEM STATEMENT

Today’s digital environment is rich with learning content, but much of it is purely didactic in nature. This content includes manuals and presentations not intended for instructional purposes and e-learning that consists of presentations and lectures with multiple choice questions. As online learning replaces instructor-led training in corporations, government agencies, and educational institutions [10], its effectiveness can be improved by transforming this wealth of didactic content into more interactive and adaptive learning experiences [5].

Here, we address aspects this transformation problem in the context of multiple research and commercial projects. A large portion of the work we report here comes from a U.S. Army Small Business Innovation Research (SBIR) project called *Tools for the Rapid Generation of Expert Models*, or TRADEM, that applies data mining to (a) deconstruct existing content at a deep and granular level and (b) reconstruct it in a form that can be used to create adaptive intelligent tutoring systems. This process automates many steps in the “ADDIE” (Analysis, Design, Development, Implementation, and Evaluation) process [1] commonly used to develop instructional content.

1.1 Motivation

The three primary benefits of applying EDM to automate a process such as ADDIE are cost, speed, and the effectiveness of the training produced.

Data about e-learning development [3] shows that about 40% of the cost involves analysis and design tasks, which includes the expensive activity of engaging with subject matter experts. Using EDM to extract the domain analyses and instructional designs from existing content is more cost-effective than going through the entire ADDIE process each time instruction is developed. For example, in Army Civilian Affairs training using TRADEM, simulations provide experiential learning on how to conduct civilian affairs in current, real-world situations. The content changes frequently, requiring continual repetition of the ADDIE process. As a result, manual processes are too slow and too expensive, but automated the generation of up to date domain models, concept and skill maps, and instructional content allows the Army Civilian Affairs Corps to rapidly deploy new training in response to a real and changing world.

Providing highly effective training also drives the development of TRADEM. Classroom instruction and most existing e-learning falls far short of the effect sizes that have been shown to be achieved with *intelligent tutoring systems* [4; 13]. While TRADEM can be used to develop and implement many different types of learning environments, our work has focused on producing intelligent tutors.

2. DESCRIPTION OF TRADEM

ADDIE’s design step consists of determining learning objectives, sequencing instruction, and writing assessments. TRADEM automates this step via an assisted full workflow solution.

Workflow: First, TRADEM extracts a topic map from a user-input corpus of content. This topic map visualizes a set of topics that cover the core topics present in the input corpus. For each node (topic) in the concept map, TRADEM then selects the pieces (granules) of the initial input corpus most associated with that topic. Next, TRADEM builds an assessment for each topic based on the granules associated with that topic. On demand, these assessments are then exported in an intelligent tutoring format for use in instruction.

Topic Generation: TRADEM ingests a corpus of content and performs a front-end analysis that results in a concept map consisting of a directed tree of topics. The topics are extracted from the corpus using topic-detection techniques [9] that are applied as described in [12]. The number of topics generated optimizes coverage of the input corpus, but the user can alter the number of topics based on pedagogical needs. This is necessary in real-world applications. For example, the user may wish to match a list of topics that appear in standardized curricula.

To determine topic relationships and order, TRADEM calculates a relation strength for each pair of topics, creating a graph with relationship strengths between all topics. This fully connected graph is transformed into a directed tree spanning all topics by inferring directionality using a precedence metric and tree selection algorithm based on aggregate relationship strength. We interpret this tree as representing optimal learner paths between any two topics in the input corpus. This mirrors the way classical instructional designs progress through a subject, including intermediate learning objectives leading to a terminal learning target [12].

Content Granules: To identify the pieces of the input corpus most closely aligned with each topic in the topic tree, TRADEM decomposes the input corpus into *granules* of content. For standard text, these are paragraphs, while slides and bulleted lists may end up as single or multiple granules. A sentence parsing algorithm is used to selecting which sentences in the granule are best suited to generate assessment questions using assessment generation techniques based on the work of Mitkov, Ha, Heilman and Smith [8; 11]. These techniques produce template forms that can be transformed into essay or multiple choice questions using a manual process. Additionally, each granule is automatically tagged with suggested relevant instructional types. For example, the *Generalized Intelligent Framework for Tutoring* (GIFT) includes an *Engine for Macro- and Micro-Adaptive Pedagogy* (EMMAP), that recognizes four pedagogical strategies: Rule, Example, Recall, or Practice [5]. This allows granules associated with each topic to be selected by an intelligent tutor based on its pedagogical needs. Thus, extracted topics are associated with meaningful chunks of corpus content that become the basis for real instruction driven by an intelligent tutoring framework.

Tutor Implementation: The target intelligent tutor we currently produce is dialogue-based tutor that we call T-Tutor. It is described in more detail in [2]. T-Tutor engages the learner in conversation in one panel and displays content in another. A chat bot powered by ChatScript [14] gives T-tutor the capability to engage in human-modeled conversation. Student responses are evaluated against target responses using ChatScript's innate functionality and standard semantic analysis techniques like those used by the AutoTutor family of tutors [6; 7].

T-Tutor uses GIFT as its core adaptivity engine[5]. GIFT guides the learner through a topic sequence from the extracted topic tree and guides learner-level pedagogy by adaptively selecting granules based on pedagogical need and learner state. In order to provide a dynamic link between the analyzed content and specific intelligent tutor, TRADEM generates on-demand JSON files that encodes all of the information needed for an intelligent tutor to adaptively and interactively implement a pedagogical plan.

Evaluation: In our Topic Detection, we use AI and data mining techniques to extract topics and sequencing data. This results in an *a priori* model based implied by the source materials. Our goal in evaluation is to use actual learning outcomes to update this model. To this end, TRADEM enables the delivery system to report observed assessment results, with each result mapped to one or more learning outcomes or topics. Once data is gathered, it will be processed to determine the goodness-of-fit between the observed data, the existing topic model, and other potential variants of this model.

3. THE BIGGER PICTURE

The processes we have described brings data mining and practices into the realms of training and education to improve speed, quality, and flexibility of content production. In addition, this approach allows for direct comparison between pedagogical approaches. TRADEM's automated methods produce standardized machine-readable data with testable topic models analyzed based on observed learning outcomes. In other words, we can mine data generated by users and determine how well a given model fits the data. Markov modeling and structural equation modeling can be used to *infer* learning effects if the model or pedagogy is changed, and will immediately update tutors constructed from the models. In other words, the use of EDM to automate ADDIE creates standardized data structures upon which e-learning content is built which, in turn, enables EDM to be used to improve the structures and make the e-learning more effective.

4. ACKNOWLEDGEMENTS

The work reported here was supported in part by Small Business Innovation Research grants and contracts from the National Science Foundation and the Army Research Laboratory.

5. REFERENCES

[1] BRANSON, R.K., RAYNER, G.T., COX, J.L., FURMAN, J.P., and KING, F., 1975. Interservice procedures for instructional systems development. Executive summary and model Florida State Univesrity, Tallahassee.

[2] BROWN, D., MARTIN, E., RAY, F., and ROBSON, R., 2014. Using GIFT as an adaptation engine for a dialogue-based tutor. In *GIFT Symposium 2* Army Research Lab, Carnegie Mellon University (to appear).

[3] CHAPMAN, B., 2010. How long does it take to create learning.

[4] DODDS, P. and FLETCHER, J.D., 2004. Opportunities for New "Smart" Learning Environments Enabled by Next-Generation Web Capabilities. *Journal of Educational Multimedia and Hypermedia 13*, 4, 391-404.

[5] GOLDBERG, B., BRAWNER, K., SOTTILARE, R., TARR, R., BILLINGS, D.R., and MALONE, N., 2012. Use of Evidence-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)* NTSA.

[6] GRAESSER, A., CHIPMAN, P., HAYNES, B., and OLNY, A., 2005. AutoTutor: An Intelligent Tutoring System with Mixed-Initiative Dialogue. In *IEEE Transactions on Education* IEEE, 612-618.

[7] GRAESSER, A., PENUMATSA, P., VENTURA, M., CAI, Z., and HU, X., 2007. Using LSA in AutoTutor: Learning through mixed-initiative dialogue in natural language.

[8] HEILMAN, M. and SMITH, N.A., 2009. Question generation via overgenerating transformations and ranking Carnegie Mellon University, Pittsburgh, PA.

[9] HORNIK, K. and GRÜN, B., 2011. topicmodels: An R package for fitting topic models. *Journal of Statistical Software 40*, 13, 1-30.

[10] LYKINS, L., DIXON, A., MILLER, L., MANDZUK, C., FRANKEL, D., MCDONALD, A., and KELLY, M., 2013. Informal Learning: The Social Revolution. In *White Papers* American Society for Training and Development, Alexandria, VA.

[11] MITKOV, R. and HA, L., 2003. Computer-Aided Generation of Multiple-Choice Tests. In *HLT-NAACL Workshop on Building Educational Applications Using Natural Language Processing*, Edmonton, Canada, 17-22.

[12] ROBSON, R., RAY, F., and CAI, Z., 2013. Transforming Content into Dialogue-based Intelligent Tutors. In *The Interservice/Industry Training, Simulation & Education Conference* National Training and Simulation Association, Orlando, FL.

[13] VANLEHN, K., 2011. The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist 46*, 4, 197-221.

[14] WILCOX, B., 2013. ChatScript SourceForge.

Using Multimodal Learning Analytics to Study Learning Mechanisms in Hands-on Environments

Marcelo Worsley
Stanford University
Graduate School of Education
mworsley@stanford.edu

Paulo Blikstein
Stanford University
Graduate School of Education
paulob@stanford.edu

ABSTRACT

In this paper, we propose multimodal learning analytics as a new approach for studying the intricacies of different learning mechanisms. More specifically, we conduct two analyses of a hands-on, engineering design study (N=20) in which students received different treatments. In the first analysis, we used machine learning to analyze hand-labeled video data. The findings of this analysis suggest that one of the treatments resulted in students initially engaging in more planning, while the other resulted in students initially engaging in more building. In accordance with prior literature, beginning with dedicated planning tends to be associated with improved success and improved learning. In the second analysis we introduce a completely automated multimodal analysis of speech, actions and stress. This automated analysis uses multimodal states to show that students in the two conditions engaged in different amounts of speech and building during the second half of the activity. These findings mirror prior work on teamwork, expertise and engineering education. They also represent two novel approaches for studying complex, non-computer mediated learning environments and provide new ways to understand learning.

Keywords

Learning Sciences, Qualitative, Computational, Constructionism

1. INTRODUCTION

Despite the many years that humans have studied learning and human cognition there are still many unanswered questions in how people learn. This has partially been the result of limitations in the ways that we are able to study learners. More specifically, a large portion of prior research was limited by a tradeoff between the types of learning environments that could be studied, and the scale at which a given phenomenon could be analyzed.

However, as the tools of educational data mining and learning analytics continue to advance, we are beginning to dismantle this tradeoff. We are now able to analyze a far greater variety of learning environments and at unprecedented scales. In this study, in order to keep the analysis verifiable, we do not yet venture to tackle big data as it relates to a large number of participants. Instead, we tackle the big data question as it relates to analyzing extremely high frequency data, from several data streams. We use multimodal learning analytic [1, 2] techniques to study speech, gesture and electro dermal activation among pairs of students as they complete a hands-on engineering design task.

The context for this paper is an extension of our prior work [3], where we present two different approaches that students use in engineering design: example-based reasoning – using personal examples from the real-world as an entry point into solving a task; and principle-based reasoning – using engineering fundamentals as the basis for one's design. These two reasoning strategies complement prior work on learning by analogy [4], expertise [5, 6] and forward-backward reasoning [7]. In [3] we describe example-based reasoning and principle-based reasoning in qualitative terms, and then proceed to use these two approaches in a controlled study (N=20) that compares how each approach impacts learning gains and performance during a collaborative hands-on activity. In that study we found that principle-based reasoning improves the quality of designs ($p < 0.05$) as well as the learning of important engineering principles ($p < 0.002$). The goal of this paper is to expound upon why these differences may have arisen between the two conditions. As such, we employ multimodal learning analytic techniques as a way to systematically study how example- and principle-based reasoning are associated with different multimodal behaviors as observed in the each student's process.

2. METHODS

In this paper, we briefly present results from two complementary analyses of example- and principle-based reasoning. The overall approach closely mirrors our previous work [8, 9] on analyzing design strategies and success in hands-on engineering tasks. Specifically, in the first analysis we manually annotate the students' actions, and segment the data based on when they explicitly evaluates their structure. The proportions of actions in the different segments are used to find representative clusters, which are subsequently used to re-label each user's sequence of segments. Finally, we compare sequences across participants.

In the second analysis we again use clustering to reduce the set of multimodal states from several hundred, down to four. However, it differs from analysis 1 in that all of the data is automatically derived from speech, gesture and skin conductance data. Additionally, instead of segmenting the data when students evaluate their structure, we use fixed 30-second time windows.

3. RESULTS

The results from the first analysis, which combined qualitative coding with X-means clustering, demonstrated that students in the principle-based condition were more likely to start the task by planning (see PREPARE in Figure 1). Planning has been associated with increased success in several domains [10,11,12,13,14]. In contrast the example-based condition was typified by students who immediately began to build their projects and overlooked the importance of thinking about and planning their structure (see IMPLEMENT in Figure 1). Furthermore, the first analysis also found that success correlates with students beginning with planning. Hence the principle-based conditioned was associated with increased planning, which may have facilitated their improved performance.

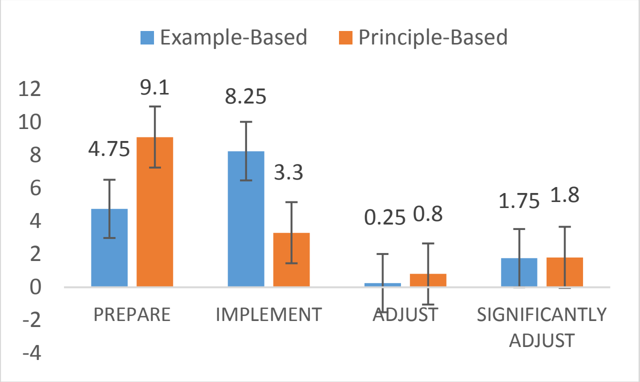


Figure 1 - Scaled Frequency of Cluster Use by Condition - the y-axis is the count of times used, and the x-axis is the different clusters, or states of user actions, as derived from clustering

In the fully-automated multimodal analysis our initial results suggest that students in the example-based condition are much more likely to transition towards an increase in speech during the latter half of the activity. This is in contrast to the principle-based group which shows no significant changes in speech, gesture or stress, over the course of the activity. In our ongoing work we are looking to better understand the nature of the multimodal interactions and what caused the students in the example based condition to engage in significantly more dialogue. We have several initial hypotheses that we will describe in future work. For example, an initial analysis of student speech during the intervention phase of the experiment found significant differences between the two conditions. Namely, students in the principle-based conditions generated more speech during the intervention phase than the example-based condition. This may have helped the students be better prepared for the activity, and allowed them to circumvent the talking observed in the latter half of the experiment for the example-based condition. However, additional analysis is required to determine a link between the speech during these two phases of the experiment.

4. CONCLUSION

Taken in concert, these two analyses provided initial explanations concerning why principle-based reasoning produced higher quality designs and greater learning gains than example-based reasoning. Based on the analysis of hand-labeled process-oriented data, in conjunction with machine learning, we were able to show how students in the principle-based reasoning condition were more likely to begin the task with planning. In contrast, students in the example-based condition were more likely to start by building. These findings aligned with previous observations made in a number of disciplines. In the second analysis, we used a completely automated multimodal algorithm to construct generalizable multimodal states and found that students in the principle-based condition had less variation in their speech, gesture and skin conductance over the course of the activity. This difference was particularly noticeable during the second half of the activity. Both of these seem to point to students being better prepared after participating in the principle-based reasoning intervention. Thus, we have shown that in addition to producing differences in learning and success, the two conditions resulted in different processes. This is important because it provides researchers with a more fine-grained representation of how the two treatments differed. Examining the underlying mechanics of different treatments provides educators and designers with a more complete set of strategies to adopt and utilize in their teaching and designing. To this end, beyond simply saying that the conditions are different, multimodal learning analytics provides us with a tool that explains how they are different, and, in so doing, starts to answer questions around why they differ. That said, there remain a number of important questions and opportunities in studying the mechanics of successful learning interventions. We intend to more closely examine the findings reported in this paper, and investigate additional hypothesis that would explain the noted differences in student outcomes in our ongoing research.

5. REFERENCES

[1] Worsley, M. 2012. Multimodal Learning Analytics: Enabling the Future of Learning through Multimodal Data Analysis and Interfaces. In Proceedings of the 14th ACM international conference on Multimodal interaction (ICMI '12). ACM, New York, NY, USA. 353-356.

[2] Blikstein, P. 2013. Multimodal Learning Analytics. In Proceedings of the 3rd Annual Learning Analytics and Knowledge Conference. Leuven, Belgium.

[3] Worsley, M. and Blikstein, P. in press. The Impact of Principle-Based Reasoning on Hands-on, Project-Based Learning. In Proceedings of the 2014 International Conference of the Learning Sciences (ICLS).

[4] Gentner, D. and Holyoak, K. J. 1997. Reasoning and Learning by Analogy, 52(1), 32–34

[5] Chi, M. Glaser, Rees. 1981. Expertise in problem solving.

[6] Nokes T J, Schunn C D and Chi M. 2010, Problem Solving and Human Expertise. In: Penelope Peterson, Eva Baker, Barry McGaw, (Editors), International Encyclopedia of Education. volume 5, pp. 265-272. Oxford: Elsevier.

[7] Ahmed, S., & Wallace, K. M. 2003. Understanding the differences between how novice and experienced designers approach design tasks, 14, 1–11. doi:10.1007/s00163-002-0023-z

[8] Worsley, M. and Blikstein, P. 2013. Toward the Development of Multimodal Action Based Assessment. In Proceedings of the Third International Conference on Learning Analytics and Knowledge (LAK '13). ACM, New York, NY, USA, 94-101

[9] Worsley, M. and Blikstein, P. accepted. Analyzing Engineering Design through the Lens of Computation. Journal of Learning Analytics.

[10] Brown, A. L., & Cocking, R. R. 2000. How people learn (pp. 285-348). J. D. Bransford (Ed.). Washington, DC: National Academy Press.

[11] Schwartz, D. L., Bransford, J. D., & Sears, D. 2005. Efficiency and innovation in transfer. Transfer of learning from a modern multidisciplinary perspective, 1-51.

[12] Hatano, G., & Inagaki, K. 1986. Two courses of expertise.

[13] Cross, N., & Cross, A. C. 1998. Expertise in Engineering Design 2. An Outstanding Designer, 141–149.

[14] Atman, C. J., Cardella, M. E., Turns, J., & Adams, R. 2005. Comparing freshman and senior engineering design processes: an in-depth follow-up study. Design studies, 26(4), 325-357.

Using Problem Solving Times and Expert Opinion to Detect Skills

Juraj Nižnan
Masaryk University Brno
niznan@mail.muni.cz

Radek Pelánek
Masaryk University Brno
xpelanek@fi.muni.cz

Jiří Řihák
Masaryk University Brno
thran@mail.muni.cz

ABSTRACT

Construction of a mapping between educational content and skills is an important part of development of adaptive educational systems. This task is difficult, requires a domain expert, and any mistakes in the mapping may hinder the potential of an educational system. In this work we study techniques for improving a problem-skill mapping constructed by a domain expert using student data, particularly problem solving times. We describe and compare different techniques for the task – a multidimensional model of problem solving times and supervised classification techniques. In the evaluation we focus on surveying situations where the combination of expert opinion with student data is most useful.

1. INTRODUCTION

One of important aspects of development of adaptive educational systems is the construction of a mapping between educational content (questions, problems) and latent skills (also denoted as knowledge components or concepts). This mapping is important for student skill estimation, which guides the adaptive behaviour of systems, and is typically constructed by a human and since it is a difficult process, it requires a domain expert. The labeling of items, particularly for large item pools, may be time-consuming, and consequently the process is rather expensive. Another approach is to use automatic construction of the mapping from the data (e.g. Q-matrix method [1, 2]). To be reliable, the automatic approach needs large amount of data. Synergy of these two approaches (e.g. [4]) may bring useful results. We can use a human expert to provide initial labeling of problems and then automatic methods can be used to detect errors that the human might have introduced and to fix them.

Depending on the quality of the provided expert labeling and amount of data, there are three possible scenarios. If the number of expert errors is small or the data are insufficient, it is best to use just the expert opinion (donated as E-zone). If the expert makes lot of mistakes and large data are available, then it is best to use just the data (D-zone). We are interested in the region between these two cases, when it is most advantageous to combine both the expert input and available data (ED-zone). Our aim is to explore techniques for such combination and to map the size of this region.

2. TECHNIQUES

In the following we assume that we have a set of students S, a set of problems P, and data about problem solving times: $t_{s,p}$ is a logarithm of time it took a student $s \in S$ to solve a problem $p \in P$. We have an expert labeling $l_E : P \rightarrow \Sigma$ where Σ is the set of skills. The expert labeling may contain some mistakes when compared to a correct hidden labeling l. The output of our algorithms is some other labeling l_A that may be different from l_E. The goal of our algorithms is to provide a more accurate labeling (according to l) than l_E.

2.1 Model with Multidimensional Skill

In this section we introduce a extension of model described in [5] for predicting how much time it takes a student to solve a particular problem. The model uses a few latent attributes: problem difficulty b_p, student skill β_s, problem skill vector $\boldsymbol{q}_p$ and a student skill vector $\boldsymbol{\theta}_s$. It assumes the following relationship between the attributes: $t_{s,p} = b_p\beta_s + \boldsymbol{q}_p^\intercal\boldsymbol{\theta}_s + \epsilon$. The vector $\boldsymbol{q}_p$ represents the weight of individual skills in the problem p. The vector $\boldsymbol{\theta}_s$ can be interpreted as the values of skills the student s has.

This model is supervised in a sense that it is learning to predict the student solving times. As a byproduct we get the Q-matrix $\boldsymbol{Q}$ which represents the problem-skill mapping that we are interested in. The objective of the model is to minimize the squared prediction error. To get the values of the parameters we use stochastic gradient descent with initial Q-matrix provided by expert labeling. After the algorithm terminates we can check for discrepancies between the expert Q-matrix and the Q-matrix outputted by the parameter estimation algorithm. We will assume that these discrepancies are expert mistakes.

2.2 Supervised Learning

The main idea of using supervised classification methods can be illustrated by the most straightforward approach which uses k-NN (k-nearest neighbors) algorithm and Spearman's correlations $r(p_i, p_j)$ of problems p_i, p_j as a measure of problem similarity. We assume that the most correlated problems belong to the same skill and thus have the same labels. So for problem p_i a new label $l_A(p_i)$ will be the most common label (provided by expert) among the k most correlated problems from P with problem p_i. This approach can find some mistakes, however it brings only small improvement of expert labeling l_E.

Similarly we can use different classification methods with different metric. A problem p_i can be represented as a vector $\boldsymbol{r_{p_i}} = \{r(p_i, p_j)\}_{1 \le j \le |P|}$ and Euclidean distance of these problems can measure similarity of problems (we assume that two similar problems have similar correlations with other problems). As classifier we have chosen logistic regression, which is more sophisticated but still computationally fast.

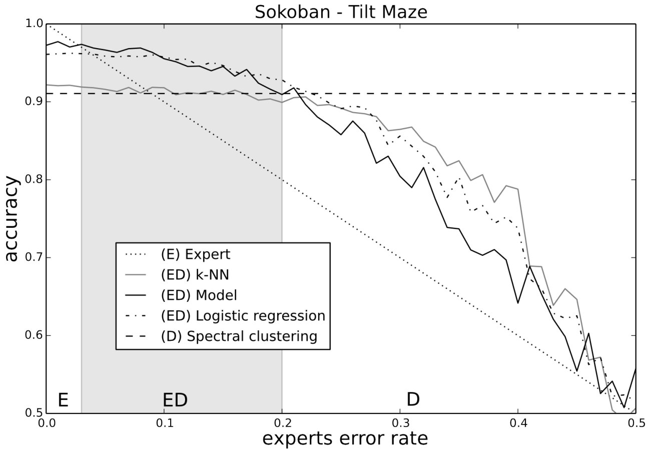


Figure 1: Comparison of techniques for particular situation. The ED-zone is marked for the model.

3. EVALUATION

3.1 Data and Experiment Setup

To evaluate our algorithms we used real data from a Problem Solving Tutor [6]. It is a free web-based tutoring system for practicing problem solving; it is available at **`tutor.fi.muni.cz`**. To simulate multiple skills for the evaluation purposes we mixed data from k problems together. Each problem type represents a single skill (or label). An expert is simulated by taking the correct labeling and introducing some random mistakes with rate $p_e \in [0, 0.5]$. Hence in this situation (as opposed to standard setting), we know the correct "latent" skills and thus we can measure accuracy of a method as the portion of the final labels assigned correctly. The expected accuracy of an expert (E) is $1 - p_e$. Spectral clustering method (see [3]) was used for the evaluation of the D approach. Finally the expert labeling was used in the ED approaches described in section 2.

3.2 Results

Figure 1 shows the comparison of the accuracies of the E, ED and D approaches. We can denote three zones within expert error rate based on which approach (E, ED or D) performs the best. We are interested particularly in the ED-zone, where the newly introduced approaches are the best, specifically in its position and width, which tells us for which values of p_e these approaches are a good choice.

The figure shows that the algorithm based on k-NN brings only small improvement. The other two approaches are significantly better and to each other comparable, however the algorithm based on logistic regression is significantly faster, because it works only with correlation vectors, which substantially reduces the amount of data. On the other hand approach based on model gives more information about problem-skill mapping, because it provides Q-matrix and not only labeling.

Experiments for other problem combinations showed that the size of the zone grows with decreasing performance of D approach and with number of skills. For larger numbers of skills the zone becomes dominant.

4. DISCUSSION

Our experiments address two types of questions: "how" and "when". The "how" question is concerned with the choice of suitable technique for combining expert opinion and student data. Here the results suggest that on one hand the choice of technique is important – note that two similar supervised approaches (k-NN, logistic regression) achieve quite different results. On the other hand, two significantly different approaches (the multidimensional model and logistic regression) achieve very similar results. The "when" question is concerned with mapping when it is useful to use the combination of expert opinion and student data. The results show that this "zone" is sufficiently large to deserve attention and it is useful to combine the expert opinion with student data for large range of quality of expert input.

5. REFERENCES

[1] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, 2005.

[2] T. Barnes. Novel derivation and application of skill matrices: The q-matrix method. *Handbook on Educational Data Mining*, 2010.

[3] P. Boroš, J. Nižnan, R. Pelánek, and J. Řihák. Automatic detection of concepts from problem solving times. In *Proc. of International Conference on Artificial Intelligence in Education (AIED 2013)*, volume 7926 of *LNCS*, pages 595–598. Springer, 2013.

[4] M. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In H. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, volume 7926 of *Lecture Notes in Computer Science*, pages 441–450. Springer Berlin Heidelberg, 2013.

[5] P. Jarušek and R. Pelánek. Analysis of a simple model of problem solving times. In *Proc. of Intelligent Tutoring Systems (ITS)*, volume 7315 of *LNCS*, pages 379–388. Springer, 2012.

[6] P. Jarušek and R. Pelánek. A web-based problem solving tool for introductory computer science. In *Proc. of Innovation and technology in computer science education*, pages 371–371. ACM, 2012.

Young Researcher Papers

Toward Collaboration Sensing: Multimodal Detection of the Chameleon Effect in Collaborative Learning Settings

Bertrand Schneider
Stanford University
schneibe@stanford.edu

ABSTRACT

In this paper, I describe part of my doctoral dissertation in which I have attempted to automatically detect a phenomenon known as the *chameleon effect* in collaborative learning settings. The chameleon effect refers to non-conscious mimicry of other comportments (e.g. postures, mannerisms, facial expressions), such that one's behavior passively and unintentionally changes to match a partner's behaviors. As described below, social mimicry is associated with more productive collaborations and potentially higher learning gains in classroom settings. I describe several studies where I was able to show that visual synchronization (i.e. joint attention), and verbal synchronization (i.e. discourse coherence) were associated with higher learning gains and better collaboration in groups of students, while body synchronization and grammatical mimicry did not predict any of those outcomes. I conclude by discussing implications for educational data mining and describe future work using additional measures such as voice synchronization (e.g. variations in pitch and volume) and arousal synchronization (i.e. variations in heart beat rhythms).

Keywords

Learning Analytics; Collaborative Learning; Mimicry effect.

1. INTRODUCTION

Over the past decades, collaborative learning has been seen as one of the most promising approaches for fostering deep conceptual understanding of complex science concepts. However, even though educational researchers and psychologists have constructed a rich corpus of studies showing the promises of socio-constructivism, much remains to be leaned about effective collaboration among students. As Dillenbourg puts it [3], collaboration in itself is neither good nor bad; there exists conditions that can support productive interactions between students and it's the goal of researchers to discover them. Moreover, he suggests that studies should focus more on process variables rather than learning outcomes: "empirical studies have more recently started to focus less on establishing parameters for effective collaboration and more on trying to understand the role which such variables play in mediating interaction. […] we argue that this shift to a more process-oriented account requires new tools for analyzing and modeling interactions". This is precisely the approach that I am taking in this paper: I use new technologies such as sensors (e.g. eye-trackers, Kinects) combined with data mining algorithms to discover new patterns in collaborative learning settings. More specifically, I used network analysis, natural language processing, supervised and unsupervised machine learning algorithms to make sense of transcripts, eye tracking, and gesture data. My approach is theory driven in the sense that I take advantage of concepts in psychology, ethology and the learning sciences to drive my analyses. One concept that I am closely looking at is the *chameleon effect*.

2. THE CHAMELEON EFFECT

The chameleon effect is defined as "the nonconscious mimicry of the postures, mannerisms, facial expressions and other behaviors of one's interaction partners, such that one's behavior passively and unintentionally changes to match that of others in one's current social environment." [1]. The main hypothesis behind my work is that a high level of mimicry in a small group of students is associated with more productive interactions (not only in terms of learning gains, but also in terms of students' quality of collaboration). I do not postulate a causal link between those two variables, even though I showed in one experiment [10] that it is possible to create interventions supporting collaborative learning groups by increasing synchronization between students. On a more theoretical level, previous literature has shown that the chameleon effect is indeed associated with more satisfying and productive interactions. In the next sections, I will briefly summarize the literature suggesting that enhanced levels of coordination support students' learning for each type of synchronization (visual, verbal and postural). I will then present my results and conclude by mentioning implications for designing learning environments.

2.1 Visual Coordination

The first example of synchronization is *visual coordination*. Historically, there is a plethora of work (summarized in [10]) showing that joint attention plays a crucial role in any kind of social interaction: From babies learning from their caregivers to parents educating their children, students learning from teachers, students collaborating on a project or for any group of adults working toward a common goal, joint attention is a fundamental mechanism for establishing common ground between individuals.

In my experiment, 21 dyads (N=42) remotely worked on a set of contrasting cases; students had to discover how the human brain processes visual information. The experiment had four distinct steps: first, students were welcomed and assigned to two different rooms. They then took a pre-test measuring their existing knowledge on the topic taught (step 1). In the second step, they collaborated via a microphone when working on the contrasting cases.

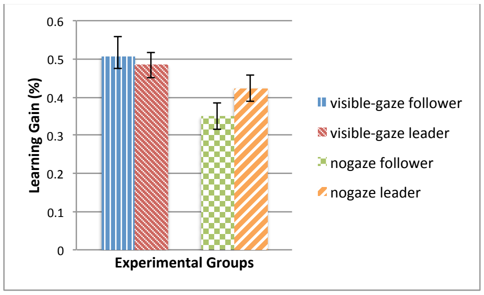


Figure 1: Results of the experiment conducted in [10].

In one condition, members of the dyads saw the gaze of their partner on the screen; in a control group, they did not have access to this information. They spent 15 minutes trying to predict how different lesions would affect the visual field of a human brain. In the third step, they then read a text for another 15 minutes on the same topic describing how the visual pathways of the brain work. Finally, they individually took a learning test to assess their understanding of the topic (step 4).

Results indicate that this intervention helped students achieve a higher quality of collaboration, as measured by [10] (F(1,10) = 24.68, $p < 0.001$) and a higher learning gain (F(1,40) = 7.81, $p < 0.01$). Additionally there was an interaction effect between two factors (experimental conditions and a follower or a leader) on the total learning score: F(1,38) = 5.29, $p < 0.05$. Followers learnt significantly more when they could see the gaze of the leader on the screen. They learnt less when they could not (for more detail, see [10]). Interestingly, participants in the "visible-gaze" condition achieved joint attention more often than the participants in the "no-gaze" condition: F(1,30) = 22.45, $p < 0.001$. More importantly for the context of this paper, *the percentage of joint attention was one of the only measures correlated with a positive learning gain*: r = 0.39, $p < 0.05$. That is, visual coordination was our best measure for predicting students' learning.

I then used network analysis techniques to further exploit this dataset. To construct graphs from gaze data, I divided the screen students had to study into 44 different areas (for more details, see [7]). In this approach, the node size in the dyad graphs is proportional to the number of times dyad members looked at the respective screen area at the same time. Edges are created between nodes when we observe saccades between the corresponding screen regions. The weight of an edge is proportional to the number of saccades between the corresponding screen end-points. Small graphs with few nodes are characteristic of poor collaboration (Fig. 2, left side), and large graphs with highly connected nodes show productive dyads (Fig. 2, right side).

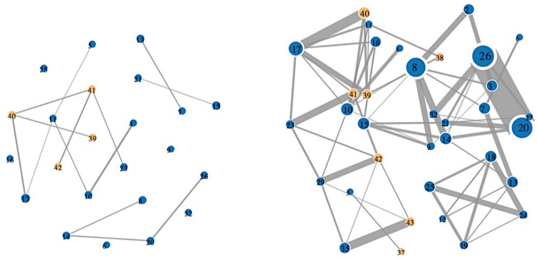


Figure 2: Graphs based on dyads' data. The size of each node reflects the number of moments of joint attention members of the group shared on one area of the screen.

Based on this new dataset, we computed various network metrics. I found that in the visible-gaze condition, there were significantly more nodes (F(1,30) = 8.57, p = 0.06), with bigger average size (F(1,30) = 22.15, $p < 0.001$), more edges (F(1,30) = 5.63, p = 0.024), and more reciprocated edges (F(1,30) = 7.31, p = 0.011). Those results indicate that *we can potentially separate our two experimental conditions solely based on network characteristics*. In [7], I also show that various network metrics correlate with different aspects of a good collaboration (e.g. the number of nodes (and edges) in the graph were associated with a better ability to reach consensus; betweenness centrality was correlated with the ability of students to sustain mutual understanding; and so on).

In summary, this first study shows that visual coordination is indicative of productive interactions in small collaborative learning groups. I will now turn to the second example, verbal coordination among students.

2.2 Verbal Coordination

Danescu [2] mentions how verbal coordination has been shown to enhance communication in organizational contexts, psychotherapy, care of the mentally disabled, and police-community interactions. Thus, there is some evidence showing that verbal mimicry leads to productive interactions. Moreover, I can further divide this concept in two different categories: what Danescu calls *convergence* (i.e. superficial coordination, such as grammatical resemblance) and what other researchers call *coherence* [3] (i.e. deep coordination, such as repeating ideas being expressed by a partner).

Concretely, Danescu used 9 categories from the LIWC corpus (Linguistic Inquiry and Word Counts - http://www.liwc.net/) to compute converge measures. Those categories are: articles, auxiliary verbs, conjunctions, high-frequency adverbs, impersonal pronouns, negations, personal pronouns, prepositions, and quantifiers. The way convergence is computed is relatively trivial:

$$P(b^t_{\hookrightarrow a} = 1 | a^t = 1) - P(b^t_{\hookrightarrow a} = 1).$$

The first expression is the conditional probability of seeing word type *t* expressed by person *b* in answer to person *a*, given that *a* used this word type in the previous utterance. The second expression is just the probability of seeing a particular word type in the entire corpus. Subtracting the second expression from the first one gives us a measure of *convergence*.

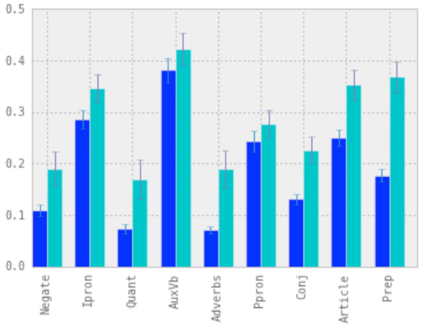


Figure 3: A replication of Danescu's results on my dataset. Errors bars show standard errors. Non-overlapping error bars show statistically significant differences. Light blue bars show the conditional probability of using a particular word type, given that an interlocutor used it in the previous utterance. Dark blue bars show the probability of using a particular word type in the entire corpus.

Reusing the dataset from [10], I was able to show that students were indeed mimicking their partners' grammatical structure (Fig. 3). However this measure was not correlated with students' learning gains or quality of collaboration. I then computed the *coherence* of students' discussion (for more details, see [11]): by segmenting the transcripts and computing document similarity measures between those sequential segments (i.e. tf-idf, followed cosine similarity measures), I was able to compute the extent to which students were reusing ideas cited earlier in their discussion. I found that students in the "visible-gaze" condition were significantly more coherent than students in the "no-gaze" condition and that this measure was positively correlated with students' learning gain: r(19) = 0.540, p = 0.011.

In summary, those results suggest that not any kind of synchronization is indicative of productive patterns of collaboration. I found that *coherence* was associated with higher learning gains, but *convergence* was not.

2.3 Postural Coordination

In previous research, I was able to show that joint attention was beneficial to establishing a common ground, which in turn positively influenced how much students learned during an activity [10]. Other lines of research (in ethology as well as in human psychology [1]) suggest that body synchronization is also associated with more productive collaborations. I was inspired by those results and decided to compute a metric for gestures synchronization using the Kinect data. The dataset comes from a study conducted with a tangible interface, where students had to reconstruct the human hearing system [8].

My approach was to first take pairs of data points (one from each student) and computes the distance between them. Distance was calculated by taking the absolute value of the difference between the joint angles of each participant. Those differences were then averaged for each time point. I created graphs with time series of those data points as well as an overall measure of body synchronization. Statistical analyses did not reveal any significant correlation between body synchronization and learning gains: r(16) = 0.189, p = 0.453. I thus conducted a second attempt that was inspired from the literature in eye-tracking studies: it usually takes +/- 2 seconds for participants in a collaborative situation to adjust their gaze to their partner's behavior. It is possible that body language obeys the same rules. Thus, I repeated the procedure above, but this time, for each data point we looked at the minimum distance in their partner body posture +/- 2 seconds. The correlation with students' learning gains did not reach significance: r(16) = 0.184, p = 0.466. It suggests that even though gaze synchronization is a strong predictor for students' quality of collaboration and learning, body synchronization does not hold the same properties, at least in the context of this experiment. Successful students were *not* more likely to coordinate their action based on their partner's behavior.

3. DISCUSSION

The work described above shows a first step in computing multimodal metrics of the *chameleon effect*. I showed how visual coordination and verbal coordination were associated with higher learning gains. I also found that grammatical coordination and body synchronization was not significantly correlated with students' quality of collaboration or learning gains. This means that the chameleon effect is not universal: at least in educational settings, it varies in its form and intensity according to different modalities.

Future work should focus on alternative measures of the chameleon effect (e.g. voice features, heart beat rhythms) and assess whether other kinds of synchronization are associated with positive learning outcomes. Future work should also explore the approach described in [7] to a greater extent: building network or probabilistic models on top of large datasets is likely to lead to additional insights in terms of students learning processes. Implications of this work are manifold. We can imagine feeding those features into a machine learning algorithms to predict students' quality of collaboration; this prediction can then be used by a teacher of by a learning environment to propose various scaffolds supporting students' learning. Finally, those metrics can potentially lead to a greater understanding of human social interactions by isolating where and when the chameleon effect actually applies. This understanding can lead to the development of new feedback loops, such as the one described in [10] (i.e. the gaze-awareness tool used by students).

4. REFERENCES

[1] Chartrand, T.L. and Bargh, J.A. The chameleon effect: The perception–behavior link and social interaction. *Journal of Personality and Social Psychology 76*, 6 (1999), 893–910.

[2] Danescu-Niculescu-Mizil, Cristian, and Lillian Lee. "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs." *arXiv preprint arXiv:1106.3077* (2011).

[3] Dillenbourg, P. (1999). What do you mean by "collaborative learning"? In P. Dillenbourg (Ed.), *Collaborative learning: Cognitive and computational approaches* (pp. 1-16). Amsterdam, NL: Pergamon, Elsevier Science.

[4] Graesser,A.C.,McNamara,D.S.,Louwerse,M.M.,and Cai, Z. "Coh-Metrix: Analysis of text on cohesion and language." *Behavior Research Methods, Instruments, & Computers 36*, 2 (2004), 193–202.

[5] Meier, Anne, Hans Spada, and Nikol Rummel. "A rating scheme for assessing the quality of computer-supported collaboration processes." *International Journal of Computer-Supported Collaborative Learning, 2*, 1 (2007), 63-86.

[6] Pennebaker, J W., M. E. Francis, and R. J. Booth. "Linguistic inquiry and word count: LIWC 2001." *Mahway, NJ: Lawrence Erlbaum Associates,* 2001, 71.

[7] Schneider, B., Abu-El-Haija, S., Reesman, J., & Pea, R. (2013). Toward Collaboration Sensing: Applying Network Analysis Techniques to Collaborative Eye-tracking Data. *ACM International Conference on Learning Analytics,* LAK '13 (pp. 107-111). Leuven, Belgium: ACM.

[8] Schneider, B., & Blikstein, P. (submitted). Computational Techniques for Detecting Productive Interaction Around an Interactive Tabletop. *EDM2014.*

[9] Schneider, B., & Pea, R. (in preparation). Toward Collaboration Sensing. *International Journal of Computer-Supported Collaborative learning.*

[10] Schneider, B., & Pea, R. (2013). Real-time mutual gaze perception enhances collaborative learning and collaboration quality. *International Journal of Computer-Supported Collaborative Learning*, *8*(4), 375-397.

[11] Schneider, B., & Pea, R. (submitted). Exploring the Effect of Mutual Gaze Perception on Students' Verbal Coordination. *EDM2014.*

Doctoral Consortium: The Use of Student Confidence for Prediction & Resolving Individual Student Knowledge Structure

Charles Lang
Harvard Graduate School of Education
226 Longfellow Hall, 13 Appian Way,
Cambridge, MA, 01238
+1 (617) 501-3967
charles_lang@mail.harvard.edu

ABSTRACT

In this paper, I describe the beginnings of some research into the use of student confidence or certainty to predict student behavior and represent the structure of knowledge.

Keywords

Confidence, Bayes, certainty, online assessment, probabilistic multiple choice, partial knowledge, Systems Theory, confidence based assessment.

1. RESEARCH TOPIC

1.1 Background

The broader educational landscape is being altered by the ease with which new assessment formats can be administered through Internet-based applications. The workhorse of educational assessment, the multiple-choice question, can now be expanded and altered in ways that were not feasible even a decade ago. A popular expansion has been to collect information about what students think about their answers along with those answers; whether they think they have performed well or poorly, whether they are guessing, or how certain they are in their answer. The family of formats that utilize this strategy is large, including metacognitive assessment, certainty based assessment, and self-efficacy assessment. One common format change is to simply ask students how certain they are in a given multiple choice answer. This format, named a probabilistic multiple-choice question (PMCQ), has been of interest to educational research for at least 100 years.[1] Presently this format is being incorporated into several online assessment systems including the McGraw-Hill LearnSmart system.

Consensus is mixed as to whether the probabilistic multiple choice question adds value above and beyond the multiple choice format though. Indeed, interpretation of confidence is somewhat disputed. During the mid-1970s the PMCQ format was dismissed as flawed on the basis of experimental psychological research that had demonstrated that human beings suffered from overconfidence bias – the tendency for people to overestimate their own accuracy.[2] Furthermore, a reliable and interpretable scoring method was never agreed upon within the psychometric community despite increases in reliability.[3]

1.2 Topic

There are two aspects of Probabilistic Multiple Choice Questions that I have been pursuing. The first is whether student confidence data produces any improvement in the prediction of student performance when compared to student correct/incorrect data. The second is whether or not student confidence might provide a way of structuring representations of individual student knowledge.

2. PROPOSED CONTRIBUTIONS

2.1 Projection

With respect to the first contribution, I have preliminary data that supports the psychometric theory of [4–6]). The suggestion of which is that whether or not student confidence outperforms correct/incorrect may depend on the level at which the prediction is made.

We performed a test in which students were shown a multiple choice item, but instead of choosing a single, correct answer they reported their confidence in each of the possibilities. They were asked to do this four times for each item, but each time the item was shown two answers were removed.

In this test student confidence appeared to be better at predicting student level performance over time, but worse at predicting class level performance over time. The interpretation according to theory is that student confidence retains information peculiar to each student that is useful for predicting their individual behavior, but creates a very noisy signal when trying to predict the average behavior of the group.

Table 1. Prediction accuracy of student confidence vs. correct/incorrect at student and class level projecting first administration and second item administration.

	Confidence	Correct/Incorrect
Student Level	0.697	0.781
Class Level	0.956	0.853

$$Accuracy_i = \frac{\bar{x}_{correct} - \sqrt{(\bar{x}_{correct} - projection)^2}}{\bar{x}_{correct}}$$

2.2 Structure

If confidence is useful for predicting individual student performance we have some hope that confidence measurements may provide insight into the structure of knowledge for individual students. This makes sense at an intuitive level, if I am an expert in history I will likely be more confident in history than biology and this will be demonstrated in a test that includes both history and biology items.

But simply plotting out confidence levels seems to provide only a gross relationship and does not tell us the relationship between domains or topic or skills. For example, we can use a rudimentary social network analysis to map out items on a test according to a student's confidence in the correct answer. Edges represent the difference in confidence between different items and nodes represent items the image is iteratively resolved so that all nodes are the correct distance from each other but the structure is not necessarily meaningful:

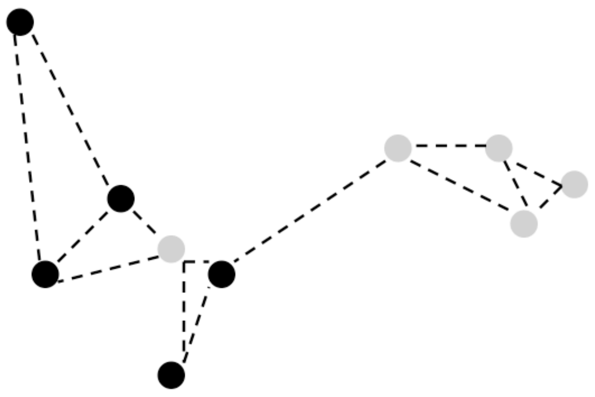


Figure 1. Social Network Analysis of one student's confidence in test items. History items are in black and biology items are in grey.

These structures seem to hint at something, but it isn't clear how to interpret the clustering. In an effort to bring structure to these diagrams I have developed an algorithm based on the Cognitive Bayesian work of Griffiths and Tennenbaum.[7]

2.3 Prediction

The fundamental idea behind applications of Bayes Theorem to people's thinking such as Decision Theory[8] and Cognitive Bayes [7,9] is to change the vantage at which it is applied. For example, instead of conditioning on the situation from the perspective of a researcher or an assessor (e.g. – the probability of the student being correct given the item) we condition on the situation from the perspective of the person being assessed (e.g. – what is her hypothesis, and on what data is she conditioning). For example, if we were studying a student as they answer the following item:

Koalas are:

A. Carnivores
B. Omnivores
C. Herbivores
D. Calmivores

We could devise a model for the way they approach each answer A, B, C & D:

$$P(koalas\ are\ herbivores) = \frac{P(koalas\ are\ herbivores).P(data|koalas\ are\ herbivores)}{P(data)}$$

In this model students weigh the likelihood of the data they have on hand against their prior beliefs, and as more data are presented, they are able to update those beliefs. For example, we might show a student pictures of koalas and every time we revealed a new picture we asked the student whether she thought the koala was a herbivore. We could model the process of the student's opinion as a Bayesian process where each new picture was a datum that changed the likelihood, generated a posterior and then that posterior became the new prior. This formalization is analogous to Snow's separation of internal and external factors: the internal factors are represented by the prior probability and the external factors are represented by the likelihood. The process whereby new data is incorporated into the prior is called Bayesian updating. Essentially, this allows us to directly account for different sources of data in a dynamic fashion, with the final iteration being the best estimate of student knowledge, accounting for external factors. The updating idea underlies features of Decision Theory and Cognitive Bayes, and is used in the classic student knowledge-tracing algorithm BKT. Where Decision Theory and Cognitive Bayes part ways though, is over the efficiency of that updating mechanism.

The Decision Theorist will assume that updating is efficient or *rational* [10] and that there is error in the individual's reporting of her posterior. Decision Theoretic questions tend to be along the lines of "Do financial analysts make rational decisions about market conditions?" The Cognitive Bayesian, however, presumes the individual can state his own posterior probability accurately, but that the incorporation of new information is rarely performed efficiently. Data may not be attended to, nor may they be wholly incorporated into a person's beliefs. A Cognitive Bayesian question tends to be drawn more from experimental psychology, asking questions such as "How do the following conditions impact peoples' prior probability in a specific task?"

The bottom line for the purposes of bringing structure to individual student confidence data is that the Cognitive Bayesian Model splits student confidence in two: the prior (what the student brought to the test inside their head) and the likelihood (the way the student is weighting new data during the test). This rudimentary but important categorization can be mapped onto the work of Snow [11] who conceived of student goal driven behavior as the interface between internal factors (cognitive, conative affective) and external factors (demand, opportunity). Ostensibly Snow's internal factors are represented by the prior probability, the posterior is the student behavior and the likelihood is how the student is mediating external factors.

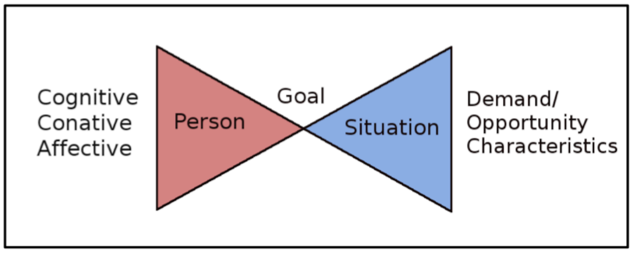


Figure 2. Snow's conception of the interface between internal (person) and external (situation) factors.

To investigate whether this algorithm is worth anything we plan to compare it to BKT and a variant of BKT developed by Wang & Heffernan[12] that has been successfully used in predicting partial knowledge (KTPC). Confidence data is currently being collected through the ASSISTments system.

Rudimentary results have been tested using Wang & Heffernan's partial knowledge data. This data is generated by scoring student performance based on how much assistance they receive (hints, trials, advice). The algorithm did not outperform KTPC in this test though partial knowledge generated in this way may be a poor proxy for confidence data.

3. Advice Sought

There are three areas I would like advice on. The first is that my background is in measurement and psychometrics. I would like to seek advice on how to adapt and change my approach and language to be appropriate for the EDM community. Second, but related, I am looking for advice on how to approach validity, in particular how to approach validity when using time series data. I can interpret the confidence data I will collect in terms of reliability, and compare the predictions of different models through correlation and standard error but I am quite adrift how this relates to a validity framework or whether it needs to?

Thanks in advance.

4. REFERENCES

[1] Williamson, G. F. Individual Differences in Belief, Measured and Expressed by Degrees of Confidence. *J. Philos. Psychol. Sci. Methods* **12,** 127–137 (1915).

[2] Langer, E. J. The illusion of control. *J. Pers. Soc. Psychol.* **32,** 311–328 (1975).

[3] Echternacht, G. The use of confidence testing in objective tests. *Rev. Educ. Res.* **42,** 217–236 (1972).

[4] Borsboom, D. *Measuring the mind : conceptual issues in contemporary psychometrics*. (Cambridge University Press, 2005).

[5] Ellis, J. & van den Wollenberg, A. Local homogeneity in latent trait models. A characterization of the homogeneous monotone irt model. *Psychometrika* **58,** 417–429 (1993).

[6] Molenaar, P. C. M. & Campbell, C. G. The New Person-Specific Paradigm in Psychology. *Curr. Dir. Psychol. Sci.* **18,** 112–117 (2009).

[7] Griffiths, T. L., Kemp, C. & Tenenbaum, J. B. in *Camb. Handb. Comput. Psychol.* (Sun, R.) 59–100 (Cambridge University Press, 2008).

[8] Schlaifer, R. & Raiffa, H. *Applied statistical decision theory*. (Division of Research, Graduate School of Business Administration, Harvard University, 1961).

[9] Tennenbaum, J. in *Adv. Neural Inf. Process. Syst.* (Solla, S. A., Leen, T. K. & Müller, K.-R.) 1098 (MIT Press, 2000).

[10] *The Probabilistic Mind: Prospects for Bayesian Cognitive Science*. (Oxford University Press, 2008).

[11] Shavelson, R. J. *et al.* Richard E. Snow's Remaking of the Concept of Aptitude and Multidimensional Test Validity: Introduction to the Special Issue. **8,** 77– (2002).

[12] Wang, Y. & Heffernan, N. Extending Knowledge Tracing to allow Partial Credit: Using Continuous versus Binary Nodes. (2011). at <http://web.cs.wpi.edu/~nth/pubs_and_grants/papers/2013/AIED2013/YuTaoContinousNodeSub.pdf>

Nonverbal Communication and Teaching Performance

Roghayeh Barmaki
Department of Electrical Engineering and Computer Science
University of Central Florida
barmaki@knights.ucf.edu

ABSTRACT

Nonverbal communication plays a vital role in determining the success or failure of people in their ordinary life and professional careers. In a classroom, successful teacher-student communication has a critical effect on teaching proficiency and student learning. The majority of interpersonal communication is nonverbal including kinesics, proxemics, and paralanguage. This research examines the applications of nonverbal techniques such as hand gestures, body postures and proximity as powerful communication skills exhibited during teaching in a virtual classroom called TeachLivE™. A reflection tool, *TeachLivE After Action Review System* (TeachAARS), is used for data collection from two perspectives: 1) evaluate the effectiveness level of teachers with ratings based on observational data, and 2) annotate the constructive and unconstructive body movements of these teachers in the virtual classroom environment. Teaching effectiveness ratings combined with collected kinesics tags from five participant teachers were analyzed. The analysis indicates that nonverbal cues, especially open hand gestures and proximity, may play an important role in the preparation of an individual for teaching. In future, the data set will be analyzed with machine learning techniques such as regression to design a predictive model of classroom preparation based on nonverbal communication skills. The goal is to use objective metrics as part of teacher preparation, helping prospective and in-service teachers to reflect on and improve their classroom performance.

Keywords

nonverbal communication, virtual reality, after action review, teacher preparation

1. INTRODUCTION

Establishing a good communication between students and the teacher introduces successful steps for both learning and teaching process. Communication is more than words, and it is important for teachers to understand the nonverbal messages they are sending and receiving in the classroom [2, 8]. Nonverbal messages include facial expressions, eye contact or lack of eye contact, proximity and closeness, hand gestures, and body language [8]. Much of the research about nonverbal communication indicates that as little as 7 percent of communication is spoken words and the majority is nonverbal and paralinguistic cues [1]. Hence it is critical for teachers to learn to apply nonverbal communication signals in the classroom.

Apart from the theoretical courses and references that help novice teachers to passively learn about teaching proficiency basics such as communication and management skills, simulation-based training systems provide a safe and comfortable environment for them to interactively practice teaching skills in a realistic classroom. TeachLivE™ is an immersive, mixed-reality virtual environment, designed at University of Central Florida, for teachers to rehearse and hone their classroom skills. In this virtual classroom, teachers interact with student avatars that are controlled in real time by a human-in-the-loop system. Having good communication skills, specifically nonverbal, is critical for teachers in a real classroom and, as such, in the virtual classroom.

This study is intended to discover and understand the correlation of classroom teaching preparedness to nonverbal signals exhibited by teachers while interacting in the virtual classroom-TeachLivE™. The study mainly emphasizes body language and proximity. These types of nonverbal behaviors are reviewed and annotated manually by experts with an after action review tool (TeachAARS) that keeps a record of each teaching session. Additionally, teaching effectiveness is also assessed based on Danielson's [3] teacher evaluation criteria. This approach involves four observers who tag the behavior of five teacher participants from the above two different perspectives. The analysis of results at this point of the study indicates that nonverbal signals are effective indicators of teaching proficiency/preparedness.

2. SIMULATION AND TRAINING

Simulation-based training systems provide learners a low-cost and hazardous-free environment in which they may practice and improve their skills. As a consequence, simulation and modeling are broadly used in a variety of fields and across different applications. As an example of simulation research that is more closely related to the focus of this study, Luciew and colleagues [6] present the details of developing interview procedure for Immersive Learning Simulations (ILS). Concurrent research of body language, facial expression and proxemics relative to the interview process are discussed in the research. Their work is focused on nonverbal expressions of human and avatar subjects that indicate the impact of nonverbal expression studies in simulation. There are many other applications of the use of modeling and simulation in education, that TeachLivE is one of the pioneers.

One of the main capabilities of training systems based on simulation is the provision of assessment and feedback. As a result, the majority of simulation-based training systems are paired with an after action review (AAR) tool that makes it possible for supervisors and reviewers to oversee the trainee's simulation sessions and provide feedback.

The TLE-TeachLivE™ (TLE represents for Teaching Learning Environment) was designed at the University of Central Florida explicitly to help in-service and practicing teachers hone their teaching skills, including those associated with classroom management, pedagogy and content delivery.

In the TeachLivE™ environment, there is typically one student who is in focus and the others who are out of focus. The student in focus is the one currently being addressed by the teacher [4]. That student is inhabited by a human-in-the-loop, called an inter-actor, who controls behaviors and interactions. Students who are out of focus are controlled by agent-based software that can be influenced by the inter-actor who can choose a behavior genre. In general, that selection is influenced by the classroom management skills of the teacher. Teachers walk into a room with a big TV screen, one camera, one wireless microphone and one Kinect sensor that is connected to the client machine. Teachers can see the virtual classroom and five student avatars in the TV and approach to students by entering to their virtual zones. For vocal interactions, there is a Skype connection between client (teacher) and server (the inter-actor station).

Every teacher can provide a lesson plan for her intended teaching session, and also determine the level of behavior escalation (0-5) in order to hone her effective teaching behaviors. Behavior escalation levels are defined for treatments of student avatars that vary from no misbehavior to intense misbehavior in the virtual classroom. These settings help teachers with professional development in areas of targeted need.

In order to facilitate the process of teacher assessment, TeachAARS, or *TeachLivE After Action Review System*, was designed and integrated into the TeachLivE system. TeachAARS does direct video/audio capturing that contains both the virtual classroom and the participant video in a paired window. In addition to directly recording sessions, TeachAARS has the capability to support behavior tagging. Each tag is associated with a sequence of frames, and thus allows selective viewing during reflection or debriefing procedure. Figure 1 displays the TeachAARS environment for teacher assessment. TeachAARS is integral to this study, as it is used to tag the nonverbal messages and body signatures that teachers use in the classroom.

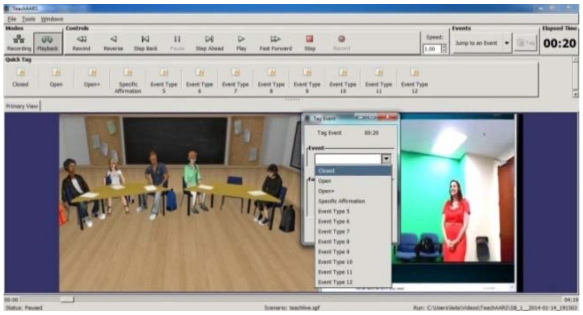


Figure 1. TeachAARS as a review tool. In the primary view, left window shows the virtual classroom, right window shows the teacher participant while interacting with the classroom. An observer annotates tags associated with observed behaviors, e.g., the closed tag if the teacher exhibits a closed posture.

3. STUDY PROCEDURE

Nonverbal communication refers to all of the elements of communication excluding the actual words used [7]. Nonverbal communication strategies are consistently noted in approaches to teacher training. The effects of strategies like eye contact, prolonged gaze, and proximity can have positive or negative effects on student behavior and classroom management, depending on the situation and context [9]. In this research, nonverbal communication skills are indicated as a major factor of teaching preparedness [3]. Two types of nonverbal expressions are investigated in this study: a) proximity b) open vs. closed body posture.

Proximity can be used to encourage student participation and strategically redirect them. Proximity also helps teachers to have better management in the classroom because the students' disruptive behaviors are controlled by approaching them [5]. On the other hand, proximity means attention, affirmation and closeness of the teacher to the speaking student [2]. In TeachLivE™, the simulation has been designed to enable the teacher to move close to the student avatar within the virtual environment. While moving, the visual perspective moves with the teacher, even allowing eye-to-eye communication. Proximity behaviors of teachers are tagged in TeachAARS by observers, to understand how frequently teachers use proximity in their teaching sessions.

Another effective measure for nonverbal cues is open vs. closed posture [10]. Open posture is often used as a measure of closeness, receptivity, and interest. Open postures illustrate positive feelings to others and show that the person is open and positive to the listener, whereas closed postures are often cited to indicate defensiveness, aggression, and avoidance [10]. In general, closed body poses demonstrate negative feelings to the other person. When somebody folds and crosses her arms, she seems to protect herself from the other person and her listener feels that she is not open and comfortable in the communication. Figure 2 represents some frequent standing open and closed body posture models [2] that reviewers use as a reference during the coding of nonverbal expressions in this study.

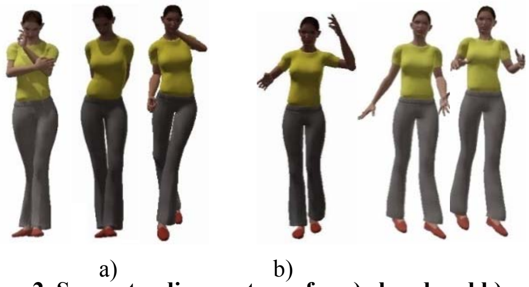


Figure2. Some standing postures for a) closed and b) open body language [6].

More explicitly, reviewers measure the frequency and the timing for teachers withholding open or closed poses.
In this research, it is hypothesized that there is a correlation between positive teaching performance and having good nonverbal signals. The first step in data collection is to review the teaching sessions of teacher participants in the virtual classroom environment, TeachLivE™. As mentioned before, TeachLivE's assessment tool, TeachAARS is used to annotate the nonverbal behaviors (proximity and body posture). As the next step, it is required to evaluate the teaching skills of the participant teachers. Two experts who were blind to nonverbal assessment results, were asked to rate the teaching performance of subjects based on Danielson's [8] teacher evaluation reference. In summary, Danielson defines a framework for a teaching evaluation instrument. Different domains of teaching evaluation are discussed in this framework. Some important domains for teaching evaluation based on Danielson's criteria are: classroom management, communicating with students, student engagement, application of pedagogy and content delivery. The inter-rater reliability for body language coding was 0.72 and 0.78 for teaching performance rating (for each category, two different reviewers observed the videos; four in total).
The collected data from coding nonverbal signals and teaching performance ratings of teachers will be used for designing a computational model for teaching practice in TeachLivE™.

4. EXPERIMENTAL RESULTS

This study is related to a national research project funded by the Bill & Melinda Gates Foundation. The research focuses on practicing biology high school teachers. They are asked to interact with the virtual classroom to teach a sample scenario (Technology applications in biology) in a nine-minute session once a month for nine consecutive months. All of the sessions of participants are recorded with TeachAARS for later evaluation.

In this paper, ten video sessions of five biology teachers are evaluated from nonverbal and teaching performance aspects. Table 1 represents a summary of collected data for these five participants.

Table 1. Mean, Standard Deviation and Range for nonverbal variables and teaching performance ratings

Variable	Mean	(SD)	Range
# open posture	15.6	9.17	2 - 29
# closed posture	10.2	3.56	7 - 14
# proximity	15	6.48	5 - 20
total # tags	40.8	15.25	14 - 50
% time open posture	43 %	35%	2% -78 %
% max time non-interrupted open posture	14.7%	14.9%	0.54% -33%
% max time non-interrupted closed posture	29 %	28.8%	9.5% -75.9%
teaching performance rating	7	1.41	5 - 9

Table 2 shows the correlations between nonverbal indicators and teaching performance rating in a correlation matrix. The last row of the table highlights the strong positive correlation of proximity and open body posture; and negative correlation of closed body posture with teaching performance. Apart from the maximum of non-interrupted open time in percentage (% max-n open) that has a small negative correlation with teaching rate, all other nonverbal variables have the expected correlation. The strong negative correlation between the maximum of non-interrupted time in closed body posture and teaching performance is considerable.

5. DISCUSSION

A successful teacher-student communication in the classroom indicates teaching proficiency and student learning. In this study two categories of nonverbal communication (proximity and body postures) are focused to discover and understand the correlation between nonverbal codes and teaching efficiency. According to the study, there is a positive correlation between proximity, open body posture and total open posture time with teaching performance rating. There exists a negative correlation between the maximum of non-interrupted closed posture and closed body posture with performance rating, too. This research is going to move forward in two main directions. The first direction will be building robust prediction models for teaching effectiveness with advanced machine learning techniques. The models can also improve with broader range of subjects, which is the goal for future work. The next direction will be collecting automated tags using the Microsoft Kinect SDK in real-time, and assessing the effectiveness of a teacher's body movement using predictive models to give them real-time feedback.

6. ACKNOWLEDGMENTS

I would like to thank Dr. Carrie Straub, Dr. Lisa Dieker and Dr. Michael Haynes as the project designers of the TeachLivE™ that supported me during the data collection and review processes. I also would like to acknowledge the efforts of Aleshia T. Hayes for her comments, knowledge sharing and proofreading. I wholeheartedly would like to acknowledge the supports of my adviser, Dr. Charles Hughes who is also one of project designers of TeachLivE™. Finally, I wish to acknowledge support to the overall TeachLivE™ project from the Bill & Melinda Gates Foundation and the National Science Foundation (CNS1051067). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the sponsors.

7. REFERENCES

[1] Borg, J. *Body language: 7 Easy lessons to master the silent language.* Pearson Education, 2008.

[2] Caswell, C. and Neill, S. *Body language for competent teachers.* Routledge, 1993.

[3] Danielson, C. *The Framework for Teaching: Evaluation Instrument.* Danielson Group. 2011.

[4] Dieker, L. A., Rodriguez, J. A., Lignugaris, B., Hynes, M. C. and Hughes, C. E. "The Potential of Simulated Environments in Teacher Education: Current and Future Possibilities," *Teacher Education and Special Education: The Journal of the Teacher Education Division of the Council for Exceptional Children*, vol.37, no.1, pp.21-33, 2014.

[5] Gunter, P. L. "On the MOVE: Using Teacher/Student Proximity to Improve Students' Behavior," *Teaching Exceptional Children*, vol.28, no.1, pp.12-14, 1995.

[6] Luciew, D., Mulkern, J. and Punako, R. Finding the Truth: Interview and Interrogation Training Simulations. *In Proceedings of the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*. National Training Systems Association, vol.2011, no.1, 2011.

[7] Mehrabian, A. "Communication without words," *Psychological Today*, vol.2, pp. 53-55, 1968.

[8] Miller, P. W. "Body language in the classroom," *Techniques: Connecting Education and Careers*, vol.8, pp.28-30, 2005.

[9] Smith, C. and Laslett, R. *Effective classroom management: A teacher's guide*. Routledge, 1993.

[10] Tucker, J. S. and Anders, S. L. "Adult attachment style and nonverbal closeness in dating couples," *Journal of Nonverbal Behavior*, vol.22, no.2, pp.109-124, 1998.

Table 2. Correlations of variables on collected data

	open	closed	Proximity	% open	% max n-open	% max n-closed
closed	-0.91581641					
proximity	-0.630252814	0.265399837				
% open	0.979459169	-0.920008359	-0.572835101			
% max n-open	0.930559511	-0.81364051	-0.661074991	0.944425568		
% max n-closed	-0.84059492	0.937655126	0.20533967	-0.809136602	-0.620323964	
teaching performance rating	0.139483102	-0.328765936	0.30072311	0.006129132	-0.148887534	-0.553393224

Data-Driven Feedback Beyond Next-Step Hints

Michael Eagle
North Carolina State University
890 Oval Dr, Campus Box 8206
Raleigh, NC 27695-8206
Maiku.Eagle@gmail.com

Tiffany Barnes
North Carolina State University
890 Oval Dr, Campus Box 8206
Raleigh, NC 27695-8206
Tiffany.Barnes@gmail.com

ABSTRACT

Intelligent tutors have been shown to be as effective as human tutors in supporting learning in many domains. Although they can be very effective, the construction of intelligent tutors can be costly. One way to address this problem is to use previously collected data to generate domain models to provide intelligent feedback to otherwise non-personalized tutors. These data-driven methods for providing next-step hints have been successful in providing feedback to students in procedural problem solving tutors. We seek to expand on next-step hints with other data-driven methods. We outline three different interventions, all of which can be generated using previously collected student data.

1. INTRODUCTION

Intelligent tutors have been shown to be as effective as human tutors in supporting learning in many domains, in part because of their individualized, immediate feedback, enabled by expert systems that diagnose student's knowledge states [13]. For example, students provided with intelligent feedback in the LISP tutor spent 30% less time and performed 43% better on post-tests when compared to other methods of teaching [1]. Similarly, Eagle, and Barnes showed that students with access to hints in the Deep Thought logic tutor spent 38% less time per problem and completed 19% more problems than the control group [4]. In another study on the same data, Stamper, Eagle, and Barnes showed that students without hints were 3.6 times more likely to drop out and discontinue using the tutor [12].

Procedural problem solving is an important skill in STEM (science, technology, engineering, and math) fields. Open-ended procedural problem solving, where steps are well-defined, but can be combined in many ways, can encourage higher-level learning [2]. However, understanding learning in open-ended problems, particularly when students choose whether or not to perform them, can be challenging. The Deep Thought tutor allows students to use logic rules in different ways and in different orders to solve 13 logic proof problems for homework.

Although they can be very effective, the construction of intelligent tutors can be costly, requiring content experts and pedagogical experts to work with tutor developers to identify the skills students are applying and the associated feedback to deliver [9]. One way to reduce the costs of building tutoring systems is to build data-driven approaches to generate feedback during tutor problem-solving. Barnes and Stamper built the Hint Factory to use student problem-solving data for automatic hint generation in a propositional logic tutor [10]. Fossati at el. implemented Hint Factory in the iList tutor to teach students about linked lists[7]. Evaluation of the automatically generated hints from Hint Factory showed an increase in student performance and retention [12].

Hint Factory creates hints by modeling previously collected student data into a Markov Decision Process and generating a next step policy, when students request a hint they are directed to the best next step. For this work, we are interested in looking into ways to expand the feedback offered to students beyond these next-step hints. We have outlined three different interventions, all of which can be generated using previously collected student data.

2. THE DEEP THOUGHT LOGIC TUTOR

In Deep Thought propositional logic tutor problems, students apply logic rules to prove a given conclusion using a given set of premises. Deep Thought allows students to work both forward and backwards to solve logic problems [3]. Working backwards allows a student to propose ways the conclusion could be reached. For example, given the conclusion B, the student could propose that B was derived using Modus Ponens (MP) on two new, unjustified (i.e. not yet proven) propositions: $A \rightarrow B, A$. This is like a conditional proof in that, if the student can justify $A \rightarrow B$ and A, then the proof is solved. At any time, the student can work backwards from any unjustified components (marked with a ?), or forwards from any derived statements or the premises. Figure 1 contains an example of working forwards and backwards with in Deep Thought.

3. DATA-DRIVEN FEEDBACK

In this section we will outline three different data-driven methods that we can use to provide hints to students. These methods are all intended to be used in conjunction with the next-step hints that have already been shown as successful.

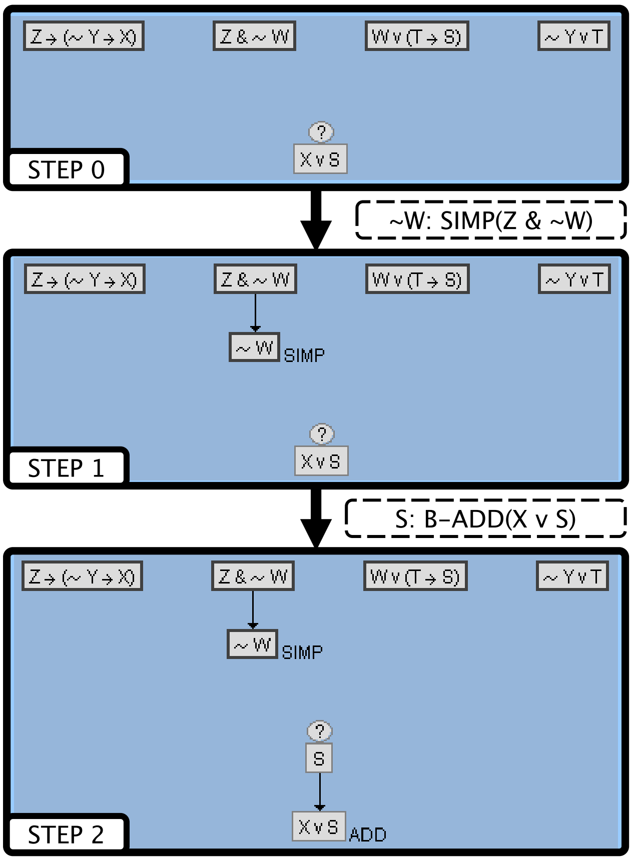


Figure 1: This example shows two steps within the Deep Thought tutor. First, the student has selected $Z \wedge \neg W$ and performed Simplification (SIMP) to derive $\neg W$. Second, the student selects $X \vee S$ and performs backward Addition to derive S.

3.1 High Level Hints

Interaction Networks describe sequences of student-tutor interactions [5]. Interaction networks form the basis of the data-driven domain model for automatic step-based hint generation by the Hint Factory. Eagle et al. applyied Girvan-Newman clustering to interaction networks to determine whether the resulting clusters might be useful for more high-level hint generation [5]. Stamper et al. demonstrated the differences in problem solving between two groups by coloring the edges between Girvan-Newman clusters of interaction networks based on the frequencies between two groups, revealing a qualitative difference in attempt paths [12]. Eagle and Barnes expanded this work into Approach Maps [6], which summarize interaction networks into the higher-level approaches used by students to solve the proofs.

In order to encouage student planning we can use the higher-level approaches discovered with Approach Maps to provide sub-goals to the students. In figure 2 we show a mock up of how the sub-goal could be presented to the student. We hypothesize that these hints will help students learn which parts of the proof to focus on in order to complete problems.

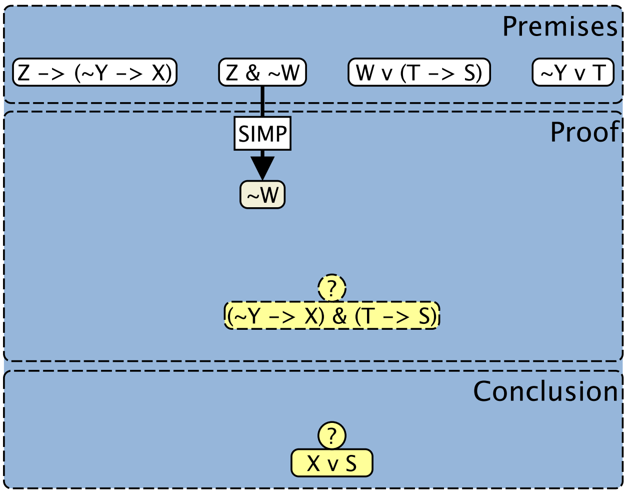


Figure 2: Example of a high level hint. DT offers the student a sub-goal based on commonly derived steps from previously collected data.

3.2 Hazard Hints

Stamper et al. in [12] and Eagle et al. in [5] found evidence that students would sometimes spend a lot of time in approaches that were unlikely to result in a solution. This discovery is important as interventions can be added to warn away from regions that do not lead to goals. For example, we could offer a message that warns them that most students who attempt the same type of proof are not successful. Fossati et al. showed that human tutors helping students with the iList tutor, suggest that students delete unproductive steps [7]. In figure 3, we show an example interface for a *hazard hint*. These types of hints would be offered whenever a student was performing a task that was unlikely to result in a successful proof, with the goal of reducing the amount of "wasted" time.

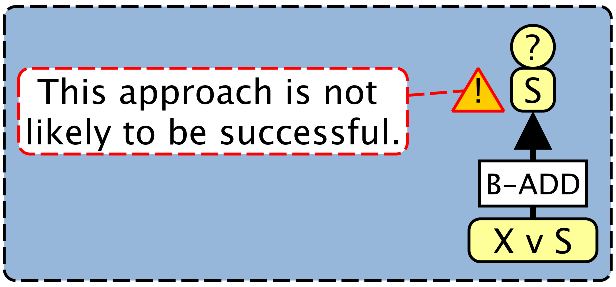


Figure 3: We can warn a student when they approach the problem in a way that is not productive.

3.3 Time Hints

Eagle, and Barnes used survival analysis to model student time-in-tutor and student dropout[4]. Survival analysis is a series of statistical techniques that deal with the modeling of time to event data [8]. It derived its name from its start within medical literature. Survival analysis is also known as reliability analysis or duration analysis.

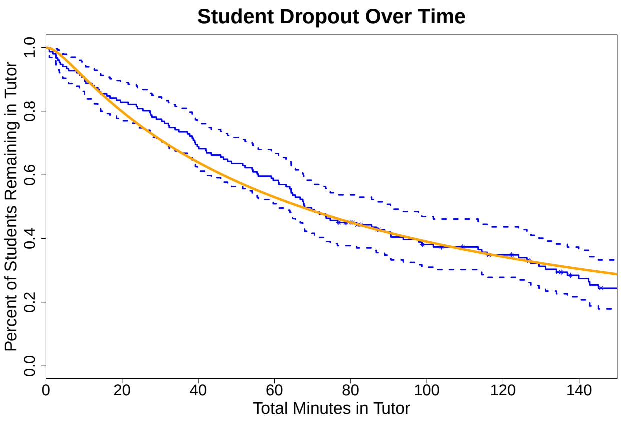


Figure 4: The Kaplan-Meier survival estimation and corresponding 95% confidence intervals show the percent of students remaining in tutor over time. The lighter (orange) line is the AFT model produced from the same data.

We start by first plotting the Kaplan-Meier survival estimator, see figure 4, which is represented as a series of declining steps which is intended to approach the true survival function. We perform our experiments on the Spring and Fall 2009 Deep Thought logic tutor dataset as analyzed by Stamper, Eagle, and Barnes in 2011[11]. We look specifically at 151 students who stopped using the tutor before completing all of the questions required for the homework assignment. Application of the AFT model provides us with coefficients of the model had the intercept (mean) as 4.20 and the SD (scale) as 1.44. The median of the survival function, the location where 50% of people have dropped out of the tutor, is found by $e^{\mu} = e^{4.20} = 66.89$, meaning that half of the students had dropped out after about an hour of tutor interactions. We have plotted the resulting survival curve in figure 4.

We hypothesize that we can prevent dropout by providing feedback when students reach certain thresholds of time within the tutor. To test this we will build survival models based on past student data, using these models we will provide feedback in the form of a pop-up window that will encourage the student, as well as provide them with resources if they are struggling. We can augment these models with information about the students current tutor performance, to get an idea of how likely the student is to complete the tutor. Overall, the use of survival modeling will provide us with more accurate representations of student time-in-tutor, and we can use this information to create interventions that will reduce the number of students who quit the tutor without finishing. In figure 5 we show an example of the type of prompt we can offer a student if our time model shows that the student is in danger of quitting the tutor.

3.4 Evaluation

Data-driven methods for offering next-step hints have been successful. We have outlined three new ways to offer feedback based on previously collected data that can be added in addition to next-step hints. In order to test the effectiveness of these forms of feedback we will seek to repeat studies like Stamper, Eagle, and Barnes' 2011 Hint Factory study [11].

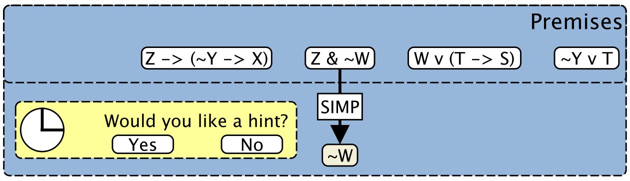


Figure 5: We can remind the student about hints if student is taking longer than predicted.

4. REFERENCES

[1] J. R. Anderson and B. J. Reiser. The lisp tutor. *Byte*, 10(4):159–175, 1985.

[2] B. S. Bloom. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Taxonomy of educational objectives: the classification of educational goals. Longman Group, New York, 1956.

[3] M. J. Croy. Problem solving, working backwards, and graphic proof representation. *Teaching Philosophy*, 23:169–188, 2000.

[4] M. Eagle and T. Barnes. Survival analysis on duration data in intelligent tutors. In *Intelligent Tutoring Systems*, Honolulu, Hawaii, 2014.

[5] M. Eagle, M. Johnson, and T. Barnes. Interaction Networks: Generating High Level Hints Based on Network Community Clustering. In *Educational Data Mining (EDM2012)*.

[6] M. Eagle and B. Tiffany. Exploring differences in problem solving with data-driven approach maps. *Educational Data Mining (EDM 2014)*, 2014.

[7] D. Fossati, B. Di Eugenio, S. Ohlsson, C. Brown, L. Chen, and D. Cosejo. I learn from you, you learn from me: How to make ilist learn from students, 2009.

[8] D. W. Hosmer, S. Lemeshow, and S. May. *Applied Survival Analysis: Regression Modeling of Time to Event Data*. Wiley-Interscience, New York, NY, USA, 2nd edition, 2008.

[9] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.

[10] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. *Educational Data Mining (EDM 2008)*, pages 197–201, 2008.

[11] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. In *Artificial Intelligence in Education*, AIED'11, pages 345–352, Berlin, Heidelberg, 2011. Springer-Verlag.

[12] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education (IJAIED)*, 22(1):3–18, 2012.

[13] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

E³: Emotions, Engagement, and Educational Games

Ani Aghababyan
Instructional Technology and Learning Sciences
Utah State University
anie.aghababyan@gmail.com
2830 Old Main Hill
Logan, UT 84322-2830

ABSTRACT

This study is an investigation of ways to collect student engagement and gameplay data from a digital educational game called Quantum Spectre in order to understand student engagement in such digital environments, and the effect of certain affective states on student gameplay behavior. Proposed study participants are elementary school students, 5th graders, who will play the game over the course of multiple class sessions. Previous research findings suggest that there is an interesting inter-relation between frustration and confusion that requires more attention; the indices of frustration and confusion are influenced by the amount of external support provided. Based on these initial findings, the proposed dissertation experiment will concentrate on significant patterns of frustration and confusion along with their effect on student gameplay and further engagement with the environment.

STATEMENT OF THE PROBLEM AND STUDY SIGNIFICANCE

Digital educational games have become a popular means of instruction in recent years (Mayo, 2009; O'Neil, Wainess, & Baker, 2005; Rodrigo et al., 2008). Many educational concepts (e.g., science, technology, engineering, and mathematics education concepts) are taught and practiced through digital educational game environments. This instructional approach is mainly justified with the realization that games can naturally motivate students to engage with the environments and learn (Prensky, 2001; Kapp, 2012). From the researcher point of view, games provide students with a safe space for failure and confidence to persist. From the perspective of educational establishments, online games offer a unique advantage of simultaneous accessibility for thousands of children along with a customizable learning pace and ability to follow students' learning trajectories. Overall, scientists consider games as a potentially powerful tool for learning (FAS, 2006).

Lack of student engagement can be a threat to the learning environment; disengaged students may not take full advantage of the learning opportunities offered through these settings. Academic affect is one factor that can either benefit or undermine students' engagement and learning. Previous research shows there is a complex interaction between affect and learning (Baker, D'Mello, Rodrigo, & Graesser, 2010). Moreover, affective states trigger different results in different human-computer interaction environments (Rodrigo & Baker, 2011) and depending on the order of affective states, the impact may be negative or positive.

Many researchers acknowledge the importance of understanding students' affective responses to success and failure. Two relevant areas of research on this topic are Angela Duckworth's work on grit (2007) and Carol Dweck's work on self-efficacy (1985; 1991). According to Dweck (2002), a learner's goal orientation (i.e., beliefs about one's abilities and the effectiveness of their effort) may influence their affective response to the success and failure they experience within an environment. As Duckworth (2007) identifies it, grit or persistence is about "sticking with things over the very long term until you master them," which includes overcoming negative experiences of frustration, confusion, and failure. Persistence is currently being researched as it is considered to be a key factor in college completion or completion of similar academic long-term goals (Duckworth, 2007; Duckworth & Seligman, 2005). Well-designed digital educational game environments should be able to provide support for high levels of frustration that could be detrimental for student engagement while developing persistence as students meet new challenges within the game.

There have been many studies concerned with student affective states and their impact on student engagement and motivation in intelligent tutoring systems (ITS) (Rodrigo et al., 2012). However, this promising work has not yet been fully extended to digital educational game environments (O'Rourke, Haimovitz, Ballwebber, Dweck & Popovic, 2014). Therefore, in the current study I address this gap by looking into student behavior, affect and the effect of emotions on student learning within digital educational game environments. The use of digital educational games is becoming widespread, however, its technological design is not on the same level with intelligent tutoring systems where the environment promotes learning through adaptive guidance.

My research is most relevant to the areas of research in affective computing, learning through digital games for learning, and game development. This will inspire game developers to design games that will be more responsive to negative displays of affect to keep students engaged in their environments. In fact, if we are able to detect negative manifestations of certain affective states, game developers will be able to incorporate this detection feature into future digital education game designs and incorporate recommender systems into educational game environments.

Hence, it is worthwhile to continue research that informs game design to include sensorless detection (i.e., not based on data collected from external sensor devices or other extremely obtrusive methods of data collection such as heart rate monitors, eye trackers or skin conductance) of affect, which will provide students with only necessary hints to persist and will not interrupt their beneficial exploration stage. This affect detection system might ameliorate students' negative perceptions of their own abilities in the fields of science and mathematics by guiding them through their confusions and frustrations associated with the learning environment.

However, in order to support learning and increase academic goal orientation while students are engaged in digital game environments for learning, we need to understand student motivation and the emotions that affect them. Students' affective states play a critical role in their performance. Potentially negative emotional states such as confusion and frustration are more crucial to investigate, since emotional variability can be one of the moderating factors of success and failure in the struggle to overcome barriers in goal attainment. Moreover, frustration and confusion are two affective states that are suggested to lead to boredom state. As literature suggest (Baker, D'Mello, Rodrigo, & Graesser, 2010), it is better to be frustrated than bored since boredom leads to disengagement and makes it much harder to bring students back to engagement and concentration from boredom emotional state. Therefore, I believe that if developed well, digital educational games with adaptive support systems may become one of the key ways to assisting students to push past confusion and frustration and develop persistence regardless of their goal orientation. The better and more precise our research findings, the more sophisticated and helpful our educational games will become.

While there have been studies looking into the effects of frustration or/and confusion on student learning and possibility of reducing frustration (Baker et al., 2010; Hone, 2006; Klein, Moon, & Picard, 2002; McQuiggan, Lee, & Lester, 2007), almost none of these studies have looked into whether there is an interaction and order to the pattern of frustration and confusion along with their influence on student engagement with the environment (e.g., concentration or boredom patterns). Kort, Reilly, and Picard (2001) attempted a model of confusion to frustration transitions but their empirical evidence did not support their hypothesized model. In addition, Perkins and Hill (1985) have hypothesized that frustration leads to boredom but their analysis did not allow for such a conclusion since they illustrated association instead of temporal or sequential connection. Yet another study investigated the decay rate of certain cognitive-affective states, however, it did not concentrate on patterns of occurrence but rather on the temporal and tripartite classification of affect (D'Mello & Graesser, 2011). This study by D'Mello and Graesser (2011) has informed the design of this proposed study. Thus, while there have been many attempts at investigating frustration and confusion sequential patterns, there seems not to be empirical evidence on this subject either due to inappropriate analysis method or inconclusive results. Thus, my work will contribute to current research on frustration and confusion by using a sequential pattern mining algorithm on categorical affect sequences in order to identifying sequential patterns and possible interdependency that need to be avoided in order to keep students engaged in digital game environments for learning and make sure students have uninterrupted opportunity for learning.

RESEARCH QUESTIONS, METHODOLOGY AND SOLUTIONS

I believe that to help students learn better and be invested in their own education, we need to understand the motivation and the emotions that affect them while going through learning processes. My dissertation will be focusing on certain emotions, frustration and confusion, manifested while playing an educational game (science learning game). In addition to investigating both frustration and confusion in EG environments, I will be evaluating the relationship between frustration and confusion and the

- What is the relationship between frustration/confusion and student success?
- Is there a significant difference on student engagement when employing self-report method of affect data collection vs. unobtrusive field observation method?
- Does the ratio of frustration to confusion states significantly change in relation to the amount of support available or is there no correlation

combination of these two affective states that has destructive effect on learning or engagement.

Some of the research questions that this study will be investigating are as follows:

- What is the relationship between frustration and confusion and when is each beneficial or negative?
- Is there a sequential pattern in the occurrence of frustration and confusion or is there no temporal pattern

  between the amount of support provided and students' frustration and confusion? For example, are low-risk environments (e.g., adult/peer assistance available) related to lower amounts of confusion and no to low amount of frustration?
- Do students resume educational gameplay after being in a frustrated emotional state? If so, when and under what conditions do they resume their gameplay?

METHODS

For the purposes of the research questions, there are several data collection source that I will be using (e.g., gameplay data, observation data, video data). Two affect data collection tools will be employed in a within subject comparison study design and sequential pattern mining tools will be utilized in order to identify significant emotional state patterns and their interaction with student performance.

For the field observations of student engagement I will be using BROMP tool [9]. This holistic coding procedure will allow me to code student emotions and behavior while they are engaged in the game-like environment. I have also developed a comparable self-report tool in order to investigate the effects of self-report and unobtrusive observations on student engagement (i.e., if there is a significant change in the levels of student engagement). Given the prominent role that self-report has in the field and the possible drawbacks that are being discussed but not tested via empirical studies, I believe that this comparison will provide an insight on the use of self-report and how it compares to unobtrusive field observation methods. My preliminary hypothesis is that self-report will takes away from the learner's concentration on the learning environment and distracts their normal thought process. Moreover, I believe that self-report does not reflect on the entire learning process but rather concentrates on the moment in time when student is requested to provide a feedback or think aloud. With this within subject design and two measures of the same construct will help me verify or reject my initial hypothesis.

The observations will be carried out in a predetermined order of the classroom and computers. Each observer will be responsible for a separate set of students and will be given 15-20 second segments to record the observed behavior and affect (time will be fixed based on game's level of interactivity). During each segment, the dominant affective state will be recorded after which the observer will move to the next student repeating this process in a cycle until the end of class period. While class A will have no gameplay interruption because of employing BROMP observation tool, class B will have a pop up self-report measure of emotions that will interrupt students work every so often (currently it is set to 300 seconds).

Along with these data collection tools, students' gameplay screens along with their facial expressions will be continuously recorded in order to provide uninterrupted engagement data for sequential pattern mining purposes in the analysis stage. Finally, students gameplay will be recorded in a clickstream data format and will be synchronized with the emotions data in order to detect patterns in their emotional and behavioral states that affect their performance in game and vice versa. I will use the models of affective computing of over time data and create detectors that will automatically identify negative affect to support student persistence through failure and negative emotions.

ANALYSIS

The uniqueness of scale of educational datasets renders many traditional statistical methods inapplicable (Azarnough, Bekki, Runger, Bernstein, & Atkinson, 2013). Sequence analyses have been used by researchers on educational datasets in order to gain more granular overlook at the data and existing patterns (Sanjeev and Zytkow, 1995; Zaıane et al., 1998; Zaıane and Luo, 2001; Pahl and Donnellan, 2003; Wang, 2002; Shen et al., 2003; Wang et al., 2004). Sequence pattern analyses are concerned with the underlying patterns and orders of events in the dataset (Agrawal & Srikant, date; Zhou, Xu, Nesbit, & Winne, 2009). Once student data is converted into a simple ordered list of items (see Appendix G, Table 11), there are numerous ways to investigate this sequential data.

The main goal of this study is to find temporal and order based patterns of frustration and confusion in students' affect data. Therefore, having continuous affect data, which I will obtain by coding video recordings of students' faces, will allow me to perform inter-sequence distance analysis (Sabherwal & Robey, 1993) with optimal matching and clustering (Bailey, 1994; Tyron, 1939) of those sequences. These findings may potentially allow for design and development of better, affect-responsive digital games for learning.

Sequence mining techniques offer several approaches to pattern mining. My interests lie with the methods that look for the most frequent patterns across a set of sequences. This way I will be able to compare different students affective states and find most frequently occurring patterns of confusion-frustration interrelation (e.g., CFFCFCFFCCFF). In addition, I will be able to assess changes in students' engagement with the game (e.g., concentrated, bored) as a result of certain confusion-frustration patterns.

Another option is motif analysis (Shanabrook, Cooper, Woolf, & Arroyo, 2010). Unlike inter-sequence analysis, motif analysis allows us to look inside the sequence for patterns instead of comparing the sequences (Hardy, & Bryman, 2004). This is a great method to look into one students affect data over hours of gameplay (e.g., investigate two strategically selected students sequences separately in order to find what are the big differences based on an extra variable such as gender or success rate etc.). Moreover, with motif analysis, I will be able to investigate one students affect data and compare it to their gameplay performance patterns.

CURRENT STATUS OF WORK

Currently, I am in the middle of my dissertation study implementation. While pilot work was conducted in order to test the same hypotheses, some of the research questions along with the learning environment (game) have been altered. There have been methods implement in order to make sure the data collection captures continuous affect data and incorporates a self-report based comparison tool to unobtrusive field observations method that was implemented in the pilot work. I addition, self-report tool has been tested with the comparable grade level students in order to test the usability and the comprehensibility of the questionnaire's content. Preliminary results indicate on an interesting inter-relation between frustration, confusion and student performance in the learning environment.

REFERENCE

Baker, R.S.J.d., D'Mello, S.K., Rodrigo, M.M.T., & Graesser, A.C. 2010. *Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive Affective States during Interactions with Three Different Computer-Based Learning Environments*. International Journal of Human-Computer Studies, 68 (4), pp. 223-241.

Craig, S., Graesser, A., Sullins, J., Gholson, B. 2004. Affect and Learning: an Exploratory Look into the Role of Affect in Learning with AutoTutor. *Journal of Educational Media,* 29, 3. Taylor & Francis, London, UK, 241-250.

D'Mello, S.K., Graesser, A.C. 2011. The Half-Life of Cognitive-Affective States during Complex Learning. *Cognition and Emotion,* 25, 7 (2011). Taylor and Francis Group, London, UK, 1299-1308.

D'Mello, S.K.., Graesser, A.C. 2012. Dynamics of Affective States during Complex Learning. *Learning and Instruction*, 22, 2 (Apr. 2012). Elsevier B.V., Oxford, UK, 145-157.

Gee, J. P. (2004). Learning by design: Games as learning machines.*Interactive Educational Multimedia*, (8), 15-23.

Gee, J.P. 2007. *Good video games+ good learning: Collected essays on video games, learning, and literacy* (Mar. 2007). Peter Lang Pub Incorporated, Bern, Switzerland.

Lehman, B., D'Mello, S. K., & Graesser, A. C. 2012. *Interventions to Regulate Confusion during Learning*. In S. Cerri, W. Clancey, G. Papadourakis, & K. Panourgia (Eds.) Proceedings of the 11th International Conference on Intelligent Tutoring Systems (pp. 576-578). Berlin Heidelberg: Springer-Verlag.

Liu, Z., Pataranutaporn, V., Ocumpaugh, J., Baker, R.S.J.d. 2013. Sequences of Frustration and Confusion, and Learning. *Proceedings of the 6th International Conference on Educational Data Mining*, 114-120.

Ocumpaugh, J., Baker, R.S.J.d., Rodrigo, M.M.T. 2012. Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. *Training Manual version 1.0*. Technical Report. New York, NY: EdLab. Manila, Philippines: Ateneo Laboratory for the Learning Sciences.

Rodrigo, M. M. T., d Baker, R. S., D'Mello, S., Gonzalez, M. C. T., Lagud, M. C., Lim, S. A., ... & Viehland, N. J. 2008. Comparing learners' affect while using an intelligent tutoring system and a simulation problem solving game. In *Intelligent Tutoring Systems* (pp. 40-49). Springer Berlin Heidelberg.

Rodrigo, M.M.T., Baker, R.S.J.d., Jadud, M.C., Amarra, A.C.M., Dy, T., Espejo-Lahoz, M.B.V., Lim, S.A.L., Pascua, S.A.M.S., Sugay, J.O., Tabanao, E.S. 2009. Affective and Behavioral Predictors of Novice Programmer Achievement. *Proceedings of the 14th ACM-SIGCSE Annual Conference on Innovation and Technology in Computer Science Education*. 156-160.

Rodrigo, M.M.T., Baker, R.S.J.d., Nabos, J.Q. 2010. The Relationships between Sequences of Affective States and Learner Achievements. *Proceedings of the 18th International Conference on Computers in Education* (Putrajaya, Malaysia, Nov 29 - Dec 3, 2010).

VanLehn, K., Siler, S., Murray, C., Yamauchi, T., & Baggett, W. (2003). Why do only some events cause learning during human tutoring? *Cognition and Instruction, 21(3),* 209-249.

MOOC Leaner Motivation and Learning Pattern Discovery – A Research Prospectus Paper

Yuan “Elle” Wang
Teachers College, Columbia University
525 W 120th St.
New York, NY 10027, USA

elle.wang@columbia.edu

ABSTRACT

The landscape of online learning has evolved in a synchronous fashion with the development of the every-growing repertoire of technologies, especially with the recent addition of Massive Online Open Courses (MOOCs). Since MOOC platforms allow thousands of students to participate at the same time, MOOC participants can have fairly varied motivation. Meanwhile, a low course completion rate has been observed across different MOOC platforms. The first and initiated stage of the proposed research here is a preliminary attempt to study how different motivational aspects of MOOC learners correlate with course participation and completion, with motivation measured using a survey and participation measured using log analytics. The exploratory stage of the study has been conducted within the context of an educational data mining MOOC, within Coursera. In the long run, research results can be expected to inform future interventions, and the design of MOOCs, as well as increasing understanding of the emergent needs of MOOC learners as data collection extends beyond the current scope by incorporating wider disciplinary areas.

Keywords

Online Learning, learner Motivation, Massive Online Open Courses, Educational Data Mining

1. INTRODUCTION

In this paper, the first section presents a literature review on motivational studies of online learners in both the generic distance learning fields and the ones specific to the MOOC settings. The second section on methodology and progress explains methodologies applied for at the current research stage as well as planned analysis for the in-progress work presented. The third part is the discussion section where potential follow-up studies are proposed. Lastly, aspects on direction of future analysis and where advice is needed are stated.

2. LITERATURE REVIEW

2.1 Motivation of Online Learners

MOOC students have demonstrated varied motivation, beyond just solely utilitarian or learning goals [34]. Kizilcec, Piech, and Schneider [21] presented a classification method grouping MOOC learners by engagement levels. Clow [9] introduced a “funnel of participation” which conceptualized a pattern of highly unequal participation of MOOC learners and further confirmed the challenges of catering to varied needs of MOOC participants with current MOOC models.

High MOOC student dropout rates have been identified and studied by both researchers in academia and journalists [3, 8, 12, 22, 29], though debate is ongoing about the importance of dropout rate within the context of MOOCs. Furthermore, doubts have been cast upon whether completing the course assignments is necessary for MOOC participants [18, 23]. As Anderson [3] pointed out, many MOOC participants enroll in courses only to satisfy their initial curiosities with no intention of completing the course. Although course completion rate is by no means the only meaningful outcome, it has become one of the most discussed metrics in the MOOC environment.

Although MOOCs are a relatively new addition to the field of online learning, the construct of learner motivation has long been seen as essential to learning and learning outcomes. Dweck [13] argued that two key goals characterize most learners: learning goals and performance goals. Learning goals or mastery goals [2] indicates learners who strive to increase their competence and master the given skill; whereas performance goals suggest that learners seek to obtain favorable assessments from others. Since then, researchers have argued for two types of performance goals [17].

2.2 Goal Orientation of Online Learners

More recently, it has been argued that different goal orientations are actually symptoms of underlying student mind-sets. Students with growth mind-sets hold beliefs that intelligence is malleable; whereas students with fixed mind-set considers intelligence an unchangeable entity [14, 15]. A study conducted by Blackwell, Trzesniewski, and Dweck [7] measured and monitored seventh grade students of these two aforementioned mind-sets and found out that students with a growth mind-set outperform their counterparts who accept a fixed mind-set, over the long-term term.

Many motivation theorists have also argued that learning/mastery goals sustain intrinsic motivation [11, 16, 20]. According to Ryan and Deci [30], intrinsic motivation refers to executing a learning activity out of one’s inherent interests, whereas extrinsic motivation implies one intends to gain a separate outcome. MOOC students presumably consist of learners possessing each (or both) types of motivation. For example, out of intrinsic motivation, one might register for an educational data mining course purely out of curiosity. In contrast, out of extrinsic motivation, one might register for the same course because the skill sets covered in this course are useful for the student to advance in his or her career.

Intrinsic motivation has long been praised to predict effective learning; however such kind of motivation is also vulnerable to various non-supportive [31]. Keller and Suzuki [19] reasoned that students of E-learning platforms confront more motivational challenges due to that they have to work independently at a distance in most cases. It is also noticed that a relatively high dropout rates have been consistently observed across E-learning platforms [27], but these environments are generally more effective for students with self-regulated learning skill.

2.3 LAK and EDM on MOOCs

Among students who do not effectively regulate themselves during online learning, disengaged behaviors may emerge, such as “carelessness” -- not demonstrating a skill despite knowing it [32] and “gaming the system” – where learners use help and feedback provided by the online learning system to avoid learning [4]. It is not yet clear what the full range of disengaged behaviors are in MOOCs, but understanding this, and the role these behaviors play in the reduction of participation in MOOCs, is a key research question. Research applying learning analytics and data mining on MOOCs has helped identify distinct behavioral patterns. As an emerging filed, existing MOOC research has focused on classifying learner behavioral patterns by levels of engagement [9, 21]; adapting existing modeling techniques to MOOC data [28]; as well as developing new models for the MOOC environment [1, 35].

3. METHODOLOGY AND PROGRESS

3.1 Research Context

The exploratory stage of the proposed project has been carried out in the context of one MOOC, titled “Big Data in Education”, offered through Coursera by Teachers College, Columbia University. (https://www.coursera.org/course/bigdata-edu). This course spanned 8 continuous weeks with 8 weekly assignments. The weekly course composed of lecture videos. Students and teaching staff participated in forum discussion accompanying weekly course releases. The motivational survey was distributed through Coursera to students who have enrolled in this course prior to the course start date. This course has an enrollment of about 48,000 students.

3.2 Survey Data

Given the heterogeneity of the motivations of MOOC learners and the current interest in course completion and other measures of participation, this proposed research intends to expand our understanding of MOOC learners by analyzing how MOOC learners’ motivation correlates with students’ degrees of course completion and participation. Two categories of motivational aspects including both general items and MOOC-specific ones has been taken into account in this initial research attempt. Specifically, both MOOC-specific motivational items including those tested by existing MOOC studies [5, 26] and two subscales of the PALS survey [24] measuring goal orientation and academic efficacy are included in a pre-course survey. The MOOC-specific items include questions such as the familiarity of the MOOC environment and course content; whereas the PALS subscales focus on learner orientations towards learning or performance goals, across learning contexts. The survey was distributed through broadcast E-mail to all registered students. As of the end of the course, the pre-course survey has gathered 2,792 responses.

3.3 Log Analytics

Learning analytics and educational data mining techniques will also be applied to study student participation. Specifically, drawing from past research in monitoring participation within online learning [10, 25], this project will analyze indicators of participation such as use of discussion forums, quiz completion rate, and video usage. All the above-mentioned data collected will then be linked to the MOOC survey, and correlation mining will be used to determine which motivational indicators can predict participation metrics, employing FDR post-hoc correction [6] to control for running too many tests. Patterns of changes in participation across the course will also be analyzed by means of sequential pattern mining. Motivational response and participation will be used as predictors of MOOC completion.

4. PROPOSED CONTRIBUTION

Although MOOC participants represent a diverse population of learners with a diverse range of motivations, they do form a new learning community with common features. The low retention rate observed across different MOOC platforms is an important engagement issue to investigate further. A low retention rate may not be inherently negative in the context of MOOCs [21, 28, 35], given that MOOC participants registering for the same course can have very different motivations and goals in mind. At the same time, some failure to complete may not be simply due to lack of student interest in completing. Therefore, understanding MOOC learners’ motivation is imperative in helping us understand course participation and completion in this new context; which failure to complete is simply an artifact of student goals? Which is due to other factors, and therefore a problem to address? Research results of the present study is expected to inform intervention of MOOC learning environments as well as providing MOOC faculty members resources in planning and modifying their courses.

5. ADVICE NEEDED FOR FUTURE ANALYSIS

The first stage of analysis serves as initial research attempt to study how different motivational aspects of MOOC participants correlate with course participation and completion. Moving forward, research and advice is needed toward further understanding of learning patterns of MOOC learners and to inform future design of interventions.

Specifically, advice on how to extract MOOC data based on existing knowledge of other online learning platforms especially intelligent tutoring systems is needed for the progressing of the current research stage. For example, what are some of the knowledge components identified in ITS can be adapted in the MOOC models? How to synchronize forum textual data with clickstream data? How can unrecognized similarities or features between MOOCs and other well-studied online learning platforms be detected? Additionally, general and specific advice on designing experimental intervention is needed in ensuring internal validity, external validity, as well as research feasibility.

6. ACKNOWLEDGMENTS

The exploratory stage of the proposed MOOC motivation study has been supported by the MOOC Research Initiative funded by the Melinda and Gates Foundation and Athabasca University under the guidance of the project Co-PI Ryan Baker.

7. REFERENCES

[1] Adamopoulos, P. (2013). What Makes a Great MOOC? An Interdisciplinary Analysis of Student Retention in Online Courses. In *Proceedings of the 34th International Conference on Information Systems, ICIS* (Vol. 2013).

[2] Ames, C., & Archer, J. (1987). Mothers' beliefs about the role of ability and effort in school learning. Journal of Educational Psychology, 79, 409-414.

[3] Anderson, T. (2013). Promise and/or Peril: MOOCs and Open and Distance Education.

[4] Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". *Proceedings of ACM CHI 2004: Computer-Human Interaction*, 383-390.

[5] Belanger, Y., & Thornton, J. (2013). Bioelectricity: A Quantitative Approach Duke University's First MOOC.

[6] Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 289-300.

[7] Blackwell, L., Trzesniewski, K., & Dweck, C.S. (2007). Implicit theories of intelligence predict achievement across an adolescent transition: A longitudinal study and intervention. *Child Development, 78,* 246-263.

[8] Carr, N. (2012). The crisis in Higher Education. MIT Technology Review Retrieved from: http://www.technologyreview.com/featuredstory/429376/the-crisis-in-higher-education/

[9] Clow, D. (2013). MOOCs and the funnel of participation. Paper presented at the LAK'13: 3rd International Conference on Learning Analytics & Knowledge, Leuven, Belgium. Retrieved from Retrieved from http://oro.open.ac.uk/36657/1/DougClow-LAK13-revised-submitted.pdf

[10] Dawson, S. (2006). A study of the relationship between student communication interaction and sense of community. *The Internet and Higher Education*, *9*(3), 153-162.

[11] Deci, E. L., & Ryan, R. M. (1985). Intrinsic motivation and self-determination in human behavior. New York: Plenum.

[12] DeWaard, I., Abajian, S., Gallagher, M., Hogue, R., Keskin, N., Koutropoulos, A., & Rodriguez, O. C. (2011). Using mLearning and MOOCs to Understand Chaos, Emergence, and Complexity in Education. *International Review Of Research In Open & Distance Learning*, *12*(7), 94-11 5.

[13] Dweck, C. S. (1986). Motivational processes affecting learning. *American psychologist*, *41*(10), 1040.

[14] Dweck, C. S., & Leggett, E. L. (1988). A social-cognitive approach to motivation and personality. *Psychological review*, *95*(2), 256.

[15] Dweck, C. S. (2010). Mind-Sets and Equitable Education. *Principal Leadership*,*10*(5), 26-29.

[16] Elliot, A. J., & Harackiewicz, J. M. (1994). Goal setting, achievement orientation, and intrinsic motivation: A mediational analysis. *Journal of personality and social psychology*, *66*(5), 968.

[17] Elliot, A. J., & Church, M. A. (1997). A hierarchical model of approach and avoidance achievement motivation. *Journal of personality and social psychology*, *72*(1), 218.

[18] Fini, A. (2009). The technological dimension of a massive open online course: The case of the CCK08 course tools. *The International Review Of Research In Open And Distance Learning*, *10*(5).

[19] Keller, J., & Suzuki, K. (2004). Learner motivation and e-learning design: A multinationally validated process. *Journal of Educational Media*, *29*(3), 229-239.

[20] Heyman, G. D., & Dweck, C. S. (1992). Achievement goals and intrinsic motivation: Their relation and their role in adaptive motivation. Motivation and Emotion, 16, 231-247.

[21] Kizilcec, R. F., Piech, C., & Schneider, E. (2013, April). Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (pp. 170-179). ACM.

[22] Knox, J., et al. (2012). MOOC pedagogy: the challenges of developing for Coursera. Association for Learning Technology. Retrieved from: http://newsletter.alt.ac.uk/2012/08/mooc-pedagogy-the-challenges-of-developing-for-coursera/

[23] McAuley, A., Stewart, B., Siemens, G., & Cormier, D. (2010). The MOOC model for digital practice.

[24] Midgley, C., Maehr, M. L., Hruda, L., Anderinan, E.M., Anderman, L., Freeman, K. E., et al. (2000).*Manual for the Patterns of Adaptioe Learning Scales (PALS)*. Ann Arbor: University of Michigan.

[25] Ming, N. C., & Ming, V. (2012, September). Automated Predictive Assessment from Unstructured Student Writing. In *DATA ANALYTICS 2012, The First International Conference on Data Analytics* (pp. 57-60).

[26] MOOC @ Edinburgh 2013 – Report #1 (2013). MOOC @ Edinburgh 2013 – Report #1. University of Edinburgh, Edinburgh, Scotland. Retrieved from http://www.era.lib.ed.ac.uk/bitstream/1842/6683/1/Edinburgh%20MOOCs%20Report%202013%20%231.pdf

[27] Moore, M. G., & Kearsley, G. (2011). *Distance education: A systems view of online learning*. CengageBrain.com.

[28] Pardos, Z.A., Bergner, Y., Seaton, D., Pritchard, D.; In Press (2013) *Adapting Bayesian Knowledge Tracing to a Massive Open Online Course in edX.* In Proceedings of the 6th International Conference on Educational Data Mining. Memphis, TN.

[29] Pappano, L. (2012). The year of the MOOC. *The New York Times*, *2* (12), 2012.

[30] Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, *55*(1), 68.

[31] Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, *25*(1), 54-67.

[32] San Pedro, M.O.C., Baker, R., Rodrigo, M.M. (2011) Detecting Carelessness through Contextual Estimation of Slip Probabilities among Students Using an Intelligent Tutor for Mathematics. *Proceedings of 15th International Conference on Artificial Intelligence in Education*, 304-311.

[33] Siemens, G. (2005). Connectivism: A learning theory for the digital age. *International Journal of Instructional Technology and Distance Learning*, *2*(1), 3-10.

[34] Siemens, G. (2006). Connectivism: Learning theory or pastime of the self-amused. *Retrieved February*, *2*, 2008.

[35] Veeramachaneni, K., Dernoncourt, F., Taylor, C., Pardos, Z., & O'Reilly, U. M. (2013, June). MOOCdb: Developing Data Standards for MOOC Data Science. In *AIED 2013 Workshops Proceedings Volume* (p. 17).

[36] White, R. W. (1959) Motivation reconsidered: the concept of competence, Psychological Review, 78, 44-57.

Personalization and Incentive Design in E-learning Systems

Avi Segal
Dept. of Information Systems Engineering
Ben-Gurion University,
Beer-Sheva 84105, Israel
avisegal@gmail.com

ABSTRACT

My thesis focuses on the design of systems to augment existing e-learning software in a way that supports both teachers and students. It addresses three central challenges: personalization of educational content to students, techniques for machine-generated interventions, and incentive designs to enhance students' learning. For each of these problems I will synthesizes approaches from informational retrieval and social choice theory. My results thus far have included a novel algorithm for sequencing content in e-learning system that uses collaborative filtering to generate a difficulty ranking over the test questions, without needing to predict students' performance directly on these questions. The algorithm was able to outperform state-of-the-art approaches from the literature on two different data sets containing millions of records. My future efforts will be directed to extending these results and to generalize my approach to the problems of intervention and incentive designs.

1. INTRODUCTION

My thesis focuses on the design of systems for e-learning that support both students in their learning processes and teachers in their understanding of how students learn. I focus on augmenting existing educational software already used in schools where impact can be achieved and for which large amounts of data is available for analysis from past students interaction. My work addresses three central challenges in the design of such systems by synthesizing techniques from information retrieval and social choice:

The first challenge is *personalization of educational content to students.* Educational content is now accessible to student communities of varied backgrounds, learning styles[1] and needs. There is thus a growing need for personalizing educational content to students in e-learning systems in a way that adapts to students' individual needs [10, 1]. My approach towards such personalization is to sequence students' questions in a way that best matches their learning styles or gains [2, 12]. To this end, I use a collaborative filtering approach [3], to generate a difficulty ranking over a set of questions for a target student by aggregating the known difficulty rankings over questions solved by other, similar students. The difficulty rankings of similar students is combined using social choice theory [6] to produce the best difficulty ranking for the target student.

The second challenge is *intelligent intervention for students.* Two foundational principles of a collaborative system [7, 4, 8] are that (1) the system pursues all possible avenues for doing its tasks, and provides support to all participants in the system. (2) the system is lightweight and avoids disrupting other participants as much as possible. Within the context of education, such a system will guide students' interactions in a way that best adapts to their abilities and learning styles, while minimizing the amount of intervention, allowing for activities that yield educational gains through explorations. To minimize intrusion, the system must be able to model the effect of interruption on the students' behavior with the educational system over time. For example, the system should be able to decide not to intervene when the student is off-track, because it predicts that this exploratory behavior will yield further educational gains. To this end, I will use approaches from the recommendation systems literature to compare students' past interactions with that of similar students, to infer the best point in time when to interrupt the user.

The third challenge is *incentive design* for influencing the behavior of students (whether as individuals or group members). Although incentive structures have been studied extensively in psychology and economics (and most recently, human computation), there has been scarce work on the design and analysis of incentives in educational contexts. To this end, I plan to model students' uses of educational content (e.g., on-line course forums, problem sets, etc...) and to compare the efficacy of different incentives (e.g., points, badges and peer pressure) towards steering student behavior and learning.

2. INITIAL RESULTS

My research efforts thus far have focused on the first challenge, that of personalizing educational content to students in e-learning systems. I developed a novel algorithm for sequencing content in e-learning systems that directly creates

[1] learning styles: e.g. as defined by [5] covering perception, input, organization, processing and understanding aspects.

a "difficulty ranking" over new questions. My approach is based on collaborative filtering [3], which generates a difficulty ranking over a set of questions for a target student by aggregating the known difficulty rankings over questions solved by other, similar students. The similarity of other students to the target student is measured by their grades on common past question, the number of retries for each question, and other features. Unlike other uses of collaborative filtering in education, this approach directly generates a difficulty ranking over the test questions, without predicting students' performance directly on these questions, which may be prone to error.[2]

The algorithm, called EduRank, weighs the contribution of these students using measures from the information retrieval literature. It allows for partial overlap between the difficulty rankings of a neighboring student and the target student, making it especially suitable for e-learning systems where students differ in which questions they solve. The algorithm extends a prior approach for ranking items in recommendation systems [9], which was not evaluated on educational data, in two ways: First, by using social choice theory to combine the difficulty rankings of similar students and produce the best difficulty ranking for the target student. Second, EduRank penalizes disagreements in high positions in the difficulty ranking more strongly than low positions, under the assumption that errors made in ranking more difficult questions are more detrimental to students than errors made in ranking of easier questions.

I evaluated EduRank on two large real world data sets containing tens of thousands of students and about a million records. I compared the performance of EduRank to a variety of personalization methods from the literature, including the prior approach mentioned above as well as other popular collaborative filtering approaches such as matrix factorization and memory-based K nearest neighbors. I also compared EduRank to a (non-personalized) ranking created by a domain expert. EduRank significantly outperformed all other approaches when comparing the outputted difficulty rankings to a gold standard.

3. FUTURE CHALLENGES AND ANTICIPATED CONTRIBUTION

My next efforts are going to focus on incentive design and intervention policies in e-learning systems. For this, I'm going to address, among others, the following topics and will be happy to get the advise of the consortium on them:

- Extrinsic vs. intrinsic motivation and respective incentives in educational systems
- Usage of persuasion technologies to steer on-line learning behavior
- Badges as an reputation incentive mechanism
- Additional game mechanics to be adapted for the context of my research
- Comparing hints to other intervention methods in the context of exploration and learning
- Influencing learning and mastery through personal vs. group incentives
- Investigating additional social choice methods for combining peers influence on personalization and intervention

My anticipated contribution will include developing novel modeling algorithms for users in e-learning systems, designing incentive mechanisms for these systems and constructing and evaluating personalization and intervention mechanisms for users by reasoning about how they respond to these interventions over time. I will evaluate my approaches in real world e-learning environments.

4. REFERENCES

[1] Y. Akbulut and C. S. Cardak. Adaptive educational hypermedia accommodating learning styles: A content analysis of publications from 2000 to 2011. *Computers & Education*, 58(2):835–842, 2012.

[2] H. Ba-Omar, I. Petrounias, and F. Anwar. A framework for using web usage mining to personalise e-learning. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 937–938. IEEE, 2007.

[3] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[4] P. Cohen and H. Levesque. Teamwork. *Nous*, pages 487–512, 1991.

[5] R. M. Felder and L. K. Silverman. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988.

[6] P. C. Fishburn. *The theory of social choice*, volume 264. Princeton University Press Princeton, 1973.

[7] B. Grosz and S. Kraus. The evolution of sharedplans. *Foundations and Theories of Rational Agency*, pages 227–262, 1999.

[8] D. Kinny, E. Sonenberg, M. Ljungberg, G. Tidhar, A. Rao, and E. Werner. Planned team activity. *Artificial Social Systems*, pages 227–256, 1994.

[9] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 83–90. ACM, 2008.

[10] D. Sampson, C. Karagiannidis, D. Kinshuk, et al. Personalised learning: educational, technological and standarisation perspective. *Digital Education Review*, (4):24–39, 2010.

[11] A. Toscher and M. Jahrer. Collaborative filtering applied to educational data mining. *KDD Cup*, 2010.

[12] L. Zhang, X. Liu, and X. Liu. Personalized instructing recommendation system based on web mining. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 2517–2521. IEEE, 2008.

[2] To illustrate, in the KDD cup 2010, the best preforming grade prediction algorithms exhibited prediction errors of about 28% [11]

Workshops

G-EDM 2014: Graph-Based Educational Datamining

Collin F. Lynch
Center for Educational Informatics
North Carolina State University
Raleigh, North Carolina, U.S.A
cflynch@ncsu.edu

Tiffany M. Barnes
Center for Educational Informatics
North Carolina State University
Raleigh, North Carolina, U.S.A
cflynch@ncsu.edu

Graph data has become increasingly prevalent in data-mining and data analysis generally. Many types of data can be represented naturally as graphs including social network data, log traversal, and online discussions. Moreover recent work on the importance of social relationships, peer tutoring, collaboration, and argumentation has highlighted the importance of relational information in education including:

- Graphical solution representations such as argument diagrams and concept maps;
- Graph-based models of problem-solving strategies;
- User-system interaction data in online courses and open-ended tutors;
- Sub-communities of learners, peer-tutors and project teams within larger courses; and
- Class assignments within a larger knowledge space.

Our goal in this workshop was to highlight the importance of graph data and its relevance to to the wider EDM community. We also sought to foster the development of an interested community of inquiry to share common problems, tools, and techniques. We solicited papers from academic and industry professionals focusing on: common problems, analytical tools, and established research. We also particularly welcomed new researchers and students seeking collaboration and guidance on future directions. It is our hope that the papers published here will serve as a foundation for ongoing research in this area and as a basis for future discussions.

The papers presented at the workshop (see [1]) covered a range of topics. Kovanovic, Joksimovic, Gasevic & Hatala focus on evaluating social networks, and specifically on the development of social capital and high-status individuals in a course context while Catete, Hicks, Barnes, & Lynch describe an online tool designed to promote social network formation in new students. Similar work is also described by by Jiang, Fitzhugh & Warschauer who focus on the identification of high-connection users in MOOCs.

Other authors turned to the extraction of plan and hint information from course materials and user logs. Belacel, Durand, & Laplante define a graph-based algorithm for identifying the best path through a set of learning objects. Kumar describes an algorithm for the automatic construction of behavior graphs for example-tracing tutors based upon expert solutions and Dekel & Gal in turn consider plan identification to support automatic guidance. Two further papers by Vaculík, Nezvalová & Popelínský, and by Mostafavi & Barnes, apply graph analysis techniques to the specific domain of logic tutoring and, in particular, on the classification of student solutions and to the evaluation of problem quality.

And finally several authors chose to present general tools for the evaluation of graphical data. Lynch describes Augmented Graph Grammars, a formal rule representation for the analysis of rich graph data such as argument diagrams and interconnected student assignments, and details an implementation of it. Sheshadri, Lynch, & Barnes present InVis a visualization and analysis platform for student interaction data designed to support the types of research described above. And McTavish describes a general technique to support graph analysis and visualization particularly for student materials through the use of interactive hierarchical edges. We thank the included authors for their contributions to the discussion and look forward to continued research.

1. REFERENCES

[1] S. Gutierrez-Santos and O. C. Santos, editors. *EDM 2014 Extended Proceedings: Workshop Proceedings of the 7th International Conference on Educational Data Mining. London, United Kingdom, June 4-7, 2014.* CEUR-WS, 2014.

Workshop of Non-Cognitive Factors and Personalization for Adaptive Learning @ EDM 2014

Steven Ritter
Stephen E. Fancsali
Carnegie Learning, Inc.
437 Grant Street, Suite 918
Pittsburgh, PA 15219 USA
1-888-851-7094 {x122, x219}
{sritter, sfancsali}
@carnegielearning.com

1. INTRODUCTION

Personalization of learning in computer-based environments is a major initiative in education today - accordingly, the United States Department of Education recently cited personalized, individualized, and differentiated approaches to instruction as a grand challenge in their National Education Technology Plan [1]. Many computer-based learning environments adapt to individual learners based on cognitive factors like skill mastery. Recent research has been directed at improving personalization in such systems by harnessing non-cognitive factors such as learner affect, motivation, preferences, self-efficacy, self-regulation and grit. The importance and promise of such work is noted in a recent draft report from the U.S. Department of Education's Office of Educational Technology [2], emphasizing non-cognitive factors like grit, tenacity, and perseverance for learning outcomes, notes "important opportunities to leverage new and emerging advances in technology (e.g., educational data mining, affective computing, online resources, tools for teachers) to develop unprecedented approaches for a wide range of students." To capitalize on these opportunities, future work on non-cognitive factors will require not only an understanding of learning science theory to inform meaningful scientific work on non-cognitive factors but also a good handle on contemporary data mining techniques to harness large scale, "big data" that is now available in a wide variety of educational technology contexts.

Recognizing that data-mining techniques offer a unique opportunity to collect and analyze information about non-cognitive factors, which can then be used to adapt instruction, research programs at various universities, companies, and government organizations focus on the influence of non-cognitive factors on student learning. Across these institutions, methods and approaches for varying learning environments differ greatly. Bringing together researchers to discuss similarities and differences between approaches is both important and timely.

As such, this workshop brings together researchers studying non-cognitive factors in a variety of environments and platforms, using various experimental, measurement, data mining, and statistical methods. In addition to presenting on-going research on specific non-cognitive factors and their impact of learning outcomes, this workshop is a venue to address common methodological questions and problems: what are suitable ways to measure, observe, detect, sense, or infer factors like learner affect or mood? How, and to what degree, or various non-cognitive factors associated with a variety of learning outcomes? How do interventions efficaciously "honor" learner preferences or motivate students? What are appropriate levels of granularity for analyses and interventions (e.g., at a level of problems, topical sections, or entire curricula)? How do lessons learned about one particular factor or learner population generalize to other non-cognitive factors and learner populations? How do theoretical advances in the learning sciences and data mining techniques complement each other to help answer questions about non-cognitive factors and personalization in disparate, adaptive learning environments in the era of "big data?"

Case studies presented in this workshop provide a host of promising answers to these questions as well as insights into on-going research and directions for future work that will seek to leverage non-cognitive factors in technology-based (and mediated) learning environments to improve learning outcomes. Finally, and appropriately for the present venue, presenters represent a diversity of approaches to analyzing and mining rich learner and/or instructional data generated by modern learning platforms and environments.

2. ACKNOWLEDGMENTS

The workshop co-chairs gratefully acknowledge members of the workshop program committee: Vincent Aleven, Ryan S.J.d. Baker, Matt Bernacki, Alan Drimmer, Andrew Krumm, Timothy Nokes-Malach, John Stamper, Candace Walkington, and Michael Yudelson.

3. REFERENCES

[1] U.S. Department of Education, Office of Educational Technology. 2010. Transforming American Education: Learning Powered by Technology (National Education Technology Plan 2010). Washington, D.C. http://www.ed.gov/sites/default/files/netp2010.pdf

[2] U.S. Department of Education, Office of Educational Technology & SRI, Center for Technology in Learning. 2013. Promoting Grit, Tenacity, and Perseverance: Critical Factors for Success in the 21st Century (DRAFT). http://www.sri.com/sites/default/files/publications/oet-grit-report_updated.pdf

Approaching Twenty Years of Knowledge Tracing: Lessons Learned, Open Challenges, and Promising Developments

Michael Yudelson
Carnegie Learning, Inc.
Pittsburgh, PA, USA
myudelson@carnegielearning.com

José González-Brenes
Pearson Research & Innovation Network
Philadelphia, PA, USA
jose.gonzalez-brenes@pearson.com

Michael Mozer
University of Colorado
Boulder, CO, USA
mozer@colorado.edu

Bayesian Knowledge Tracing is a popular method for student modeling because of its capability to infer student's dynamic knowledge state in real time as the student is solving a series of problems (Corbett & Anderson, 1995). After its introduction in 1995, many extensions to the original technique have been proposed to improve. Variants include: fitting model parameters to individuals rather than populations (Lee & Brunskill, 2012), crossing skill and student parameters (Yudelson, Koediger, & Gordon, 2010), contextualizing model parameters based on past and current usage of an intelligent tutoring system (Baker, Corbett, & Aleven, 2008, Baker et al., 2010; González-Brenes, 2014; Pardos et al., 2010) and on latent characteristics of students and problems (Khajah et al, 2014), clustering similar students and sharing parameters among them (Pardos et al, 2012), soft sharing of parameters via hierarchical Bayesian inference (Beck & Chang, 2007; Beck, 2007), and considering knowledge state as a continuous variable (Sohl-Dickstein, 2013; Smith et al., 2004).

At this workshop we look back at the twenty years of research on Bayesian Knowledge Tracing and examine the problems that are actively investigated by educational data mining researchers today.

A tangible number of papers are discussing practical questions of training Bayesian Knowledge Tracing models. Derrick Coetzee looks at the amount of data that is necessary to produce a usable model. According to the author, in the situation when there is not enough of training data, the model would perform poorly. In this work, synthetic data is used to estimate the standard deviation of the prediction error that is found to be proportional to the inverse of the size of the training set. Also, author finds that parameter values close to 0 or 1 are easier to arrive at when facing the shortage of training data.

Another paper by Dhanni et al. discusses alternative objective functions used for training BKT models. Traditionally, log-likelihood is used. However, authors find that rood mean squared error (RMSE) when used leads to a more accurate model, when log-likelihood and another metric, area under the ROC curve, result in less accurate model. The authors' conclusions are based on the distance metric between ground truth parameters used to generate synthetic data and the model parameters under consideration as characterized by the corresponding value of the metric.

Nelimarkka and Ghori look at BKT performance in situations when priors of the skill masteries assume extremely low or extremely high values. The authors find that extremely high values of priors lead to worse model performance.

An interesting work by Rosenberg-Kima and Pardos seeks to detect whether the data model was trained on is synthetic or real. Authors stipulate that given this information it would be possible to better define the goodness of model's fit.

Another group of papers is discussing extensions to the BKT model. Xu et al. talk about using a signal from a portable electroencephalography (EEG) device as a sensor for determining student's emotional state. Authors report that EEG-informed BKT model results in significant improvement of the model performance.

Zhu et al. talk about a special Sequence of Actions (SOA) model that takes advantage of the student attempts and hit requests from the previous problem the student solved. A two-step modeling approach is compared to the standard BKT and the assistance model (AM). Results are showing that the SOA model has a reliably better accuracy than BKT and AM.

Hawkins and Heffernan look at the problem of correlating student performance on the current problem and the previous problem when both address similar skills. Authors introduce a BKT variant that takes into account the similarity between the current and previous problem. It is shown that the new model can capture the effect of problem similarity on performance, and moderately improve accuracy on skills with many dissimilar problems.

Student gaming behaviors are an intensively researched area today. As it was shown before, misuse of intelligent tutoring system's hints could endanger learning. Schultz and Arroyo present a variation of BKT model that predicts gaming behaviors and retains the prediction of performance. The new model is compared to standard BKT model as well as the models that target engagement specifically.

Workshop on Feedback from Multimodal Interactions in Learning Management Systems

Lars Schmidt-Thieme
ISMLL, University of Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

Arvid Kappas
Jacobs University Bremen, Germany
a.kappas@jacobs-university.de

Carles Sierra
IIIA, Spanish Research Council, Spain, University of Technology, Sydney
sierra@iiia.csic.es

Emanuele Ruffaldi
PERCRO, Scuola Superiore Sant'Anna, Pisa, Italy
e.ruffaldi@sssup.it

Ruth Janning
ISMLL, University of Hildesheim, Germany
janning@ismll.uni-hildesheim.de

ABSTRACT

Virtually all learning management systems and tutoring systems provide feedback to learners based on their time spent within the system, the number, intensity and type of tasks worked on and past performance with these tasks and corresponding skills. Often the analysis of learner / system interactions is limited to these high-level interactions, and does not make good use of all the information available in much richer interaction types such speech and video. In this workshop we brought together researchers and practitioners interested in developing data-driven feedback and intervention mechanisms based on rich, multimodal interactions of learners.

1. WORKSHOP PAPERS

The workshop contributions addressed topics from affect recognition in intelligent tutoring systems to online learning and collaborative learning.

Interventions During Student Multimodal Learning Activities: Which, and Why? [1] This paper describes a Wizard-of-Oz study investigating the potential of Automatic Speech Recognition together with an emotion detector support young children in their exploration and reflection.

Multimodal Affect Recognition for Adaptive Intelligent Tutoring Systems [2]. This paper aims to support student performance prediction and sequencing models for intelligent tutoring systems by cues taken from multimodal input such as speech input from the students.

Collaborative Assessment [3]. In this paper is introduced an automated assessment service for online learning support in the context of communities of learners. The goal is to introduce automatic tools to support the task of assessing massive number of students as needed in MOOCs.

Mining for Evidence of Collaborative Learning in Question & Answering Systems [4]. This paper illustrates how the collaborative nature of feedback can be measured in online platforms, and how users can be identified that need to be encouraged to participate in collaborative activities.

Creative Feedback: a Manifesto for Social Learning [5]. In order to ground and motivate the definition and use of "creative feedback" the paper takes a historical look at the two concepts of creativity/creative and feedback.

2. CONCLUSIONS

The contributions present a growing interest in the adoption of technique of real-time and offline feedback for improving the learning process. There are open research question on the selection of the best feedback mechanism, and on integrating such feedback into learning analytic frameworks.

3. REFERENCES

[1] Grawemeyer, B., Mavrikis, M., Gutierrez-Santos, S. and Hansen, A. 2014. Interventions during student multimodal learning activities: which, and why? In Workshop Proceedings of EDM 2014.

[2] Janning, R., Schatten, C. and Schmidt-Thieme, L. 2014. Multimodal Affect Recognition for Adaptive Intelligent Tutoring Systems. In Workshop Proceedings of EDM 2014.

[3] Gutierrez, P., Osman, N. and Sierra, C. 2014. Collaborative Assessment. In Workshop Proceedings of EDM 2014.

[4] Loeckx, J. 2014. Mining for Evidence of Collaborative Learning in Question & Answering Systems. In Workshop Proceedings of EDM 2014.

[5] d'Inverno, M. and Still, A. 2014. Creative Feedback: a manifesto for social learning. In Workshop Proceedings of EDM 2014.