2014

Mining and Analyzing the Academic Network

Zaihan Yang
Lehigh University

Mining and Analyzing the Academic Network

by

Zaihan Yang

A Dissertation
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Doctor of Philosophy
in
Computer Science

Lehigh University
May 2014

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Zaihan Yang
Mining and Analyzing the Academic Network

Date

Prof. Brian D. Davison, Dissertation Director, Chair
(Must Sign with Blue Ink)

Accepted Date

Committee Members

Prof. Mooi Choo Chuah

Prof. Jeff Heflin

Prof. Roger Nagel

Prof. Catherine Ridings

Acknowledgments

First of all, I owe my sincerest thanks to my advisor, Prof. Brian D. Davison, for his encouragement, patience, guidance and insightful advice throughout my PhD study. Prof. Davison provides me a stable and enjoyable studying environment, where he gives me much freedom to explore various kinds of research topics and encourages me to work independently. As a research advisor, his passion towards research, hard-working spirits, good sense of finding problems, and special insistence on details, not only in developing ideas, but also in conducting and analyzing experiments as well as organizing and writing papers, have greatly inspired me and influenced my perspectives towards academic research. He has shaped me from many aspects from figuring out working directions to writing papers. I am grateful to him for the fruitful discussions, stimulating feedbacks and practical helps I got from him, and will never forget the many late-nights and early-mornings that he spent with us revising papers and preparing for submissions. Beyond just a research advisor, Prof. Davison is a kind and generous friend, who is always considerate and willing to provide help. I will always remember the great patience and support he offers when I had some difficulties in obtaining visa back to school. Without his advice and support, the completion of this dissertation would be impossible.

I would also like to thank my committee members: Prof. Mooi Choo Chuah, Prof. Jeff Heflin, Prof. Roger Nagel, and Prof. Catherine Ridings, for the help, feedback and advice they have provided during the writing of this dissertation and in my general examination. Sincere gratitude also goes to Prof. Charles Smith, who provides me the opportunity to work as the teaching assistant for several semesters.

Many thanks to my WUME lab mates: Liangjie Hong, Dawei Yin, Na Dai, Xiaoguang Qi, Zhenzhen Xue, Ovidiu Dan, Jian Wang, Nie Lan and Will West, and friends from other labs: Dezhao Song, Xingjian Zhang, Jin Chen, Yingjie Li and Yang Yu, for their valuable discussions and participation in my experiments. I would especially thank Liangjie, Dawei, Na and Xiaoguang, for their help, comfort and encouragement when I had difficulties in research. I appreciate those happy days spent together with these fellow students. It was full of laughters and fun. No

matter how far away we would be in the future, such a long-time experience studying together and the friendship among us will be precious and long-lasting memory in my mind.

Last but not the least, I am deeply indebted to my family. My deepest thanks go to my dearest father Binwen Yang and mother Dianzi Chen. Their endless love and unconditional support is the most powerful source of strength leading me move forward and going through all the difficult times in my life. Special gratitude also goes to my husband, Feng Zhu, who has always been my best friend and mental support. Without his love and companionship, I could not have overcome such difficulties and completed this work. I would also thank Feng's parents, who are always kind and supportive to me.

Much of the work presented in this dissertation was supported in part by a grant from the National Science Foundation under award IIS-0545875, as well as an equipment grant from Sun Microsystems. 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.

Dedication

Lovingly dedicated to my parents, Binwen Yang & Dianzi Chen and Feng

Contents

List of Tables

List of Figures

Abstract

Social Network research has attracted the interests of many researchers, not only in analyzing the online social networking applications, such as Facebook and Twitter, but also in providing comprehensive services in scientific research domain. We define an Academic Network as a social network which integrates scientific factors, such as authors, papers, affiliations, publishing venues, and their relationships, such as co-authorship among authors and citations among papers. By mining and analyzing the academic network, we can provide users comprehensive services as searching for research experts, published papers, conferences, as well as detecting research communities or the evolution of hot research topics. We can also provide recommendations to users on with whom to collaborate, whom to cite and where to submit.

In this dissertation, we investigate two main tasks that have fundamental applications in the academic network research. In the first, we address the problem of expertise retrieval, also known as expert finding or ranking, in which we identify and return a ranked list of researchers, based upon their estimated expertise or reputation, to user-specified queries. In the second, we address the problem of research action recommendation (prediction), specifically, the tasks of publishing venue recommendation, citation recommendation and coauthor recommendation. For both tasks, to effectively mine and integrate heterogeneous information and therefore develop well-functioning ranking or recommender systems is our principal goal. For the task of expertise retrieval, we first proposed or applied three modified versions of the PageRank-like algorithms into citation network analysis; we then proposed an enhanced author-topic model by simultaneously modeling citation and publishing venue information; we finally incorporated the pair-wise learning-to-rank algorithm into traditional topic modeling process, and further improved the model by integrating groups of author-specific features. For the task of research action recommendation, we first proposed an improved neighborhood-based collaborative filtering approach for publishing venue recommendation; we then applied our proposed enhanced author-topic model and demonstrated its effectiveness in both cited

author prediction and publishing venue prediction; finally we proposed an extended latent factor model that can jointly model several relations in an academic environment in a unified way and verified its performance in four recommendation tasks: the recommendation on author-co-authorship, author-paper citation, paper-paper citation and paper-venue submission. Extensive experiments conducted on large-scale real-world data sets demonstrated the superiority of our proposed models over other existing state-of-the-art methods.

Chapter 1

Introduction

1.1 Overview

A social network [182] is a social structure consisting of individual entities (represented as nodes), which are connected via relationships (represented as links). With the rapid growth of social media, especially online networking applications such as sharing sites (e.g., YouTube [198], Flickr [53]), instant message applications (e.g., MSN, Skype), microblogs (e.g., Twitter [175], Weibo [160]), social communication networks (e.g., Facebook [48], Myspace, RenRen [142]) and professional networks (Linkedin [103]), people are more closely linked with each other, and are more likely to exchange information, share messages, opinions, and (or) personal experience or status over online social networking. Social network has greatly reshaped the pattern of people's lives.

The **academic network**, according to our definition, is a social network, particularly constructed for the academic research environment to model academic entities as well as their mutual relationships. In an academic network, nodes are often associated with academic (scientific) factors, such as authors (researchers), papers, publishing venues, and affiliations, and links are representing the relationships among academic factors, such as the co-authorships among authors and citationships among papers.

The research in mining and analyzing the academic network has attracted much attention these years due to many applications arising in the academic environment. For example, one of the information needs of many academic/research committees or organizations is to evaluate the expertise of a researcher in a specific domain, as it plays an important role in determining people's job promotion, funding support application, scientific awards assignments as well as paper reviewing assignments. This information need can be satisfied by the research task known as expert search or expertise ranking, where given a query representing a research domain, we can identify and generate a ranked list of researchers based on their estimated expertise. For another example, researchers are often in need of finding the most relevant or the most recent publications related to their own research, even though people can fulfill this task manually by themselves, a well-developed automatic recommender system can largely relieve the burden of users and provide more accurate and complete list of papers for the researchers to refer. This kind of application and information need has stimulated the research task as citation recommendation and prediction. Other important applications in academic research domain include recommending or predicting future co-authors to collaborate, recommending publishing venues for a paper to consider to submit, detecting research communities, predicting future research hot topics, and etc. All these applications have motivated the research in mining and analyzing the academic network.

On the other hand, with the rapid development of online digital libraries, the proliferation of large quantities of scientific literature provides us abundant opportunities to extract the textual content of scientific factors (i.e., publishing papers) as well as their mutual relationships (citation, co-authorship), and therefore makes the research in mining and analyzing academic network workable and applicable. Several widely-deployed search engines, such as Microsoft Academic Search[1], ArnetMiner[2], have been particularly developed for academic search purpose, and demonstrated their success.

There are several properties that are associated with the academic network.

[1]http://academic.research.microsoft.com/

[2]http://www.arnetminer.org

Figure 1.1 shows an example of a subset of a typical academic network with it properties.

- **Heterogeneous** First of all, it is often a heterogeneous network, composed of multiple types of academic entities (authors, papers, venues) and academic relationships, for example, the bidirectional co-author relationships, and directional citation relationships. To better model and integrate this heterogeneous information, and to evaluate their individual importance remains one of the challenges in academic network research.

- **Community-based** Secondly, within the network, we can discover communities, and these communities (clusters) are often related with specific topics. For example, as shown in the illustrated figure, researchers can form different communities: IR-based community, Networking-based, AI-based or Business-based, determined by the main research domains the researchers focus on. Papers can construct such communities in a similar way. Entities with one community (either researchers or papers) would have more and closer interactions among themselves and thus form a community than other researchers from outside of this community.

- **Temporal dynamic** Thirdly, the academic network remains dynamically changing over time. For example, researchers transfer to different institutions, papers keep on attracting new citations, and researchers gradually accumulate their research experience by publishing new papers and attracting new citations and therefore gaining reputation and growing to researchers with higher expertise, all of which emphasize the importance of temporal information in academic network analysis.

To provide effective models that can represent all these properties remains challenging.

In this dissertation, we pay particular attention and focus on two main tasks that have fundamental applications in academic network research: the task of modeling expertise retrieval, also known as expert search, expertise ranking, and the

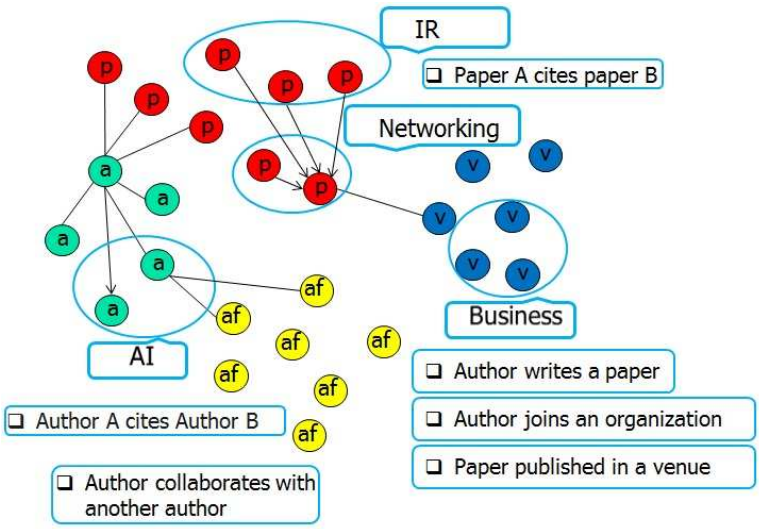

Figure 1.1: Academic Network and its properties

task of research action prediction and recommendation. To be more specific, we focus on prediction and recommendation over three research actions: publishing venue prediction/recommendation, coauthorship prediction/recommendation, and citation prediction and recommendation. **For both tasks, properly mining and effectively integrating heterogeneous information and therefore developing well-functioning ranking or recommendation systems is the principal and targeted goal**.

1.1.1 Modeling Expertise Retrieval

Modeling expertise retrieval, also known as expert (people) search or expertise ranking, has been a promising research topic due to the ever-growing trend of users' information needs to identify and interact with other people with relevant expertise (knowledge).

Resorting back to the development history of information retrieval (IR) technology, much of the research has been focused on traditional document (textual)

retrieval in the 1970s and 1980s, whose main task is to efficiently identify documents that are relevant to some information need. With the advent of Web, which has generated a large and ever-increasing volume of online documents (web pages), identifying relevant documents becomes difficult by manual browsing. Various web search engines have therefore been developed to facilitate users browsing and searching over the Internet. With a web search engine, people often represent their information need as a query, and the search engine would then return a ranked list of documents with regards to their estimated relevance to that query. More recently, the rapid increase in the amount of information available online has led to people's information needs going beyond the traditional plain document retrieval. Instead, they begin to search for other kinds of entities, such as books, movies, music, and restaurants. 'People', is one particular kind of such entity.

Searching for people with relevant expertise is of great importance not only to employees in an organization, but also to online social media users, as well as to researchers in the academic domain. In the enterprise organization settings, it is believed that finding the right person with an appropriate skill or knowledge is often crucial to the success of a project being undertaken [125]. In an online social media environment, users are often interested in finding other users who share similar interests (e.g., Flickr, Twitter), or identifying users who can provide the most valuable answers to a proposed question (e.g., Yahoo! Answers). In the academic environment, it is also of great importance to evaluate the expertise of a researcher in a specific domain, as it can offer help in determining the job promotion and funding assignment.

Initial research in evaluating people's expertise is mainly focused on unifying disparate databases of the organization [192] or simply counting bibliographic records of researchers [58]. With the advent of TREC enterprise track initiated in 2005 [34], which provides an open platform with two standard data sets and evaluation benchmark, much more research efforts in the computer science and information retrieval community have been devoted into expertise ranking research. Several groups of new models have been proposed and developed ever since. The TREC data sets, however, are more enterprise oriented. Later on, with the rapid development of

online digital library, one group of research focus has been paid on identifying experts within a pure academic environment. Generally speaking, there exists two main methodologies for expertise ranking: the content-based approach, where the expertise of a researcher is often characterized by examining the documents associated with them, and the graph-based based approach [39, 170, 206, 30, 57], where researchers' expertise are more widely investigated via exploring their interactions with other academic entities. The content-based approach can be further divided into language model based approaches [8, 108, 50, 40] and topic model based approaches [147, 173, 168].

In this dissertation, one of the principle challenges (objectives) of our work is to develop an effective and efficient expert search system, to further extend and enhance previous works. We first focus on a random-walk based approach, for which we propose several new models, including a heterogeneous PageRank algorithm and a modified PageRank algorithm incorporated with temporal information. We further effectively integrate topic-based link analysis, which has demonstrated its success in web search domain, into citation network analysis. We construct our academic network as a multi-type heterogeneous network which integrates several academic factors, by which we can evaluate an author's expertise in a more complete and thus more accurate way. We then focus on a topic modeling based approach for which, we propose a joint topic modeling approach, which extends the original topic models by integrating more supportive factors; we then propose a supervised topic modeling approach by incorporating the pair-wise learning-to-rank mechanism into the generative process. Extensive experiments have been carried out on real world data sets, which demonstrate the superiority of each of our endeavors over several state-of-the-art algorithms.

1.1.2 Research Action Recommendation and Prediction

In an academic environment, we can take many 'research actions', such as writing papers, citing other papers, or collaborating with other researchers. When modeled as an academic network, these actions will be represented as 'links' between scientific

factors. Therefore, the problem of research action prediction and recommendation is equivalent to the problem of link prediction and recommendation. In this dissertation, we are particularly interested in three kinds of research action prediction: (1) publishing venue prediction/recommendation, where we aim to predict the real publishing venue of a given paper or provide recommendations as to where to submit for given a paper; (2) citation prediction/recommendation, in which we would provide a ranked list of papers (authors) for a given paper (author) to cite; (3) and coauthorship prediction/recommendation, which aims to generate a ranked list of authors for a given author to consider to collaborate with in the future.

We made contributions in the following three directions. In recommending publishing venues, we adopt the memory-based collaborative-filtering framework and provide two extensions to the original model by incorporating stylometric features into computing the similarity between pairs of papers, and differentiating the importance of different types of neighboring papers by tuning and optimizing their associated weights.

We further demonstrate the joint topic model, as mentioned in the work on modeling expertise retrieval, to be an effective method not only in evaluating researchers' expertise, but also in predicting both publishing venues and cited authors for a given author. Experiments based on real world data sets indicate that we can make improved predictions on these two tasks as compared to previous topic model based approaches which integrate fewer informative factors.

Finally, we extend the tensor factorization model maximizing MAP to model coupled higher order data in an academic environment, and demonstrated its capability in multiple types of research actions prediction and recommendation, including predicting co-authorship among authors, citation-ships among papers, citations between authors and papers, as well as publishing venue prediction.

1.2 Contributions

This dissertation aims to develop effective models for two main applications in academic network mining and analysis: modeling expertise retrieval and research actions prediction and recommendation. Although there are many differences between these two applications, a key common point is that there exist heterogeneous and mutually influenced meta data in both applications that will play an important role in determining the application's performance. For the task of ranking experts, researchers' expertise is often represented in many aspects which when integrated together can provide a more comprehensive and accurate evaluation of researchers' expertise. For example, a prestigious researcher is believed to have more publications and more citations, to publish in high-ranked conference/journal, to work often with other excellent researchers, and/or to work in good institutions; For the task of research action prediction, it is believed that people's decision is often made under the consideration of multiple aspects. For example, when deciding where to submit a finished paper, a researcher will not only consider the topic match with the potential venue, but also his/her own historical publication experience, and/or other related papers' choices. Therefore, to effectively mine and extract useful heterogeneous information and develop efficient algorithms to integrate these heterogeneous information will be of great importance. Motivated by this observation and intuition, this dissertation proposes a general framework to take into account multiple heterogeneous information, and combines information retrieval models, link analysis algorithms and machine learning techniques in a unified way to improve the performance of these applications.

Based on this framework, challenges for each application are addressed and individual specific models are proposed to solve them respectively. We summarize in Table 1.1 the models we developed, which illustrates the relationships between the models, the used data, and the utilized techniques.

Basically, for modeling expertise retrieval, we follow the main stream of how to effectively model topics into ranking process. We first integrate topical information

Table 1.1: Models developed within this work

Model	Heterogeneous Data	Techniques
Topic-drive multi-type citation network analysis	authors, papers, venues, affiliations multiple relations, temporal factor	Link Analysis
Joint topic models	authors, papers, citation, venues	Topic Models
Learning-to-rank with topic models	authors, papers author-specific features	Learning-to-Rank; Topic Models
Venue classification and recommendation	venues, papers, stylometric features	Classification; Collaborative Filtering
Joint multi-relational model	authors, papers, venues, words author-specific features, citation coauthor-ship, publication temporal factor	Matrix factorization; Tensor factorization

into link analysis by applying the Topical PageRank algorithm into citation network analysis; we then directly model authors' interests by an extended author-topic model, based upon which authors are ranked by their learned topic distributions; we finally integrate pair-wise learning-to-rank into topic modeling process, which makes it easy to incorporate other supporting features related to authors' expertise. For the task of research action recommendation, we focus on adapting and modifying the collaborative filtering approach, which is the state-of-the-art approach for recommender systems.

We then discuss the main contribution of each of our proposed models.

1. **An enhanced topic-driven multi-type citation network analysis approach for expertise ranking**

   In this work, we develop enhanced random-walk based algorithms for evaluating researchers' expertise. Particularly, we construct a multi-type heterogeneous academic network by incorporating more instructive scientific factors as compared to previous work which limit themselves to a subset of all the available informative knowledge. Particularly, we integrate author, papers,

venues, affiliations and their mutual relationships to form a multi-type academic network. To our best knowledge, we are the first to explicitly incorporate the affiliation factor. Based on this network, several investigations have been made. Firstly, we investigate the performance on different academic network structure design. We propose two versions of the academic network construction, and test via experiments the importance of introducing additional scientific factors as well as additional relationships (links); Secondly, we introduce a topic-based link analysis approach into academic network analysis to distinguish the different endorsement of academic links on different topics, and therefore, we can better model the different reputations of a researcher on different topics; Thirdly, we propose a heterogeneous PageRank algorithm to differentiate the importance and contribution of scientific factors in providing supportive evidence to the reputation of a researcher; Fourthly, we incorporate temporal information into consideration, not only by considering some temporal characteristics related with individual researchers, but also considering the influence between researchers that would be affected by time. We incorporate such temporal information into the PageRank algorithm, to better model researchers' expertise that changes over time. The importance of temporal information is evaluated for the task on predicting future award winners in the SIG community, one specific application of expertise ranking. Extensive experiments have been carried out on real world data sets on each different settings and demonstrate the superiority of our proposed approaches over several state-of-the-art algorithms.

2. **A novel learning-to-rank topic modeling approach for expertise ranking**

   In this work, we propose a supervised machine learning mechanism by distinguishing the importance with regards to their estimated expertise over pairs of authors into the topic modeling process, which to our best knowledge, results in the first work integrating pair-wise learning-to-rank into topic models.

We choose to make use of topic modeling as the basic methodology for expertise ranking in this work since it can effectively overcome the data sparsity problem as compared to other bag-of-words approaches, and well discover the underlying semantic meanings of word tokens. We incorporate the pair-wise learning-to-rank scheme into topic modeling under the hypothesis that with the guidance and support of prior knowledge, we can unearth latent topics within author profiles more accurately, and thus can better model and estimate an author's expertise. Moreover, we extend the original proposed model by incorporating additional features, each of which measures the expertise of an author from a different aspect. We apply these two models into two expert search related applications: predicting future award winners and PC members. Experiments have been conducted over real world data sets to demonstrate its effectiveness as compared with several other state-of-the-art algorithms.

3. **An enhanced collaborative-filtering model for publishing venue prediction and recommendation**

   In this work, we focus on the prediction and recommendation for one particular kind of research action: choosing a proper publishing venue to submit given a target paper.

   Before developing models, we first carry out empirical studies on determining whether writing styles can play an important role in correctly classifying publishing venues. This study is initiated by the observations that today we have many different kinds of publications covering different topics and requiring different writing formats. Even though the research on authorship identification has been well developed, no prior work has been carried out on investigating the different writing styles of publishing venues. Our work takes this first step. By approaching the task using the traditional classification method, we extract three types of writing style-based features, and conduct detailed experiments in examining the different impacts among features, classification techniques, as well as the influence of venue content, topics and genres. Experiments on

real data from real-world digital libraries demonstrate that publishing venues are indeed distinguishable by their writing styles.

We then approach the task of publishing venue prediction and recommendation by a memory-based collaborative-filtering (CF) methods, in which other neighboring papers with known venues will be utilized to predict or recommend venues for the target paper. Moreover, we propose two extensions to the original CF model: one is to incorporate stylometric features to better measure the similarity between papers. We introduce this extension based on the observations from venue classification results. For the second extension, we divide all the neighboring papers of the target paper into four categories, and differentiate the importance of each category of neighboring papers via tuning and optimizing their associated weights. Experiments based on real world data set demonstrate that our approach provide effective recommendations, and that both of the extensions can bring improved performance.

4. **An extended joint topic modeling approach for academic network analysis**

   In this work, we propose a novel probabilistic topic model that jointly models authors, documents, cited authors, and venues simultaneously in one integrated framework, as compared to previous work which embeds fewer components. We show the wide applicability of this model, as it can be adopted for three typical applications in academic network analysis: expertise ranking, cited author prediction and venue prediction. For fulfilling expertise ranking, we introduce the method on how to integrate the topic distributions computed from topic modeling results to represent the expertise of an author for a specific query. We further combine the topic modeling results with the traditional language model based approach and random-walk based approach to further improve the ranking performance. Experiments based on two real world data sets demonstrate the model to be effective, and it outperforms several state-of-the-art algorithms in all three applications.

5. **A joint multi-relational model for several recommendation tasks in academic environment**

   In this work, we target four specific recommendation tasks in the academic environment: the recommendation for author coauthorships, paper citation recommendation for authors, paper citation recommendation for papers, and publishing venue recommendation for author-paper pairs. Different from previous work which tackles each of these tasks separately while neglecting their mutual effect and connection, we propose a joint multi-relational model that can exploit the latent correlation between relations and solve several tasks in a unified way. Moreover, for better ranking purpose, we extend the work maximizing MAP (mean average precision) over one single tensor, and make it applicable to maximize MAP over multiple matrices and tensors. Experiments conducted over two real world data sets demonstrate the effectiveness of our model: 1) improved performance can be achieved with joint modeling over multiple relations; 2) our model can outperform three state-of-the art algorithms for several tasks.

1.3 Dissertation Organization

In this dissertation, we focus on research in mining and analyzing the academic network. Particularly, we address new problems and propose several novel models for two main applications in academic network analysis: modeling expertise retrieval and research actions prediction and recommendation. The remainder of the thesis is organized as follows.

Chapter 2: We review the background knowledge on expertise retrieval and recommender systems in this chapter. For the task of expertise retrieval, we first introduce its problem statement, the research development history and discuss the main research challenges in modeling expertise retrieval. Special focus has been paid on introducing the current state-of-the-art methodologies for expertise retrieval.

Evaluation methods, metrics and experimental data sets used in these related models are also discussed. For recommender systems, we concentrate on introducing the state-of-the-art approach for recommender systems design: the Collaborative Filtering (CF) approach. Both the neighborhood memory-based CF methods, and one representative model-based CF method: the Matrix Factorization method are addressed.

Chapter 3: We present our topic-driven multi-type citation network analysis approach for expert search. Network structure design is introduced first, followed by the introduction on how to integrate the topic-based link analysis into citation network analysis. We then present the heterogeneous PageRank algorithm and a modified PageRank algorithm with temporal information incorporated. Extensive experiments based on the ACM data set are presented and discussed.

Chapter 4: We present a novel learning-to-rank topic modeling approach for expert search. Model design is introduced, followed by theoretical derivations to solve the model using variational inference. We further present an extended version of this model by incorporating additional supportive features with regards to authors' expertise. We apply the model into two expert search related applications, and conduct empirical studies in evaluating models' performance as compared to other state-of-the-art algorithms.

Chapter 5: We introduce and discuss our empirical study on venue classification in which three groups of stylometric features of publishing venues are identified, with their contribution to venue classification results being examined and reported. We then apply a memory-based collaborative filtering method for venue prediction and recommendation, and propose two extensions to the original model. We report on experiments conducted on two real world data sets to demonstrate the effectiveness of our model.

Chapter 6: We present an extended joint topic model for academic network analysis in this chapter by simultaneously modeling cited author and publishing venue information, and show its applications in expert search, publishing venue prediction and cited author prediction. We report and discuss the experiments results over two real world data sets.

Chapter 7: We propose a joint multi-relational model that can exploit the latent correlation between relations and solve several tasks in a unified way. This model is especially designed for four recommendation tasks: the author-author coauthorship recommendation, author-paper citation recommendation, paper-paper citation recommendation and paper-publishing venue recommendation. Moreover, for better ranking purpose, we extend the work maximizing MAP over one single tensor, and make it applicable to maximize MAP over multiple matrices and tensors. Experiments conducted over two real world data sets demonstrate the effectiveness of our model.

Chapter 8: We summarize the dissertation in this chapter and provide some directions to be explored in future work.

Chapter 2

Background

2.1 Expertise Retrieval: Introduction

As 'people' have become one important source of information, there has been increasing demand and interest for people to find each other as a source of inquiring questions, seeking help, making friends, or building social communities. Finding and ranking people with regard to their estimated expertise over a topic has a wide range of applications in people's lives, as it can help to facilitate finding experts in research or industry organizations, facilitate making decisions on job recruitment or promotions and more.

In this chapter, we give an overall summary of the task and current state-of-the-art approaches for expertise retrieval.

Problem Identification: What is Expertise Retrieval?

Expertise retrieval addresses the task of identifying and ranking a list of people with their estimated relevant expertise for a given query. In a typical expert finding process, given a query, the participating system will return in response a ranked list of candidate persons with respect to their predicted expertise. Figure 2.1 illustrates

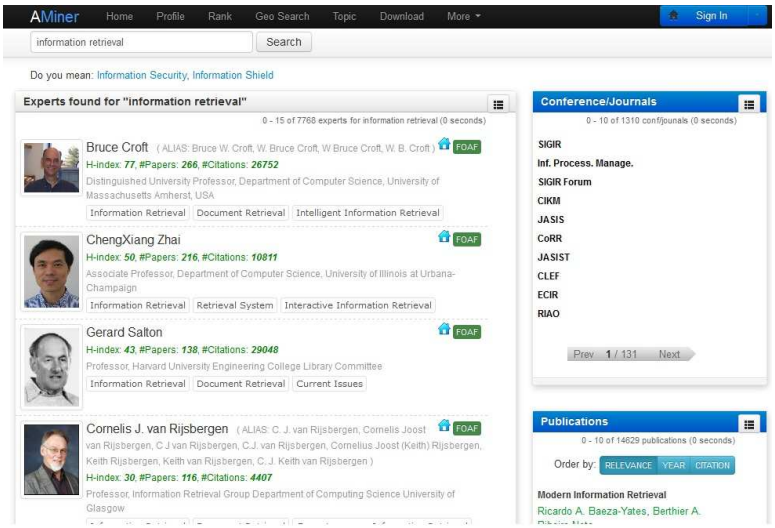

Figure 2.1: One Sample for Expertise Retrieval (results from ArnetMiner expert search engine)

one such example: the returned ranking results as we issue the query 'information retrieval' to a typical expert search engine (ArnetMiner[1]) indicating that we are looking for the experts on 'information retrieval'. As shown in the figure, several well-known researchers have been identified with their photos and basic information provided.

Why to retrieve people's expertise?

Expertise retrieval emerges as an important research topic as the result of the vast development of world wide web and information retrieval technology. People are currently regarded as one important source of information. Identifying experts is beneficial for multiple reasons.

On one hand, expertise retrieval is an effective supplement to traditional document-centric retrieval. Since not all the information can be possibly documented, much important information can only be transferred through experience and informal conversations, and therefore many information-gathering tasks would be better handled

[1]http://arnetminer.org/

by finding a referral to a human expert rather than by simply interacting with online documentary information sources. Besides, as compared to spending lots of time and effort in accumulating experience and finding a piece of information from the very beginning, individual users will sometimes find it more convenient and effective to directly find an expert and leverage on their expertise to tackle problems. These advantages make finding experts a better way to solve problems than searching documents in some occasions.

On the other hand, the wide range of real-world applications of expertise retrieval stimulate research work in this area. Here we illustrate several such examples.

- **Finding experts in organizations**

  Knowledge in an organization is contained in the skill, experience and expertise of its people [25]. These organizations can either be industry enterprises or academic institutes. Finding the right person in the organization with appropriate skills and knowledge is often crucial to the success of problem solving or projects being undertaken [8]. In a research organization, for example, people often need to find specialists or professors to answer questions with whom to collaborate; in an enterprise, the organizers usually hope to assign tasks to those who have obtained enough skills and experience to fulfill that task. Identifying the appropriate person is of great importance to these organizations. Much of the recent work in expert finding has been to address this need in such organizations.

- **Finding experts in online social media communities**

  Web-based communities have become important places for people to seek and share expertise [203]. Typical communities include the help-seeking question-answering systems [105], online discussion forums [204], Blogs [8] and Twitter [175]. Identifying the most influential experts in such communities can help us choose the best one to answer our questions, or to follow those learned people for updated information over certain topics.

- **Facilitating automatic reviewer assignment**

Peer-review is an important practice for quality control in scholarly publications [15]. It is the duty of journal editors, funding program managers (e.g., at NSF), conference program chairs and research councils to assign submitted papers to reviewers with appropriate knowledge and experience. Traditionally, this complicated job is manually handled by a few people, which turns out to be labor-intensive. An expert finding system which can automatically extract and identify the expertise of each reviewer can largely relieve the burden of journal/conference editors.

- **Facilitating job recruitment, promotion and award assignment**

  In modern society, the most common practice for people finding a job is via submitting personal resumes and other supporting documents. Employers need to find out proper candidates with related skills and work experience by reviewing hundreds of thousands of such documents. Developing automated job recruitment system which can evaluate applicants' expertise and identify proper candidates can greatly relieve the work of employers and improve both efficiency and accuracy. The same process can benefit people who are responsible for making decisions on who needs a promotion. Expert finding can also help in facilitating the identification of nominees and award winners for scientific awards.

Research Development Outline

People have had such information needs for experts even before the invention of computers. With the development of computer and information technology, people started to concentrate on developing automatic computer-supported expert finding systems. Ever since then, the development history of the research on expertise retrieval can be divided into three periods: 1) historical work; 2) focused attention because of the TREC Enterprise track, and 3) modern work.

Before the TREC Enterprise Track: In the initial stage, the traditional

approach for expertise retrieval relies on creating, organizing and controlling expertise information in a database [134], which is often constructed by individual users manually inputting their personal information and using keywords to describe their own expertise. This method is labor-intensive, and is inconvenient for in-time update. With the development on information knowledge, more efforts have been taken into developing automatic expert finding systems which utilize modern information retrieval technology. However, during this period of time, different systems focus on different specific document types, for example, organizational technical reports, software source codes, emails, and more. Without standard working data collections and queries, the proposed algorithms and approaches are hard to be evaluated and compared. Besides, no unified models have been provided that can tackle heterogeneous data collections.

The TREC Enterprise Track: from 2005 to 2008, the task on expertise retrieval was launched as part of the Enterprise Track in Text REtrieval Conference (TREC) which provided two standard data collections with queries and labeled ground truth and therefore generated a common platform for researchers to empirically evaluate their proposed methods and techniques for expert finding. Ever since then, expertise retrieval has received a substantial boost in attention from information retrieval, data mining and machine learning communities.

After the TREC Enterprise Track: after the advent of TREC Enterprise Track, much more work has been developed and evaluated based upon other data test beds rather than those provided by TREC Track. These test beds more focused on scientific literature within academic environments. Typical such data sets include the UvT data collections, DBLP, CiteSeer, ACM and ArnetMiner data sets, for which we will introduce in more detail in the following sections. Compared to the models proposed for expertise retrieval task in TREC Track, which emphasize their research on mining candidate-document pairs associations and extracting the expertise mainly from related documents, more research efforts have been made on mining other supporting expertise, i.e., the social interactions among expert candidates. Our work developed in this dissertation falls into this group of research.

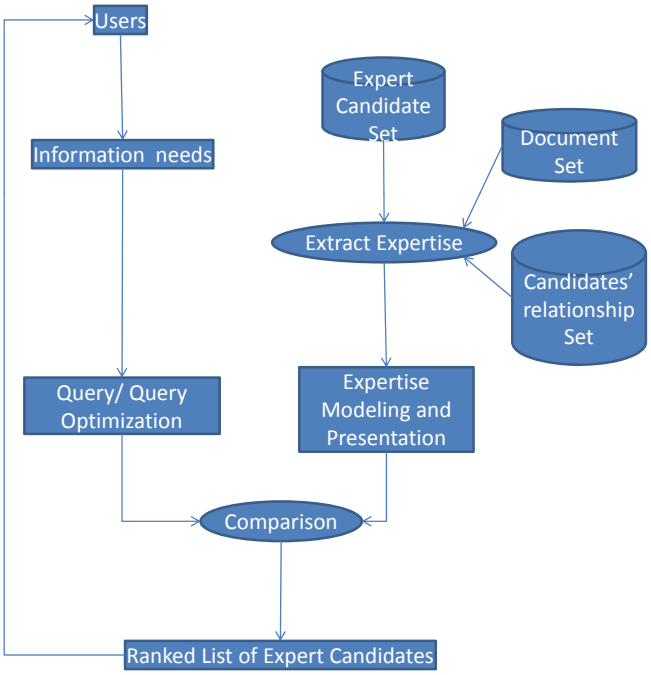

Figure 2.2: Expert Finding System Architecture

2.2 Expertise Retrieval Systems

2.2.1 A Typical System Framework

In an automated expert finding system, users can input as queries the particular kind of expert they are seeking, and the system will return a list of experts in the order of their relevancy to the query (topic). Figure 2.2 shows the typical framework of an expert finding system.

Two components are of special importance to an expert finding system: the process of how to collect and well represent the expertise of a candidate; and the process of how to evaluate the relevancy of candidates' expertise to the query. Generally speaking, two types of expertise evidence have been explored in previous research work: the supporting documents associated with expert candidates and the social interactions among expert candidates. Several models and algorithms have been

proposed with the goal to accurately and efficiently estimate the relevance of the identified evidence of expertise to the query, which we will discuss in the following sections.

2.2.2 Expert Search Engine

With the rapid development of world wide web and information retrieval technology, document-centric search engines have shown great success and re-shaped people's daily life. Under this background, the ever increasing information needs for people search stimulates the emergence and development of search engines particularly designed for expertise retrieval. ArnetMiner and Microsoft Academic Search are two representatives of them, both of which focus on the academic domain, and their main functionality is to provide ranking for academic related entities, i.e. authors, papers, conferences and organization, as well as mine and analyze their mutual interactions.

ArnetMiner system[2] is developed by Tsinghua University of China, which aims to 'provide comprehensive search and mining services for researcher social networks', particularly in the computer science domain. The main search and analysis functions in ArnetMiner include profile search, expert finding, conference analysis, course search, sub-graph search, topic browser, academic ranks, and user management. They provide visualization tools to represent their ranking or analysis results for better users' experience. Figure 2.1 shows one example for their expert finding results for query 'information retrieval'. Microsoft Academic Search[3] developed by Microsoft Research Asia is another well-known public expert finding system. Compared to ArnetMiner which focuses on computer science domain, Microsoft Academic Search supports expertise retrieval for 15 different research disciplines, and further divides each discipline into finer-grained sub-disciplines. In computer science domain, for example, they identify 23 sub-categories which cover the main research topics for computer science. Similar to ArnetMiner, Microsoft Academic

[2]http://arnetminer.org/

[3]http://academic.research.microsoft.com/

Figure 2.3: Microsoft Academic Search Engine search example: Coauthor-graph

Search also visualizes the results. One prominent instance is that they provide graph visualization over several types of relationships for each researcher, for example, his/her co-authors or citations. Figure 2.3 shows such an example for Prof. Brian D. Davison.

2.3 Main Challenges in Expert Search

Finding an expert is a non-trivial task and it brings new challenges to those associated with traditional document retrieval. We list here several key challenges.

Identify and extract proper sources of evidence to represent the expertise of experts

Since 'expertise' is an abstract concept without concrete definition, one of the

most challenging components for expert finding is to identify proper sources to represent the 'expertise' of expert candidates. Generally speaking, two main categories of expertise have been identified: 1) the documents associated with the expert candidates, which include their published papers, project reports, emails, posts, blogs, tweets, product reviews or any other content. The principal intuition behind making use of such source of expertise is that if an expert can generate more documents that are highly relevant to a query, then this expert would also have a higher probability to be an expert for that query. 2) the interaction or relationships with other candidates or other types of entities. For example, there is a greater probability for a researcher to have high expertise for a query topic if he can collaborate with many other experts on this topic or be cited more often by other experts. These two types of expertise evidence can be combined.

Estimate the relevancy of an expert candidate to the query

Given the identified expertise evidence, the key component of expert finding is to develop proper approaches to mine information from such evidences and estimate the relevancy of the expert candidate to the query. This is the part that most research endeavors emphasize. Multiple approaches have been provided to solve this task, which we will introduce in more detail in the following sections.

Name Disambiguation

People's names are often ambiguous: they can be written in various formats, for example, some people put their given name first while others put the family name first. There are many abbreviations, multilingual issues, and that some identical names belong to different people. These will have a large effect on the process of accurately extracting the associated documents with a specific expert or build his connections with others. Name disambiguation is a separate research topic in IR community, however, is not the research focus in this dissertation.

Heterogeneous data integration

As pointed out, there are many kinds of evidence to represent an expert's expertise, and these evidences often come from heterogeneous sources, whose importance in determining the expertise of an expert may vary. To find approaches to effectively integrate these heterogeneous sources is a challenging and interesting research task.

Expertise evolution

People's expertise will change over time. A new researcher in a specific research domain will gradually accumulate his reputation and become a respected scientist in this area in the future. Examining the pattern of growth is interesting and may offer help in predicting some events, such as scientific award assignments.

Evaluation Problem

In order to evaluate the performance of proposed algorithms, we need to have standard data collections, queries and labeled ground truth. Before the advent of TREC Enterprise Track, we lacked such information for quite a long time. The data collections provided by TREC Enterprise Track, however, are very limited to the data sources within the W3C or CISRO organizations, and have much noise. With the rapid development of online digital libraries, scientific literatures provide us plenty of excellent data sources for evaluating people's expertise, especially the research scientists. However, we still lack proper queries, and the ground truths often need to be manually labeled.

In the following sections, we will first introduce some standard or widely used data collections, as well as the data collections we used in this dissertation, and then focus on introducing the main existing approaches for expertise retrieval.

2.4 Experimental Data Collections

To evaluate the performance over different expertise retrieval algorithms, standard test data collections as well as queries and their associated ground truths are of great importance. In this section, we briefly review several such data collections developed in previous research, two of which are provided by the Enterprise Track of TREC,

focusing on the enterprise domain, and three of which focus on the academic domain.

The W3C Collection[4]: the W3C data collection is the first standard data collection provided by the Enterprise Track in TREC, whose appearance has initiated the rapid development in expert finding research in the IR community. It was used as the working data set for the Enterprise Track in 2005 and 2006. The collection is composed of the internal documentation of the World Wide Web Consortium (W3C) crawled in June 2004. It contains 331,037 documents from the following six sub-collections: email discussion forum (lists), source code documentation (dev), web pages (www), wiki (esw), miscellaneous (other), and personal homepages (people). In total, 1,092 expert candidates represented by their full names and email addresses have been identified, In 2005, 50 queries have been provided using the titles of the working groups in W3C, and that all members of each group are considered as the relevant experts for that query. In 2006, 49 queries have been provided by the TREC participants, and their associated ground truths also manually generated by those participants based on assessing the supporting documents of each candidate.

The CERC Collection[5]: the CSIRO Enterprise Research Collection (abbreviated as the CERC data collection) was used at the Enterprise Track of TREC in 2007 and 2008. It is the result of crawling the publicly available pages on the official web set of CSIRO, which contains 370,715 documents. There is no explicit expert candidates list provided but a list of email addresses used by CSIRO employees. In total, 127 queries and their associated relevant experts list were developed by several science communicators invited by the TREC organizers.

The UvT Expert Collection[6]: the UvT collection concentrates on the academic domain. It was developed by using the public data about employees of Tilburg University in Netherlands. The collection contains for each expert candidate a page in both English and Dutch which includes the expert's contact information, research

[4]http://research.microsoft.com/en-us/um/people/nickcr/w3c-summary.html
[5]http://es.csiro.au/cerc
[6]http://ilk.uvt.nl/uvt-expert-collection

Table 2.1: Data Collections Summary (1)

Data Collection	CandidatesNo.	DocumentsNo	QueriesNo	Total Relevant Judgments
W3C	1092	331037	99	9860
CERC	3500	370715	127	2862
UvT	1168	36699	1491	4318
DBLP + Google Scholar	574369	953774	17	244
INDURE	12535	NA	100	6482

and course description and publication records. There are 1,880 expert candidates in total. 981 queries have been provided, and their associated ground truths are generated by the university's employees themselves.

The DBLP Collection [40]: the DBLP data collection is a subset of the DBLP database which contains records of 953,774 papers. It is often used in combination with data from other digital libraries to retrieve detailed information of each paper. For example, in the research work carried out by H. Deng [40], they further incorporate for each paper in the DBLP records its abstract by downloading from Google Scholar. Assessments for expert candidates were conducted manually, and a four-grade score is assigned to each candidate indicating his/her different levels of expertise. A. Hogan et al. used in their work [74] DBLP data records combined with CiteSeer data set to retrieve papers' abstracts.

The INDURE Expert Collection [50]: the Indiana Database of University Research Expertise (INDURE for short) is a collection mainly containing data for faculty in PURDUE university. The data information comes from four sources: (1) a profile filled by each faculty member indicating his or her main research areas; (2) faculty homepages; (3) faculty's NSF funded projects descriptions; and (4) faculty's own publications and their PhD students' dissertations.

Table 2.1 shows a summary over these data collections mentioned above.

Even though these data collections have been used in previous research, they have some limitations. The two data collections provided by TREC Enterprise Track: the W3C and CERC data collection are very limited to the data sources within the W3C or CISRO organizations, and have much noise. The UvT and INDURE data collection are created from one single data source within one organization, and therefore are not easy to be generalized. DBLP data collection contains no citation information, and has to be integrated with other data sets, such as Google Scholar or CiteSeer. Such an integration process is prone to introduce additional data noise. To avoid such limitations, we adopt the following three data collections throughout the work we present in this dissertation, including (1) the ACM data set; (2) the CiteSeer data set; and (3) the ArnetMiner data set. We choose to use these three data sets as our working data sets because of the following three reasons. First of all, they are academic-centric data sets, and are therefore appropriate for our research in mining academic networks. Secondly, they are more general, consisting of authors and papers from different organizations. Thirdly, they are in plain text or XML format and are self-contained. We can not only retrieve from these data sets the content-based information of papers, such as their titles, abstracts, authors and publishing venues, but also the social interactions among those academic factors, such as the co-authorship among authors, and citations among papers. We do not need to further integrate them with other supporting data collections.

We introduce the three data sets as follows:

- **ACM data set**: The ACM data set is composed of papers crawled from the ACM digital library[7]. For each paper, we crawled one descriptive web page for it; we extracted and recorded the information of each paper's title, abstract, publishing venue, authors, affiliation of each author, and citation references. Simple statistics shows that there are 172,890 distinct web pages within the crawled data set that appear to have both title and abstract information. These papers are published between 1951 to 2009.

[7]http://dl.acm.org/

Name ambiguity is a common problem in representing author names and venue names. While not eliminating the problem, to minimize ambiguity in the use of author names, we concatenate the authors' first and last name, and remove the middle name (if present). We then use exact match to merge candidate author names. Finally, we obtain 170,897 distinct authors. Due to possible venue name ambiguity, we first convert all upper-case characters into lower-case, and remove all non-alphabetical symbols. We further removed all digits as well as the ordinal numbers, such as the 1st, the 2nd, and applied Jaccard similarity match to merge duplicate venue names. We finally obtained 2,197 distinct venues.

In extracting citation references, the title is the representative of each paper, and we only consider those cited papers for which we also crawled the corresponding web page for it.

- **CiteSeer data set**: The CiteSeer data set is distributed by the 2011 HCIR challenge workshop[8]. The whole data corpus is divided into two parts: the meta-data about a paper, such as its title, publishing venue, publishing year, abstract, and information about citation references are kept in XML format; and the full content of that paper is in plain text format. From the distributed data corpus, we collected 510,231 distinct scientific papers published between 1934 and 2010. After applying the same working process as we did for ACM data set to merge ambiguous author names and venue names, we finally obtained 479,805 authors and 65,441 venues.

- **ArnetMiner data set**: The ArnetMiner data set we utilized in this dissertation is the data set 'DBLP-Citation-network V5' provided by Tsinghua University for their ArnetMiner academic search engine [170]. This data set is the crawling result from the ArnetMiner search engine on Feb 21st, 2011 and further combined with the citation information from ACM. The original

[8]http://hcir.info/hcir-2011

Table 2.2: Data Collections Summary (2)

	ACM	ArnetMiner	CiteSeer
authorNo.	170,897	798,385	479,805
paperNo.	172,890	1,558,415	510,230
venueNo.	2,197	6,010	65,441
year range	[1951, 2009]	[1936, 2011]	[1934, 2010]

data set is reported to have 1,572,277 papers and to include 2,084,019 citation-relationships. After carrying out the same data processing method as we did for the ACM data set, we find 1,558,415 papers, 795,385 authors and 6,010 venues. Papers in this data set are published between 1936 to 2011.

Table 2.2 shows a summary on the ACM, CiteSeer and ArnetMiner data sets.

Experiments reported in this dissertation are conducted on one or two of these three data sets. For better evaluation purpose, further data preprocessing may be carried out over these original data sets for which we will introduce in more detail in the following chapters respectively.

2.5 Evaluation Metrics

To evaluate the performance of different approaches, several metrics have been adopted. The most representative ones include: 1) P@k; 2) MAP; 3) MRR; and 4) NDCG@k.

MRR, MAP, and P@k are suitable metrics for binary relevance ranking performance evaluation, where entities (documents, people or other entities) are either relevant (relevance=1) to a query or non-relevant to a query (relevance=0). MRR works for the situation when there is only one relevant entity in the data corpus. NDCG@K works for multiple levels of relevance, both the relevance scores for retrieved entities and their ranking positions are important to the final ranking performance.

- **P@k** (abbreviated for *Precision at rank k*): $P@k$ measures the fraction of the top-k retrieved entities (either documents or authors) that are relevant for a given query, which can represented as:

$$P@k = \frac{\#(relevant\ entities\ in\ top\ k\ results)}{k} \tag{2.1}$$

- **MAP** (abbreviated for *Mean Average Precision*): *Average Precision* (**AP**) emphasizes returning more relevant documents earlier (to rank them at higher positions). Given one single query, AP is defined to be the average of the $P@k$ values for all relevant entities:

$$AP = \frac{\sum_{k=1}^{K}(P@k \times rel(k))}{R} \tag{2.2}$$

  where, k is the rank, K is the total number of entities retrieved, and R is the total number of entities that are relevant to the given query. $rel(k)$ is a binary indicator function satisfying $rel(k) = 1$ if the document at rank k is relevant to the query, or 0 otherwise. **MAP** is the mean value of the **AP** computed across all queries. The computing process for MAP can be described as: 1) first mark the position of all relevant entities $k_1, k_2, \ldots, k_R$, where R is the total number of all relevant entities; 2) compute the $P@k$ scores at all places in $k_1, k_2, \ldots, k_R$; 3) average over $P@k$; 4) average across all queries.

- **MRR** (abbreviated for *Mean Reciprocal Rank*): MRR measures the ranking performance when there is only one relevant entity for any given query in the ranking system. Suppose for a given query q, the only relevant entity is ranked at position k, then the *Reciprocal rank* for query q is $\frac{1}{k}$. MRR is then the mean reciprocal rank across all queries.

- **NDCG@k** (abbreviated for Normalized Discounted Cumulative Gain at rank k) is a traditional metric for a ranking system when there are multiple levels of relevance for entities over queries. The computation process can be described as follows.

  Suppose we have a collection of n queries denoted as $\mathcal{Q} = q^1, \ldots, q^n$. For each query q^k, we have a collection of m_k relevant documents (assume the entity

is document here) $\mathcal{D} = d_i^k, i = 1, \ldots, m_k$, whose relevance to q^k is given by a vector $r^k = (r_1^k, \ldots, r_{mk}^k \in \mathbb{Z}^{mk})$. Suppose we have a ranking function denoted as $F(d, q)$ that outputs a computed relevance score in a real number for every document-query pair (d, q), and suppose document d_i^k is ranked as position j_i^k within the collection set $\mathcal{D}^k$ for query q^k, then the NDCG value for ranking function $F(d, q)$ can be computed as:

$$\mathcal{L}(Q, F) = \frac{1}{n} \sum_{k=1}^{n} \frac{1}{Z_k} \sum_{i=1}^{mk} \frac{2^{r_i^k} - 1}{\log(1 + j_i^k)} \tag{2.3}$$

where Z_k is the normalization factor, and it is computed as the DCG score when all documents are ideally ranked in descending order of their relevance scores. NDCG@k is the NDCG score for the fraction of the top-k returned documents.

2.6 Existing Approaches

In this section, we briefly review the main approaches developed for expertise retrieval. We divide the approaches into two main categories: **the Content-based approach**, in which the expertise evidence of candidates is mainly extracted from the textual documents associated with them, and that the relevance of a candidate expert to a query is computed via the relevance of those supporting documents to a query; and the **Graph-based approach**, in which candidates' expertise can be represented via their social interactions with other academic entities. The content-based approach can be further divided into generative probabilistic models, discriminative probabilistic models, voting models, and topic modeling based models; Categorized like this, however, it is worth mentioning that many existing models actually combine the content-based and graph-based approaches, and benefit from the advantages of both of them.

2.6.1 Generative Probabilistic Models

Mathematically interpreted, the task of expert finding can be represented as computing the probability of an expert candidate e being an expert given the query topic q, i.e., $P(e|q)$, which based on Bayes' Theorem, can be factored as:

$$P(e|q) = \frac{P(q|e)P(e)}{Pq} \approx P(q|e)P(e) \tag{2.4}$$

Equation 2.4 successfully transfers the computation of $P(e|q)$ into the approximated $P(q|e)$, which represents the fundamental idea of generative probabilistic model, i.e., the relevance of a given query to a candidate expert can be estimated as the probabilistic likelihood that the query topic is generated by the given candidate. Several generative probabilistic models have been developed. Two of them proposed by Balog et al. [8] are the most representative ones, as they often serve as the comparison baseline algorithms in subsequent research efforts.

In [8], Balog et al. developed two different versions of computing the likelihood $P(q|e)$, both of which are essentially based upon the standard statistical language model. In the first version which is referred as the **Candidate Model**, all documents related to a candidate can be utilized to generate a textual representation of this expert's expertise, and that the relevance of the query to the candidate can be computed via estimating the relevance of the query to the textual representation using the traditional language models. This process can be presented as:

$$P(q|\theta_e) = \prod_{t \in q} (1-\lambda)(\sum_{d} P(t|d)P(d|e)) + \lambda P(t)^{n(t,q)} \tag{2.5}$$

where λ is the parameter for smoothing, θ_e denotes the candidate language model for candidate e and $n(t,q)$ is the term frequency of term t in query q.

In the second version which is referred to as the **Document Model**, all documents relevant to the query are retrieved and estimated first, and then the expert candidates that are associated with these relevant documents will be regarded as the experts for the given query. Under this scheme, the probability of $P(q|e)$ can be computed as:

$$P(q|e) = \sum_{d} \prod_{t \in q} P(t|\theta_d)^{n(t,q)} P(d|e) \tag{2.6}$$

where θ_d indicates the document-centric language model.

Both of the candidate model and document model are built upon the assumption that the query prior $P(q)$ and candidate prior $P(e)$ are uniform and therefore can be ignored, and that the appearance of terms and candidates are independent given a document.

A similar model to the *Candidate Model* was proposed by Fang and Zhai [49], and Petkova and Croft [134] provided an extension of the candidate model by combining multiple sources of document collections. H. Deng et al. made two important improvements over the *Document Model* [40, 41]. In [40], they proposed a weighted language model that takes into consideration not only the relevance of supporting documents to the query $P(q|d)$ but also the importance of individual document, i.e., the prior probability $P(d)$, which is often regarded as uniform and therefore ignored in previous research. In [41], Deng et al. investigated a new smoothing method by using community context instead of the whole collection to enhance the *Document Model*.

2.6.2 Voting Models

The Voting model [108] is inspired by the data fusion techniques which attempt to effectively combine supporting evidences from different sources. Using data fusion for expert finding, as introduced in the voting model, each document associated with the candidate expert and relevant to the given query will represent one 'vote' for determining the relevance of the document to the query. However, the weight on the votes can be varied; for example, it can be a binary vote, the reciprocal rank of the document for the query, or the specific relevancy score of the document to the query. In [108], the authors identified 12 different data fusion techniques in representing the 'weight' of such votes, and they further enhanced the original model by utilizing query expansion [109]. Experiments indicate that the voting model can retrieve competitive results as the probabilistic generative model proposed in [8].

2.6.3 Discriminative Probabilistic Models

As a counterpart to the generative probabilistic model, the discriminative probabilistic model directly estimates the probability that a candidate is an expert for a given query. One representative work in applying the discriminative probabilistic model into solving the task on expert finding is the work conducted by Fang and Zhai et. al [50], in which they cast the expert finding task into a classification problem where the relevant query-candidate pairs are treated as positive samples and the irrelevant query-candidate pairs are regarded as negative samples. Under this scheme, the probability likelihood over the training data set can be presented as:

$$L = \prod_{m}^{M} \prod_{k}^{K} P_\theta(r = 1|e_k, q_m)^{r_{mk}} P_\theta(r = 0|e_k, q_m)^{(1-r_{mk})} \tag{2.7}$$

where θ indicates the set of model parameters.

The authors proposed two specific models to measure the relevancy probability for expert-query pair: $P_\theta(r = 1|e_k, q_m)$, both of which take the associated documents as the connecting bridge. In the 'Arithmetic Mean Discriminative' model, this probability can be computed as: $P_\theta(r = 1|e, q) = \sum_{t=1}^{n} P(r_1 = 1|q, d_t)P(r_2 = 1|e, d_t)P(d_t)$, and in the 'Geometric Mean Discriminative' model, the probability can be computed as $P_\theta(r = 1|e, q) = \prod_{t=1}^{n} P(r_1 = 1|q, d_t)P(r_2 = 1|e, d_t)P(d_t)$. Here, r_1 and r_2 are binary indicators, representing the relevance of document d_t to query q, and the relevance of candidate expert e to document d_t respectively. To further compute the relevancy for query-document pairs and expert-document pairs, a group of features can be directly incorporated; for example, the probability over query-document pair can be computed as $P(r_1 = 1|q, d_t) = \sigma(\sum_{i=1}^{N_f} \alpha_i f_i(q, d_t))$, where f_i are the query-document related features, and $\sigma(\cdot)$ is the sigmoid function.

The ability to directly incorporate features is the most prominent property and advantage of the discriminative probabilistic model as compared to the generative probabilistic model.

Two other representative works in applying discriminative probabilistic models to solve the task of expert finding include Moreira et al. [126] and Macdonald

and Ounis [110]. Both work directly utilize several existing learning-to-rank [104] algorithms into expert finding, where learning-to-rank is a prominent research area in recent IR research, and has shown great success for documents' ad hoc retrieval. In the work [126], seven learning-to-rank algorithms: AdaRank [186], Coordinate Ascent [121], RankNet [23], RankBoost [55], Additive Groves [164], SVMmap [200] and RankSVM [82] have been used over a set of self-identified features; In work [110], they applied two learning-to-rank algorithms: AdaRank [186] and Automatic Feature Selection (AFS) [122] on the features derived from their voting models.

2.6.4 Topic-Modeling-based Models

Topic modeling has emerged as a popular unsupervised learning technique for content representation in large document collections. This kind of generative model was first envisioned for pure contextual analysis while ignoring the linkage structure among text data. Representative models of this type of analysis (e.g., the LDA model [18] and pLSA model [72]) exploit the co-occurrence patterns of words in documents and unearth the semantically meaningful clusters of words (as topics). Researchers have since added extensions to model authors' interests in their proposed author-topic model [147], and therefore makes the topic modeling available for expert finding. Several following models have been proposed to further overcome the limitations of the author-topic model [147] and improve the expert ranking performance, including the author-conference-topic model [168], citation-author-topic model [173], author-conference topic-connection model [180], and context sensitive topic models [88].

2.6.5 Graph-based Models

The principal idea of the content-based approach is to evaluate the relationship between the expert candidate and the query topic via supporting documents; however, candidates' expertise can also be represented via their interaction between other candidates or other types of entities. This idea stimulates another direction of research which centers on generating an expertise graph (or called an expert social

network) in which the expert candidates or other types of entities are represented as nodes and their interactions (relationships) as links, and applying graph-based algorithms or link analysis approaches into expert finding. We refer to this group of models as graph-based models.

One intuitive link analysis approach is to utilize simple statistical measurement which indicates estimating the expertise of candidates by simply counting the their in-degree, out-degrees, or other measurements such as centrality, closeness and betweenness. Typical research work in this category include the work of Zhang et al. [?] which identified the experts in a Java Forum by counting the number of others users they replied, and the work of Kolla and Vechtomova [96] which builds an expertise graph from W3C email lists, and ranks the experts by either the number of their in-coming emails or out-going emails.

PageRank [132] is a popular link analysis approach which has demonstrated its great success in World Wide Web in determining the authority of a web page. The fundamental idea behind PageRank is that a web page will have a higher authority if it is pointed to by more other web pages with high authority. This basic assumption is appropriate to be applied into determining the expertise of people, since people tend to have higher expertise (authority) if they have more valued interactions with others, for example, a researcher more often collaborating with or being cited by other researchers with high expertise. Due to this similarity, a number of PageRank-like algorithms have been proposed in expert finding. Typical work includes [30, 57, 19].

HITS [93] is another popular link analysis algorithm widely used in the World Wide Web. It assigns two scores to each web page: a hub score and an authority score, which can be iteratively updated by looking at the hub and authority scores of other web pages pointing to and being pointed from the current web page. Inspired by this algorithm, a group of HITS-like models have been proposed including [25, 179] for expertise retrieval.

Pure graph-based models which ignore the relevancy derived from associated documents are often query-independent models, and the background expertise graph

they rely on only includes one type of node: the expert candidate nodes. Several models have been proposed providing extensions by building a bipartite graph which incorporates both expert candidate nodes and the supporting nodes. Two representative models following this line include the work proposed by Serdyukov et al. [153] which provided three versions of random-walk algorithms: a finite, infinite and a specialized parameter-free absorbing models over a bipartite graph consisting of expert candidate and top retrieved documents. Another example is Zhou et al. [206] who proposed a query-specific co-ranking algorithm over bipartite graphs to integrate an author-coauthor relationship network and the paper citation network. PopRank [131] provides further development by integrating one more factor: the conferences and journal factor in addition to authors and papers.

Hong et al. [39] proposed a new graph-based model which introduced the graph regularization techniques into expert ranking based upon the assumption that similar documents in content are likely to have similar relevance scores with respect to a query. In their following work [41], they defined community-sensitive authorities for authors, and proposed a query-sensitive AuthorRank algorithm to model author's authority based on a co-authorship network.

2.7 Recommender Systems: Introduction

Recommender systems [145] has increasingly demonstrated its success in online personalized businesses, by which various commercial items including retailing products, movies, books, musics, advertisements, etc. can be suggested to individual users to suit their tastes. This largely stimulates the rapid development of e-commerce webshops like Amazon, eBay and Netflix. Recommender systems then gradually became a promising technology in social media and social networking applications, where it can provide tag recommendations in social sharing or bookmarking systems (like Flickr[9] or del.icio.us[10]), and generate link predictions in social media networks, for example, suggesting the 'friends you may also want to know

[9]http://flickr.com
[10]http://del.icio.us

or follow or connect' on Facebook, Twitter and LinkedIn, or the blogs/tweets/news articles that you may feel interested on Blog, Twitter or online news websites. Recommender systems has become an indispensable technology that dramatically affects people's daily life.

We also need recommendation in academic environments. There are hundreds of thousands researchers in the academic community, producing millions of research papers to date, and the number of new papers has kept on increasing with time. Statistics have shown that based on the DBLP scientific data set, computer scientists published 3 times more papers in 2010 than 2000. On the other hand, the rapid development of online digital libraries have made these published papers as well as their associated information, such as their authors, publishing venues much easier to access. These often result in the information overload problem for individual researchers when they want to identify the proper papers to cite, or choose a proper conference/journal to submit. Recommender systems can offer help in solving these problems, and therefore is another research focus in this dissertation.

Collaborative Filtering approach (CF for short) [61] is the current widely adopted and state-of-the-art technique in recommender systems whose fundamental idea is to establish the connections between users and other entities via analyzing their historical interactions. To take the most widely used application scenario for CF, the user-item-rating recommendation as an example, the underlying assumption is that an individual user will prefer the items which other similar users prefer, or the items that are more similar to those items that the user has originally rated (liked). CF can be further divided into neighborhood memory-based approaches and model-based approaches due to the different mechanism on how to analyze the historical interaction or how to establish the current connection.

Due to the wide range of applications of recommender systems, and the varied specific algorithms developed to tackle the problems in each individual application, we concentrate in this chapter on the introduction of the fundamental technique for recommender systems: the CF method. We will introduce in detail the neighborhood memory-based models, and we choose to introduce the current state-of-the-art model-based approach, the matrix factorization model[97].

2.8 Collaborative Filtering

2.8.1 Neighborhood Memory based CF

Neighborhood memory-based CF is widely used in the user-item-rating prediction scenario due to its simple intuition and easy implementation. It can be further categorized into user-based CF and item-based CF.

User-based CF

In the user-based CF method, predictions are made by first identifying other users who are similar to the target user (similar in user profiles or historical rating patterns) and then takes a weighted combination of their ratings to the target item. More formally speaking, let us suppose a be the target user and i be the item which is not rated by a yet, then the predicted rating of a to i: p_{ai} can be computed as:

$$P_{ai} = \overline{r}_a + \frac{\sum_{u=1}^{N}(r_{ui} - \overline{r}_u)\omega_{au}}{\sum_{u=1}^{N}\omega_{au}} \tag{2.8}$$

where r_{ui} is the real rating of user u to item i. $\overline{r}_a$ and $\overline{r}_u$ are the mean ratings of user a and u respectively, and ω_{au} is the similarity score between user a and u. N here indicates the number of users that are similar to the target user u.

Item-based CF

In the item-based CF method, predictions are made by first finding similar items to the target item and then calculating a weighted combination of their ratings by the target user u. It can be formally represented as:

$$P_{ai} = \overline{r}_i + \frac{\sum_{k=1}^{M}(r_{ak} - \overline{r}_k)\omega_{ik}}{\sum_{k=1}^{M}\omega_{ik}} \tag{2.9}$$

where $\overline{r}_i$ and $\overline{r}_k$ are now the mean rating of item i of item k based on all previous ratings. ω_{ik} is the similarity weight of item i and k. M indicates the number of all similar items to the target item.

Compute the Similarity

As we can see from both the user-based and item-based CF models, one key function is to compute the similarity between either users or items. Traditionally, there are two widely used similarity computation algorithms; one is based on the Pearson Correlation [143] score and the other is based on Cosine Similarity.

Pearson Correlation Score

Pearson correlation measures the extent to which two variables linearly relate to each other [143]. Suppose we are calculating the similarity between user a and u, then based on Pearson Correlation, it can be computed as:

$$\omega_{au} = \frac{\sum_i^M (r_{ai} - \overline{r}_a)(r_{ui} - \overline{r}_u)}{\sigma_a \sigma_u} \tag{2.10}$$

where M indicates the total number of items that have been rated by both users a and u. σ_a is the standard deviation of all ratings of user a.

Cosine Similarity

Cosine similarity [150] is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. Suppose we have two users a and u in a user-item-rating system, each of which can be represented as N dimensional feature vector, i.e. $\boldsymbol{s}$ and $\boldsymbol{t}$ respectively, then the cosine similarity between these two users can be presented as:

$$\omega_{au} = \frac{\boldsymbol{a} \cdot \boldsymbol{u}}{\| \boldsymbol{a} \cdot \| \| \boldsymbol{u} \|} = \frac{\sum_{n=1^N} s_n \times t_n}{\sqrt{\sum_{n=1}^N (s_n)^2} \sqrt{\sum_{n=1}^N (t_n)^2}} \tag{2.11}$$

2.8.2 Latent Factor Model-based CF: Matrix Factorization

Latent factor model based CF is an alternative approach for the neighborhood memory based approach whose principal idea is to uncover the latent features of each participating entity in a recommendation system that can explain the observed data. Under this scheme, each entity will be represented as a feature vector whose values

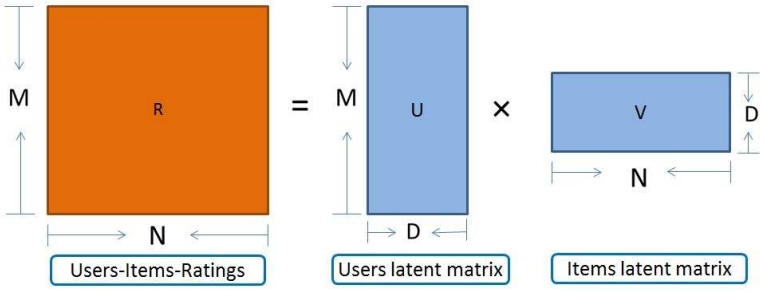

Figure 2.4: Matrix Factorization example

are unobserved. There exist a bunch of latent factor models, such as the pLSA [72] model, LDA [18] model, neural networks [149], singular value decomposition (SVD for short) [38] model, matrix factorization (MF for short) [97] and tensor factorization (TF for short) [140], among which, the MF method has shown to be the most state-of-the-art approach in recommender systems.

Basic Matrix Factorization

The fundamental mechanism of MF is to represent the relationships between two types of entities in a recommender system as a matrix, and that this matrix can be factorized into two lower dimensional matrices. Figure 2.4 shows an illustration over the traditional user-item rating recommender system.

As we can see, the left big matrix indicates the real interactions between users and items. Suppose we have M users and N items, the big matrix is of dimension $M \times N$, and each entry of the matrix r_{ui} represents the observed rating of user u to item i. In the right part, the big matrix is factorized into two lower dimensional matrices, each of which represents the latent factor space for users and items with dimensions $M \times D$ and $N \times D$ respectively, where D is the latent vector dimension and is normally much smaller than either M or N. Accordingly, each user u will be associated with a vector $p_u \in \mathbb{R}^D$ and each item i will be associated with a vector $q_i \in \mathbb{R}^D$. The resulting inner product of $q_i^T p_u$ captures the interaction between user u and i, and therefore approximates the observed rating of user u on i: r_{ui}. We can

denote this predicted value as:

$$\hat{r}_{ui} = q_i^T p_u \tag{2.12}$$

We refer equation 2.12 as the basic MF model. The major challenge is now to infer the mapping of each item and user to their associated latent vectors. Due to the data sparsity problem of the user-item rating matrix, addressing only the relatively few observed entries is prone to overfitting. In order to avoid that, regularization mechanism is introduced. To learn the latent factor vectors (p_u and q_i), the objective function is to minimize the regularized squared error on the set of known ratings, which can be presented as:

$$\min_{p*,q*} \sum_{(u,i)\in \boldsymbol{S}} (r_{ui} - q_i^T p_u)^2 + \lambda(\parallel q_i \parallel^2 + \parallel p_u \parallel^2) \tag{2.13}$$

where, $\boldsymbol{S}$ is the set of the (u, i) pairs for which their values r_{ui} are known. We refer this model as the regularized MF model.

Biased MF model

The basic MF model provides an open and flexible learning framework where various data aspects and application-specific requirements can be well accommodated. For example, in the user-item rating system, different users or items may have their own bias independent of any interactions. Empirical studies have shown that some users always tend to give higher ratings and that some items are easier to receive higher ratings; therefore it would be inaccurate to just model the interaction between user u to item i as $q_i^T p_u$ — their individual bias should also be considered. Accordingly, this leads to the biased version of the MF model, which is denoted as:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \tag{2.14}$$

where μ is the average ratings across all users and items in a particular user-item-rating system. b_i is the bias for item i, and b_u is the bias for user u.

Adding the regularization scheme to avoid overfitting, the regularized MF model with bias can be represented as:

$$\min_{p*,q*,b*} \sum_{(u,i)\in \boldsymbol{S}} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(\parallel q_i \parallel^2 + \parallel p_u \parallel^2) + b_u^2 + b_i^2 \qquad (2.15)$$

where latent factor vectors $q*$ and $p*$ and all the biases are the parameters that we need to learn from the training set with observed ratings. Once these parameters are inferred, the predicted rating over any unknown user and item pairs $\hat{r}_{ui}$ can be computed via following equation 2.14.

2.8.3 Solving Matrix Factorization Model

To solve the MF model, we need to estimate the value for parameters. Generally, there are two main optimization techniques that are widely used in recommender systems: the *Stochastic Gradient Descent* (SGD for short) [20] and the *Alternating Least Squares* (ALS for short) [5].

Stochastic Gradient Descent (SGD)

Stochastic gradient descent is a dramatic simplification of *gradient descent* [20] which is an iterative optimization technique. To better understand both the gradient descent and stochastic gradient descent, let us consider the following example.

Suppose we consider a simple supervised learning setup in which each sample in the training set z is a pair of (x, y) composed of an arbitrary input x and a scalar output y. ω is the associated weight for x in each data pair (x, y). We choose to use a function $f_\omega(x)$ to predict the value of y where ω is the parameter and denote the loss function as $l(\hat{y}, y)$ which measures the error between the predicted value of $\hat{y}$ and the real value of y. In order to compute the parameters ω and therefore the function $f_\omega(x)$, we need to minimize the loss $Q(z, \omega) = l(f_\omega(x), y)$ averaged on all samples in the training set.

Using the gradient descent method to compute $\boldsymbol{\omega}$ (the weight vector of all ωs associated with all xs), we can first randomly set the initial value of $\boldsymbol{\omega}$, and then

iteratively update its value until it finally converges. In each update iteration, $\boldsymbol{\omega}$ can be updated on the basis of its gradient in the descending direction:

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \gamma \frac{1}{n} \sum_{i=1}^{n} \bigtriangledown_{\boldsymbol{\omega}} Q(z_i, \omega_t) \tag{2.16}$$

where, γ is called the step-size or learning factor.

Stochastic gradient descent simplifies the computing procedure. Instead of computing the errors across all training samples and then get the gradients based on them, in SGD, each iteration estimates the gradient only on the basis of a single randomly picked example in the last iteration z_t, and $\boldsymbol{\omega}$ can be updated as:

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \gamma_t \bigtriangledown_{\boldsymbol{\omega}} Q(z_t, \omega_t) \tag{2.17}$$

The stochastic process $\boldsymbol{\omega}_t, t = 1, 2, \ldots$ therefore depends on the individual examples randomly picked at each iteration. Since the stochastic gradient descent algorithm does not need to remember all examples selected during previous iterations, it can improve computational efficiency enough to be part of a deployed system.

When applying SGD to solve the regularized basic MF model defined on the user-item rating system, each rating in the training set will be looped through, and in each such loop, both the user and item latent factor vector will be modified in the opposite direction of the gradient computed from the objective function. Based on equation 2.18, we can achieve the updating rules for q_i and p_u as:

$$\begin{aligned} q_i &\leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda \cdot q_i) \\ p_u &\leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda \cdot p_u) \end{aligned} \tag{2.18}$$

where $e_{ui} = r_{ui} - q_i^T p_u$ is the prediction error. γ is the learning rate.

The advantage of SGD is that it is efficient, easy to implement, and is applicable over large-scale and sparse machine learning problem. However, it is sensitive to feature scaling.

Alternating least squares (ALS)

Alternating least squares [5] is a block-coordinate descent algorithm whose fundamental idea is to minimize the objective function by updating one specific type of parameter while fixing all others and repeats the same process for each learned parameter sequentially, ensuring that each step would decreases the objective function until it finally converges.

When applying ALS to a regularized basic MF model for user-item ratings, since both p_u and q_i are unknown, equation 2.13 is therefore not convex. However, if one parameter is fixed, then the problem would become quadratic and there would exist a closed form for the optimization. Following this idea, the ALS algorithm will iteratively rotate between fixing p_us and q_is. When all p_us are fixed, the algorithm will recompute the value for q_is by minimizing the squared error. The same process will be done by fixing q_i while updating p_u. This entire procedure will be recursively executed with each iteration decreasing the squared error until finally converged.

2.9 Evaluation metrics

We introduce several measurements that have been widely used in recommender systems to evaluate recommendation/prediction performance. These include Precision, Recall, Accuracy, F_1 measure, RMSE, and AUC.

Precision, recall, accuracy and F_1 measure are all defined in terms of a set of retrieved entities (web documents, people, papers, or other entities) and a set of relevant entities. They can also be defined by using four traditional terms in classification task: true positive tp, true negative tn, false positive fp, and false negative fn. tp indicates the number of positive (relevant) entities that are also predicted as positive samples; fp is the number of entities that are actually negative (non-relevant) entities but are predicted as positive entities; fn indicates the number of entities that are actually positive but predicted as negative, and tn indicates the number of actually negative entities that are also correctly predicted as negative.

- **precision**: in traditional IR system, precision is the fraction of retrieved entities that are relevant to the search:

$$precision = \frac{\|relevant\ entities \cap retrieved\ entities\|}{\|retrieved\ entities\|} \tag{2.19}$$

or

$$precision = \frac{tp}{tp + fp} \tag{2.20}$$

- **recall**: recall indicates the fraction of entities that are relevant to a query that are successfully retrieved.

$$recall = \frac{\|relevant\ entities \cap retrieved\ entities\|}{\|relevant\ entities\|} \tag{2.21}$$

or

$$recall = \frac{tp}{tp + fn} \tag{2.22}$$

- F_1 **measure**: F-measure (or F-score) is the harmonic mean of precision and recall. In a general case, it can be computed as:

$$F_\beta = (1 + \beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{2.23}$$

The most widely adopted F-score is the the F_1 measure where β is set to be 1, indicating that precision and recall are evenly weighted. We have:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.24}$$

- **Accuracy**: Accuracy can be computed as: $Accuracy = \frac{tp+tn}{tp+tn+fp+fn}$

Precision@k, Recall@k and Accuracy@k are adopted for computing the corresponding Precision, Recall and Accuracy values for the top-k returned entities.

- **RMSE** (abbreviated for Root Mean Squared Error): RMSE is widely used for rating-related recommendations, such as user-item ratings or user-movie ratings prediction. It amplifies the contributions of the absolute error between the predicted values and real values. Suppose for a given user-item pair, the

real rating for user i to item j is r_{ij}, and the predicted value is $\hat{r}_{ij}$, then the overall RMSE value for the recommender system is:

$$RMSE = \sqrt{\frac{1}{\parallel \boldsymbol{T} \parallel} \sum_{(i,j)\in \boldsymbol{T}} (\hat{r}_{ij} - r_{ij})^2} \tag{2.25}$$

where $\boldsymbol{T}$ is the set of all user-item pairs whose ratings are predicted.

- **AUC** abbreviated for Area Under the ROC Curve: ROC curve is a two-dimensional depiction of a classifier's performance, on which the true positive rate ($\frac{tp}{tp+fn}$) is plotted on the Y-axis and false positive rate ($\frac{fp}{fp+tn}$) is plotted on the X-axis. AUC indicates the actual area under the ROC curve, which can be computed as:

$$AUC = \frac{s_0 - n_0(n_0+1)/2}{n_0 n_1} \tag{2.26}$$

where n_0 is the number of positive samples, n_1 is the number of negative samples, and $s_0 = \sum_{i=1}^{N} r_i$, r_i is the rank of the i^{th} positive sample in the ranking list, given that we have N positive samples in total.

Other ranking based IR metrics such as MAP and NDCG can also be used as the evaluation metrics for recommendation tasks whose definitions have been introduced in detail for the task of expertise retrieval.

Chapter 3

Expert Ranking: Topic-Driven Multi-Type Citation Network Analysis

In this chapter, we present an enhanced integrated probabilistic model which combines both content-based and graph-based approaches for expert ranking in an academic environment. We construct a heterogeneous academic network which consists of multiple types of academic entities. We introduce the application of Topical PageRank into link analysis over the academic network and propose a heterogeneous PageRank-like algorithm into exploring the impact of weighting various factors. Comparative experimental results based on data extracted from the ACM digital library show that 1) the multi-type academic network works better than the graphs integrating fewer types of entities, 2) the use of Topical PageRank can further improve performance, and 3) Heterogeneous PageRank with parameter tuning can work even better than Topical PageRank.

3.1 Introduction

Estimating researchers' contributions or reputations is of great importance since it can offer support when making decisions about researchers' job promotions, project funding approvals, and scientific award assignments. With the rapid development of academic digital libraries, the increasing volume of online scientific literature provides abundant sources of reputation evidence in terms of researchers' (authors') publications, as well as the citation relationships among these publications, both of which can be taken advantage of in evaluating researchers' reputations.

In order to evaluate the reputation of a researcher, especially within one scientific domain, there are typically two basic approaches. One is called the content-based approach, in which relevant documents representing expertise of a researcher can be considered, and information retrieval models can be applied to evaluate the relevance of these documents and thus authors to the query topic [8, 50, 108]. Researchers' publications in the academic digital libraries provides such good expertise resources.

Another important approach, which is also our main focus in this chapter, is via social network analysis [182]. The citation network[1] is one form of social network in which scientific factors, like authors and papers, can be represented as nodes, and their mutual interactions such as citations, can be modeled as edges.

Citation network analysis has long been a popular mechanism to evaluate the importance of publications and authors. Initially, citation analysis mainly focused on counting the number of citations [58, 59]. Under this scheme, an author will have higher reputation if he can be cited by many other authors.

With the recent success of graph-theoretic approaches in ranking network entities, researchers have begun to introduce link analysis approaches like PageRank [132] and HITS [93] into citation network analysis. Further attention has also been paid to integrate different kinds of citation networks, including a coauthor network for authors and a citation reference network for papers and take advantage of their mutual reinforcement to improve reputation ranking performance. The assumption

[1]We simply name this academic network as the citation network, since in this work, 'citation' is the primary relationship among scientific factors that we considered.

in this group of approaches is that more influential authors are more likely to produce high quality and thus highly cited papers, and well-cited papers can bring greater prestige to their authors.

In spite of the constant improvement in citation network analysis, including combinations with content-based approach, integration of different kinds of citation works, there still remain some limitations. For example, current citation network analysis seldom goes beyond that of the citation relationship among authors or papers. PopRank [131] integrates conferences and journals, yet there are still some other useful and easily available information in the scientific literature, such as authors' affiliations. In this chapter, we propose a novel probabilistic model which can integrate the citation between authors, papers, affiliations and publishing venues in a single model. Affiliation offers a good indication of authors' expertise, since high quality organizations tend to hire researchers (authors) with higher reputation.

In order to explore the different impact among factors, we propose a heterogeneous PageRank, permitting us to consider different propagation rates among factors. Furthermore, one distinguished contribution of our work is that we introduce the topical link analysis, which has shown success in web page authority evaluation, into citation network analysis. In summary, our main contributions include:

1. Proposing a novel probabilistic model which combines content-based analysis with a multi-type citation network, integrating relationships of authors, papers, affiliations and publishing venues in one model. This model can be extended to include more types of social factors.

2. Proposing a heterogeneous PageRank random surfer model compared to the original uniform PageRank model, to reflect the impact among different factors.

3. Introducing topical link analysis into citation network analysis. In particular, Topical PageRank [130] is adopted for citation link analysis.

4. A comparative study using ACM digital library data on various PageRank extensions as well as different complexity of citation networks.

The rest of this chapter is organized as follows. We introduce the multi-type citation network framework and the heterogeneous PageRank random surfer model in section 3.2. Section 3.3 introduces topical link analysis model. Experiments and results analysis are described in section 3.4. We review related work in section 3.5 and conclude this chapter in section 3.6.

3.2 Multi-type Citation Network Framework

In this section, we introduce the definition of our multi-type citation network framework. Two versions of the framework are considered, reflecting different relationships among factors.

3.2.1 Notation and Preliminaries

In a multi-type citation network, different kinds of social factors, as well as their mutual relationships are considered and integrated. The citation network can be formally denoted as $G = (V, E)$, where

- V is a set of nodes, representing social factors. In our current integrated network, V is combination of four different types of social actors: authors, papers, affiliations and venues.
- E is a set of directed edges, representing relationships among every pair of social actors. All the possible relationships we may have are the relationships between authors, papers, affiliations and venues.

Due to different relationships among the four types of social actors we can consider, we construct two versions of the multi-type citation network, to which we refer as 4-T graph version-1 (4-T) and 4-T graph version-2 (4-TV2) respectively.

3.2.2 Framework Version-1

In 4-T graph version-1, we consider the citation relationship among every pair of social factor types. The graph (shown in Figure 3.1) is directed and can be viewed

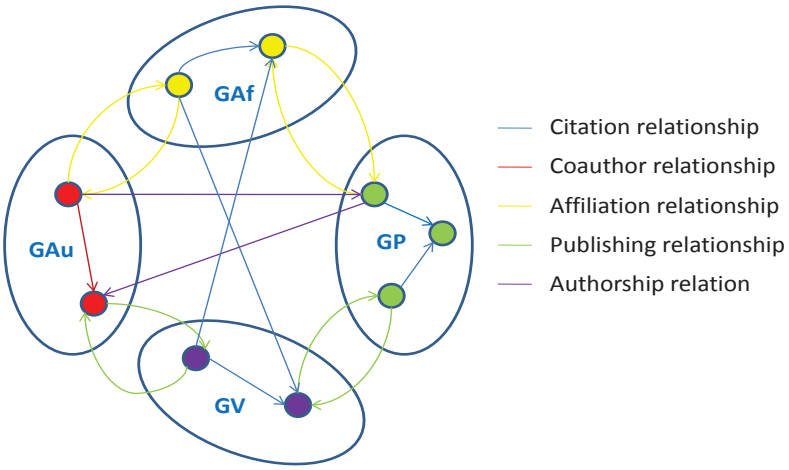

Figure 3.1: Multi-type (4-T) Citation Network version-1

as a combination of subgraphs, including those representing each of the types of social factors:

1. Author Graph G_{Au}. There would be one edge from author au_i to author au_j and one edge from author au_j to au_i if they coauthored at least one paper or if author au_i cites author au_j. We say that author au_i cites author author au_j if and only if there is at least one publication of au_i that cites one of the publications of au_j. We do not count the number of co-authorship or citationship in this framework, and thus there would be only one edge between two authors even though they coauthored more than once. The same mechanism works for other subgraphs defined in the following.

2. Paper Graph G_P. There would be one edge from paper p_i to p_j, if p_i cites p_j in its references.

3. Affiliation Graph G_{Af}. There would be one edge from af_i to af_j if two authors, each of which comes from af_i and af_j respectively, coauthor in at least one paper, or there is at least one paper produced in affiliation af_i that cites one of the publications from af_j.

4. Venue Graph G_V. One edge will be drawn from v_i to v_j if there is at least one paper which is published in v_i that cites one of the papers published in v_j.

as well as graphs that relate one type of social actor to another:

1. Bipartite AuthorPaper Graph G_{AuP}. There would be one edge from au_i to p_j, if au_i is one of the authors of p_j. Correspondingly, there would one edge from p_j to au_i, indicating that it is written by au_i.

2. Bipartite AuthorAffiliation Graph G_{AuAf}. One edge would be drawn from au_i to af_j and af_j to au_i, if au_i belongs to the affiliation of af_i. One distinct author may belong to different affiliations in different periods of time; thus it is possible for one author node to point to several affiliation nodes.

3. Bipartite AuthorVenue Graph G_{AuV}. If there is at least one paper written by au_i and published in v_j, there would be a corresponding edge from au_i to v_j and from v_j to au_i.

4. Bipartite PaperAffiliation Graph G_{PAf}. One edge will go from paper p_i to affiliation af_j if p_i is written by an author that belongs to af_j.

5. Bipartite PaperVenue Graph G_{PV}. One edge will go from p_i to v_j and v_j to p_i if p_i is published in v_j.

6. Bipartite AffiliationVenue Graph G_{AfV}. If there is one paper belonging to affiliation af_i published in v_j, there would be an edge from p_i to v_j and from v_j to p_i.

3.2.3 Framework Version-2

There may exist redundant information within edges in version-1, since most relationships are generally inferred from the citations among papers (some others are generated via coauthor-ships). As a result, we introduce a simplified version of the graph.

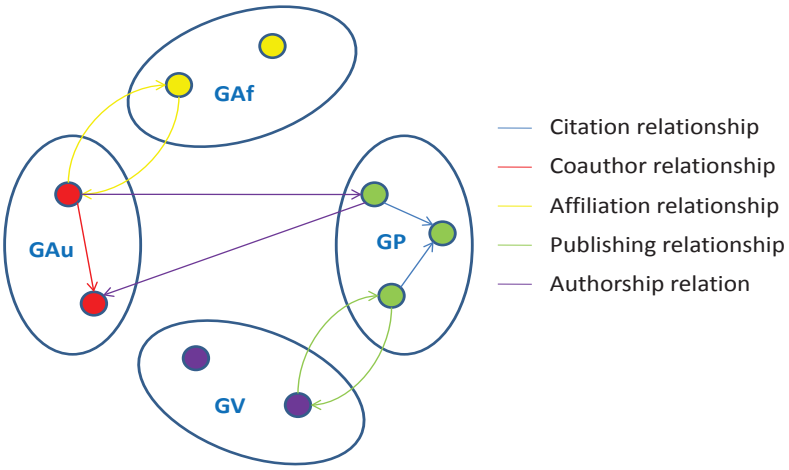

Figure 3.2: Multi-type (4-TV2) Citation Network version-2

In this simplified version, we only consider the coauthor relationship between authors, while ignoring the citation relationship between them. Affiliation nodes will only be connected with author nodes, and venue nodes will only be connected with paper nodes. There are no direct edges within the affiliation graph and venue graph. The relationships between authors and venues can be related by firstly relating authors to papers, and then papers to venues. A similar process works when representing the relationship between affiliations and papers. Figure 3.2 illustrates the simplified version of the multi-type graph.

3.2.4 Heterogeneous PageRank

In the original homogeneous PageRank, each node evenly distributes its authority score among its children. Using such an even propagation in the multi-type citation network, author nodes will evenly distribute its authority to other authors, papers, affiliations, and venues (under framework version-1), which may not well represent the actual interaction possibilities among nodes of different entities. In order to better represent the different impact among multiple types of social actors, we propose a heterogeneous PageRank algorithm based on the assumption that there would

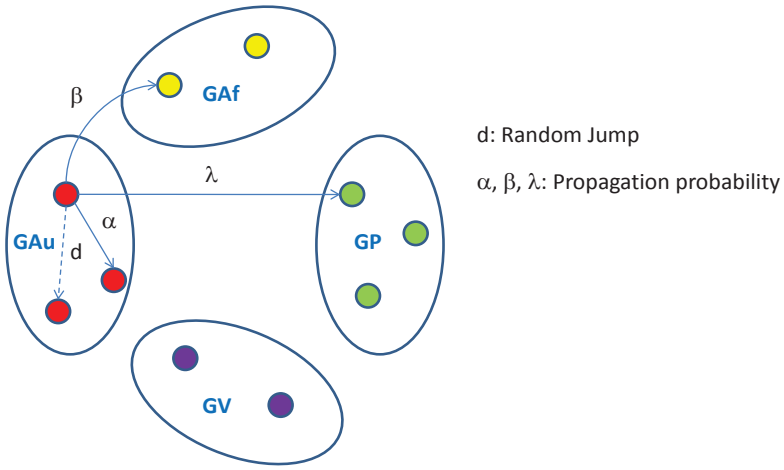

Figure 3.3: Heterogeneous PageRank

be a different propagation probability for a node to follow different kinds of out-going links (links to different types of nodes). (See Figure 3.3) This heterogeneous PageRank can be described as:

$$PR(i) = (1-d)\sum_{j:j\to i}\beta_{ji}\frac{PR(j)}{O(j)_{type(i)}} + d\frac{1}{N} \tag{3.1}$$

where:

- j and i are two nodes of any types, where j has out-going link to i.
- d: random jump.
- β_{ji}: is the parameter determining the propagation probability from node j to i. β_{ji} is equal to β_{jk} if node i and node k are of the same type. $\sum_{type(i)}\beta_{ji} = 1$, where node j has an out-going link to i.
- $O(j)_{type(i)}$ is the number of outlinks j has to the nodes of the same type with i.
- N: total number of nodes in the network.

3.3 Topical Link Analysis in citation networks

In our description so far, all social factors in the citation network are given one single global score, which represents their authority for all topics. However, a researcher who is an expert in information retrieval may not be an expert in computer architecture. Under such circumstances, it is more reasonable to give authority score for researchers in terms of their reputation for different topics. In Web domain, some ranking schemes are designed to take topical information into account (e.g., as in [67, 130]). In this section, we first review Topical PageRank [130], one successful ranking scheme, and show how it can be adapted into citation network analysis.

3.3.1 Topical PageRank

The basic idea of Topical PageRank [130] is to incorporate a topic distribution into the representation of each web page as well as the importance score of each page. Therefore, there are at least two vectors associated with each page: the content vector C_u : $[C(u_1), C(u_2), ..., C(u_T)]$, which is a probability distribution used to represent the content of u across T topics, and the authority vector, A_u : $[A(u_1), A(u_2), ..., A(u_T)]$, which is used to measure the importance of the page, where $A(u_K)$ is the importance score on topic K.

Topical PageRank is also a random surfer model. On each page, the surfer may either follow the outgoing links of the page with probability $1 - d$ or jump to a random page with probability d. When following links, the surfer may either stay on the same topic to maintain topic continuity with probability α ("Follow-Stay") or jump to any topic i on target page with probability $1 - \alpha$ ("Follow-Jump"). The probability of jumping to topic i is determined by $C(u_i)$. When jumping to a random page, the surfer is always assumed to jump to a random topic i ("Jump-Jump"). Thus, the surfer's behavior can be modeled by a set of conditional probabilities:

$$\begin{aligned} P(Follow - Stay|v_k) &= (1-d)\alpha \\ P(Follow - Jump|v_k) &= (1-d)(1-\alpha) \\ P(Jump - Jump|v_k) &= d \end{aligned} \tag{3.2}$$

And the probability to arrive at topic i in target page u by the above actions can be described as:

$$\begin{aligned} P(u_i|v_i, Follow - Stay) &= \frac{1}{O(v)} \\ P(u_i|v_k, Follow - Jump) &= \frac{1}{O(v)} C(v_i) \\ P(u_i|v_k, Jump - Jump) &= \frac{1}{N} C(v_i) \end{aligned} \tag{3.3}$$

where $O(v)$ represents the out-degree of page v. Therefore, the authority score $A(i)$ on page u is calculated as follows:

$$\begin{aligned} A(u_i) = (1-d) \sum_{v:v\to u} \frac{\alpha A(v_i) + (1-\alpha) C(v_i) A(v)}{O(v)} \\ + \frac{d}{N} C(u_i) \end{aligned} \tag{3.4}$$

where $A(v) = \sum A(v_i)$. Note that authors in [130] also proposed a Topical version of the HITS algorithm, which we leave for future work.

3.3.2 Topical Citation Analysis

Inspired by the principal idea and demonstrated success of Topical PageRank in ranking web pages, we want to introduce such a topical link analysis approach into authors' reputation ranking. Similar to web pages, publications may also cover different topics, and thus when paper a cites paper b, it may do so because paper a finds one specific topic t in paper b to be interesting and useful. The same is true for authors' authority propagation. Believing in the prestige of a person on one aspect (say, for example, on data mining) does not mean that this person also owns a high reputation on other aspects (e.g., networking). When authors choose to collaborate and coauthor with each other, they may have mutual trust and interests on some certain aspect (topic). Publishing venues are normally more focused on certain research areas than others. SIGIR, for example, has a high prestige in the information retrieval research field, while SIGCOMM is well-established in the networking domain. Compared to papers, authors or venues, affiliations have less obvious topic-specific differentiation; however, we can still imagine that one affiliation is better

Table 3.1: Queries

algorithms and theory	security and privacy
hardware and architecture	software engineering and programming language
artificial intelligence	machine learning and pattern recognition
data mining	information retrieval
natural language and speech	graphics
computer vision	human computer interaction
multimedia	networks and communications
world wide web	distributed and parallel computing
operating systems	databases
real time and embedded systems	simulation
bioinformatics and computational biology	scientific computing
	computer education

known for doing certain kinds of research than others.

3.4 Experimental Work

3.4.1 Data Collection

We conducted experiments on the ACM data set (see introduction in Section 2.4) which consists of 172,890 papers, 170,897 authors, 45,965 affiliations, and 2,197 publishing venues. After extracting these factors (paper, authors, affiliations, venues, and the citation relationship among papers), we constructed two versions of the multi-type citation network as we introduced in Section 3.2.

3.4.2 Evaluation

In the portal website of Microsoft Academic Search (abbreviated as MAS)[2], which is a free computer science bibliography search engine, we found 23 categories (listed

[2]http://academic.research.microsoft.com/

in Table 3.1) covering the main 23 disciplines of computer science research. We used these 23 categories as testing queries, as they represent reasonable topics on which searchers might look for papers, authors, or conferences.

While the link-based citation network analysis is our research focus, we did not use it exclusively for retrieval. Instead, we combine it with the use of a content-based approach. For each author, a profile is constructed by concatenating all of the author's publications in terms of title, abstract and ACM categories. The Okapi BM25 [146] weighting function is used to evaluate the relevance of the authors' profile to the queries. As a result, for each author, there would be two ranking results: one from using BM25, and the other from a link-analysis approach. These two rankings can then be combined as follows:

$$\lambda * rank_{BM25}(a) + (1 - \lambda) * rank_{CitationNetwork}(a) \tag{3.5}$$

Since we lack a standard evaluation benchmark for the dataset, we developed three different approaches to measure the performance of expert ranking algorithms.

In the first approach, we collected all the PC members in the related conferences for each research area during 2008 and 2009. Microsoft Academic Search (previously known as Libra) provides a ranked list of conferences for each of its 23 categories. We retrieved the top 10 conferences for each category and collected their 2008 and 2009 PC members. For those conferences which have no 2008 or 2009 conference, we simply collected the PC members of its two most recently held conferences. To be qualified to participate as a PC member is a reasonable indication of the academic reputation of a researcher. To assign different "relevance" scores for those PC members, we normalized across the number of years (two at the most) and the number of different conferences in which one performs as a PC member.

In the second approach, we collected all the ACM fellows, ACM distinguished and senior members provided from the ACM website. Since there are not research area descriptions for ACM distinguished and senior members, we manually labelled the members into different categories and thus we only used a subset of ACM distinguished and senior members to generate our relevant lists. The subset we retrieved

is determined by the mixed group of top 60 ranked authors from all ranking algorithms. Since we focus on top-ranked results in our evaluation metrics, we believe this subset can provide us enough evidence to judge authors. To differentiate the relevance score, we gave a relevance score of 3 to ACM fellows, 2 to distinguished ACM members and 1 for ACM senior members.

We utilized human judgements to generate relevant lists in the third approach. In our evaluation system, the top ten and twenty returned authors by various ranking algorithms were retrieved and mixed together. We then manually but blindly judged the relevance for each author in the mixed ranking list with the corresponding query. Four judges were asked to search using Google Scholar (or other web search engines) using the author name as query, and go through returned webpages (homepages in most cases) related to the author to make a judgment on his professional prestige.

After generating the relevant lists, we can compute and compare the retrieval and ranking performance of different ranking algorithms. We took the well-known metric, the Normalized Discounted Cumulated Gain (NDCG) as our main metric. We tested on NDCG@10 and NDCG@20 respectively.

3.4.3 Experimental Results

We made several groups of comparisons to test the performance of different algorithms in their abilities of finding the most influential authors in 23 different research fields (represented as queries).

Multi-type Citation Network

Figures 3.4 to 3.7 indicate the NDCG results for different kinds of citation network analysis approaches using original uniform PageRank as propagation mechanism and using ACM members and PC members as evaluation dataset respectively. Table 3.2 shows the results from human judgements. To reduce the amount of manual labelling, we only gave to judges the results when combined with BM25 with parameter λ set to 0.5. We also introduced the ranking method of in-domain citation count as one of our compared approach. We took the 23 categories provided by

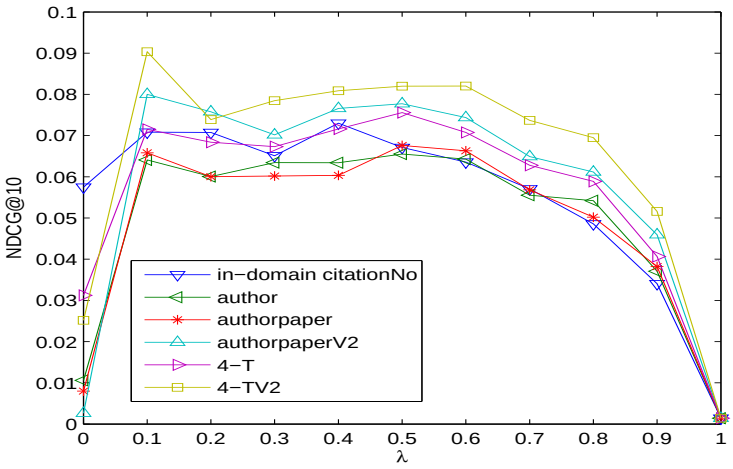

Figure 3.4: Comparison among different levels of citation network (NDCG@10 for ACM members) as the BM25 weight (λ) is varied.

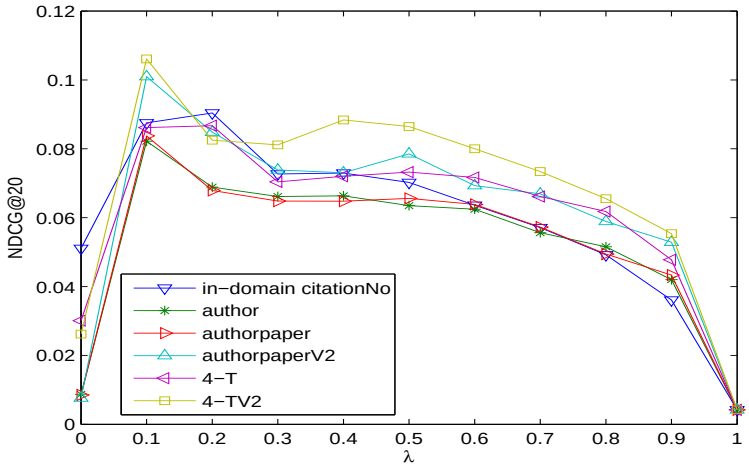

Figure 3.5: Comparison among different levels of citation network (NDCG@20 for ACM members) as the BM25 weight (λ) is varied.

MAS as domains, and regarded it as in-domain citation if two papers are within one domain and there is a citationship between them.

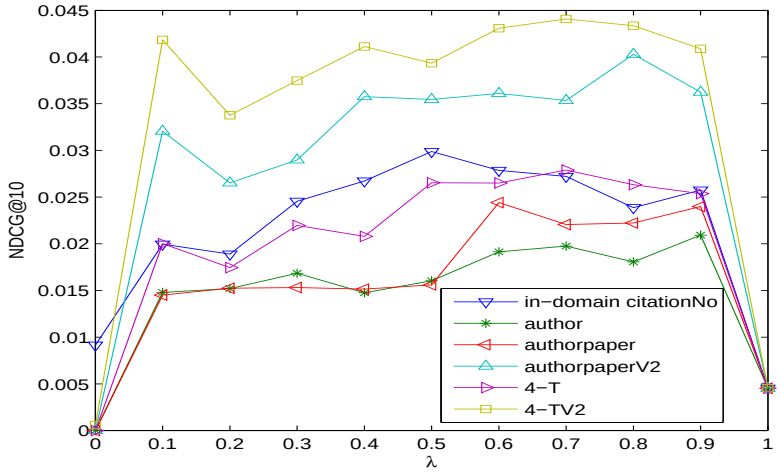

Figure 3.6: Comparison among different levels of citation network (NDCG@10 for PC members) as the BM25 weight (λ) is varied.

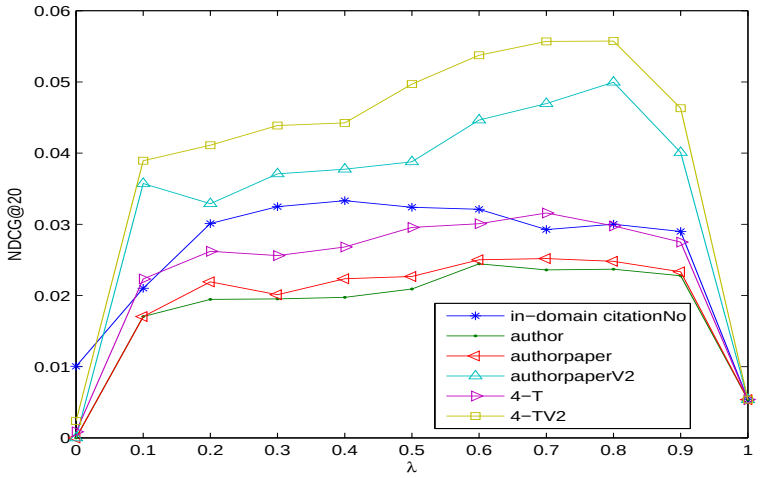

Figure 3.7: Comparison among different levels of citation network (NDCG@20 for PC members) as the BM25 weight (λ) is varied.

Several conclusions can be drawn from these results. First, there is a noticeable consistency with regard to the performance of ranking algorithms for the three different evaluation methods. 4-TV2 always works the best in all scenarios. This

Table 3.2: NDCG Results from human judgements (λ=0.5)

citation graph	NDCG@10	NDCG@20
in-domain citationNo	0.6820	0.6748
author	0.6390	0.6025
atuhorpaper	0.6455	0.6167
authorpaperV2	0.6899	0.6614
4-T	0.6545	0.6401
4-TV2	0.6988	0.6889
Topical 4-T	0.7004	0.6848
Topical 4-TV2	**0.7490**	**0.7231**

demonstrates our initial intuition that affiliations and conference venues can provide useful information and thus make them important and unignorable social factors in determining authors' reputations, and that the mutual reinforcement among different factors can improve ranking performance.

Secondly, we also noticed that different versions of the citation graph do have different impact on the overall performance. From the above figures, we find that version2 always work better than version1. This may be caused by the fact that removal of possible duplicate citation relationships can avoid authority being scattered over duplicate links. The results give us an indication that we should not only consider increasing the number of social factors to explore, but also need to pay attentin to how to effectively find the relationships among them and thus properly organize them.

We note that the absolute NDCG values for ACM members and PC members are comparatively low, but this may be caused by the incomplete collection of papers from the ACM digital library, and the incomplete citation relationships we collected. As we mentioned before, we only took those citations for which we have also crawled the corresponding web page into account. Besides, some distinguished researchers may have published in many journals or other papers which are not normally collected by the ACM digital library. However, since there is a high consistency among

Table 3.3: Top-level topics from the ACM Digital Library.

computer applications	computer systems organization
computer aided engineering	computing methodologies
computing milieux	data
general literature	hardware
information systems	mathematics of computing
software	theory of computation

all the different evaluation approaches, and the NDCG value of using human judgement is pretty high, we can have confidence in the evaluation using ACM members and PC members.

Topical PageRank

A key issue in Topical PageRank is to generate the static per-document content vector. We made use of the hierarchically-organized ACM categories provided by authors of each paper for topic distribution generation. We extracted the top level primary category and additional category for all the papers in the dataset and thus get 12 categories in total (listed in Table 3.3). We regard these categories as topics. With category information provided, computing topic distributions for papers is straightforward.

Since each author is represented by a profile which is a concatenation of all the papers he has written, we can accumulate all the topics mentioned by each published paper, and then compute the topic distribution. The same mechanism works for computing the topic distribution of venues, for which we collected all the papers published in that venue, accumulated papers' topics, and then computed the corresponding distribution. We did the same thing for generating affiliations' topic distribution by collecting the papers written by authors from that affiliation, and taking use of the papers' topic distribution to compute the affiliations' topic distribution.

Since we take the categories provided by MAS as experimental queries, and Microsoft Academic Search also lists group of papers for each category, we randomly

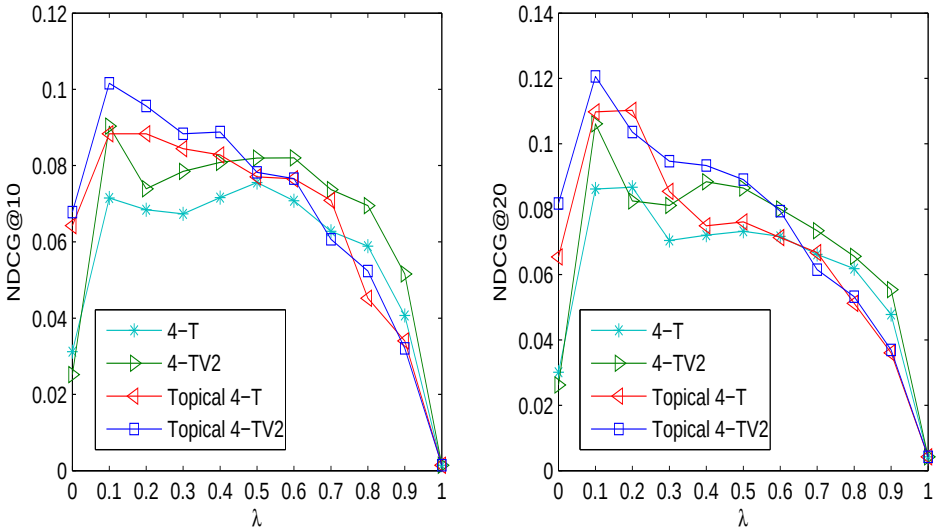

Figure 3.8: Topical PageRank Performance (NDCG@10 and NDCG@20 for ACM members) as the BM25 weight (λ) is varied.

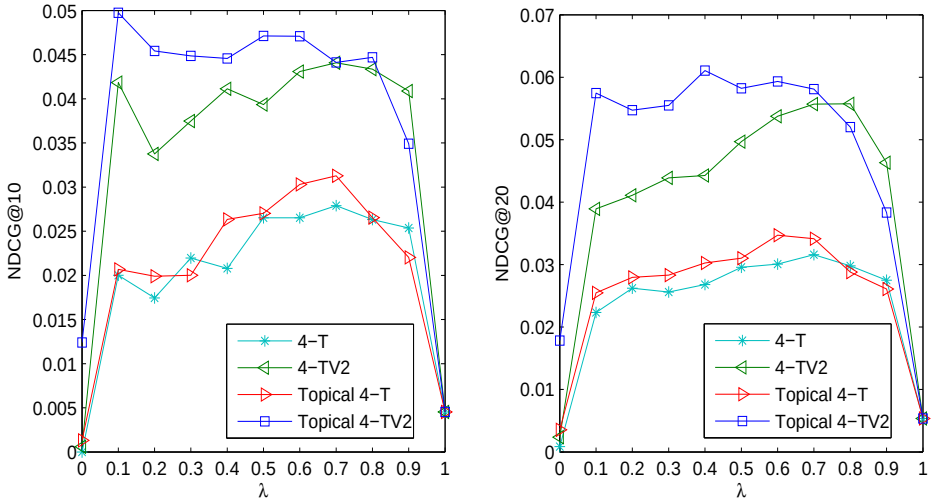

Figure 3.9: Topical PageRank Performance (NDCG@10 and NDCG@20 for PC members) as the BM25 weight (λ) is varied.

chose five papers from each category and take use of papers' topic distribution to compute the topic distribution for queries.

See figs. 3.8 and 3.9 for the results of topical experiments.

Table 3.4: TopicalV2 vs CoRank (on PC members)

Citation graph	NDCG@10	NDCG@20
Topical 4-TV2	**0.0497**	**0.0611**
CoRank	0.0219	0.0308

We set the α value to be 0.85 in all experiments. We found once again a high consistency among the results from different approaches , and that introducing Topical PageRank can improve the performance indeed. The improvement of the best performance of Topical 4-TV2 over that of 4-TV2 is 12.9% (NDCG@10) and 14.2% (NDCG@20) for ACM members, 12.7% (NDCG@10) and 9.7% (NDCG@20) for PC members, and 6.8% (NDCG@10) and 5.1% for human labelling results.

Comparison with two baselines

One of the main characteristics of our multi-type citation network analysis approach lies in its combination of both content-based approach and graph-based approach. We took two other approaches as our comparison baselines, one is BM25, a purely content-based approach, and the other is the CoRank approach proposed in [206].

The results of incorporating BM25 has been shown in all the above figures, since it is equivalent to pure BM25 when λ is set to be 1. As we can see, our multi-type citation network outperforms BM25 in all different scenarios.

The CoRank algorithm generates author and paper rankings by taking propagation between authors and papers into account. It is a graph-based approach. Instead of building a big graph for all the authors and papers in the dataset, they first rank authors in terms of their topic weights in a certain domain, retrieve the top 500 authors, and build the graph based on these authors and their publications. The graph they build is thus query-specific. We have implemented this algorithm (we determined the topic weight by counting the number of papers belonging to a topic (query)), and Table 3.4 compares the results between CoRank and our TopicalV2 at its best performance. As we can see, TopicalV2 outperforms CoRank. We used PC members for evalution in this experiment.

Table 3.5: NDCG@20 for Heterogenous PageRank

Best Perf. on Training	Parameter settings p1 p2 p3 p4 p5 p6 p7	Perf. on Test
0.0944	0.4 0.1 0.5 0.2 0.3 0.5 0.6	0.0905
0.0868	0.6 0.1 0.3 0.4 0.1 0.5 0.5	0.1262
0.1108	0.6 0.1 0.3 0.4 0.1 0.5 0.4	0.0040
0.0957	0.4 0.3 0.3 0.4 0.1 0.5 0.2	0.0929
0.1045	0.6 0.1 0.3 0.4 0.1 0.5 0.4	0.0465
Average performance on Test		0.0720

Heterogeneous PageRank

We propose a heterogenous PageRank algorithm with the intention of exploring the different impact among social factors. The parameter β_{ij} indicates the authority propagation probability among factors i and j. It is actually a parameter optimization problem if we want to get the best performance by tuning the parameters.

We work on 4-TV2, and thus there are seven parameters in total: the propagation probability between authors to authors (p1), authors to papers (p2), authors to affiliations (p3), papers to authors (p4), papers to papers (p5), papers to venues (p6), and the combination parameter with BM25 (p7). We perform greedy search, testing on the possible combinations of the parameters with a stepsize of 0.2 (the combination parameter p7 with BM25 has a stepsize of 0.1). In order to test the system performance on unseen data, we further group the 23 queries into 5 groups, and use five-fold cross validation approach to evaluate system performance. We evaluate on PC member-based evaluation.

The algorithm under different parameter scenarios converges within 8-17 iterations. As indicated in Table 3.5, the average performance using heterogeous PageRank is even better than the best performance of topical 4-TV2 (0.0611, which is the best performnace in all the previous experiments). This improvement is around 17.8%. This demonstrates our initial intuition that considering different effect among factors can improve performance.

3.5 Bibiliographic Notes

Citation analysis has a long history in assessing the research performance of individual scholars, publishing journals or papers, as well as research groups. Originally, citation analysis focused on counting the number of citations. Journal impact factor [58, 59], the most classical citation indicator, is defined as the average number of citations per article a journal receives over a two-year period. Hirsch number (h-index) [71], another famous citation indicator, is also defined in terms of citation counts.

Inspired by the success of graph-theoretic approaches in ranking network entities, scientists gradually realized that simply counting the number of citations cannot represent well the true prestige. Without distinguishing between citations, the citation from a good paper with high impact will have the same weight as citations with lower impact. Pinski [137] was the first person who realized this problem and proposed an improved recursive approach. With the great success of link analysis approaches, like PageRank and HITS in ranking web pages' authorities, much recent research work, such as that by Chen et al. [31], has introduced the PageRank algorithm into citation network analysis replacing hyperlinks with citation references.

Further research work has been carried out in combining the content-based approach with citation network for reputation evaluation. P. Glenisson et al. [60] combined full text and bibliometric information in mapping scientific disciplines, and Bogers et al. [19] made the first investigation into combining and comparing the citation analysis with content-based approach for finding academic experts.

Research work has been carried beyond the citation network analysis domain in integrating different types of entities. Davison [36] proposed a unified approach to analyze multiple term and document relationships. With similar idea, a so-called link-fusion [185] unified link analysis framework has been proposed which considered both the intra- and inter- type link structure among multiple-type inter-related data objects. Most recently, Guan [64] proposed a multi-type framework integrating users, documents and tags for tag recommendation. In [181], Wang et al. proposed a more general and fundamental method for analyzing semantic relations among

any multiple type of data. Compared to these works mentioned above,our emphasis in this chapter is using multi-type factors integration for citation network analysis.

Similar to those work in web or data management research domains, researchers have already started to pay attention to the integration of different kinds of citation networks. The assumption is that different citation relationships can mutually reinforce each other, and thus can improve ranking performance. Zhou et al. [206] is a representative work in this direction in which they proposed a query-specific co-ranking framework which can integrate an author-coauthor relationship network and the paper citation network. Compared to their work, our multi-type network provides a more comprehensive framework, and our proposed citation network framework is a global, query-independent one.

PopRank [131] is another representative work whose main idea has been implemented in Microsoft Academic Search[3], a free academic search engine. One advantage of PopRank is that it integrates conferences and journals in addition to authors and papers into consideration. Our framework integrates one additional factor: author affiliation and we combine content-based analysis and link structure analysis in our framework.

One distinguished contribution of our work, compared to all others discussed above, is that we introduce topical link analysis into consideration. In web research domain, many improvements to PageRank have been proposed, including Topic-Sensitive PageRank [67] in which a separate PageRank score calculation is performed for each topic. With that influence, Nie et al. [130] proposed a Topical PageRank and Topical HITS model which embed topical information into authority propagation and demonstrated better performance over original PageRank and HITS. Even though there has been research work showing use of topical information in analyzing authors' publications content (e.g., [123, 111]), no research work, to the best of our knowledge, has introduced topical information into citation network link analysis. We remedy this situation.

[3]http://academic.research.microsoft.com/

3.6 Summary

Previous work has investigated the value of integrating author and paper information in citation networks in ranking authors' reputations. PopRank is a work we have identified which integrated conference venues into consideration. We further observed that there are yet more useful information we can extract and make use of, for example, affiliations. To test this idea, we proposed in this chapter a multi-type citation network framework which integrates citations among authors, papers, affiliations and publishing venues into one model, and used a PageRank-based algorithm to rank authors' authority. In order to test the different impact among factors, we further proposed a heterogeneous PageRank algorithm in which social factors may propagate authority to neighboring factors with different probabilities. Moreover, in order to better evaluate the prestige of an author in different kinds of research topics, we incorporated topical link analysis into the citation network. We conclude from experimental results that:

- Multi-type citation networks can effectively improve ranking performance. Affiliation and publishing venues provide additional useful information in evaluating authors' reputations.
- Topical link analysis shows positive improvement in ranking authors' authority.
- Heterogeneous PageRank, with parameter tuning, can work even better than Topical PageRank.

Chapter 4

Expert Ranking: Integrating Learning-to-Rank with Topic Modeling

In the previous chapter, we present an integrated model which combines both content-based and enhanced PageRank-like graph-based approaches into expert finding. Particularly, the traditional BM25 approach is used for retrieving the content-based relevancy for authors over queries. However, due to the data sparsity problem, the bag-of-words based approach (i.e.: BM25) cannot accurately discover the latent semantics of authors' supporting documents and therefore may deteriorate the performance in evaluating authors' expertise. Generative topic modeling offers a good solution in capturing the underlying meanings. We therefore focus on providing improved topic modeling based approach into expert finding. On the other hand, even though both probabilistic discriminative models and generative models have been proposed to tackle the problem of expert ranking, the combination of them is seldom explored. In this chapter, we introduce a pairwise learning-to-rank framework into topic modeling, making the traditional unsupervised topic modeling process a supervised one. Such a combination can help us solve the data sparsity problem, and provides a platform to integrate additional features of authors.

4.1 Introduction

Generative topic modeling has become a popular machine learning technique and has shown remarkable success not only in text mining, but also in modeling authors' interests and influence, and predicting linkage among documents (authors). Ever since the success of the original two representative topic models, the pLSA [72] and LDA [18], which focus on pure content analysis by discovering the latent topics from large document collections, a large body of literature on topic models has been established, mostly by incorporating additional contextual information, such as time, geographical locations, or integrating linkage or social network information. Authorship is one important contextual feature, which when incorporated into topic modeling, can be used to derive the topic distribution over authors rather than documents, and therefore can be used to model authors' interests and influence.

Most of the existing topic models, however, are unsupervised. Documents or authors are treated equally, while no prior-knowledge of their different importance over topics has been explored or investigated. However, this may not well represent the real situation, in which we sometimes can know in advance that some document is more about a certain topic than other documents, and that one author (researcher) is more prestigious in one research domain than other authors. By exploring this prior-knowledge and applying a supervised learning scheme into the topic modeling process, we hypothesize that we can achieve more accurate and cohesive topic modeling results, which can in turn help in better distinguishing the different importance (ranking) of new documents (authors) in terms of their relevance or authority over topics.

In this work, we concentrate on the ability of topic models in modeling authors' authority (interests or influence) in a research domain[1], a typical task known as expert ranking (expertise ranking or expert finding). In spite of many recent developments fulfilling this task, several challenges still remain. First of all, the sparseness problem in document content would prevent the 'bag-of-words'-based

[1] in this chapter, we use research domain, community and its associated query as interchangeable concepts

algorithms (term frequency, TFIDF, language model) from being accurate. It is well-acknowledged that documents related to an author provide strong evidence in evaluating authors' expertise; however, such document content (especially considering the paper abstract) is normally very sparse, and therefore, a 'bag-of-words' based algorithm cannot effectively capture the underlying semantics. The topic modeling approach, however, is believed to provide a better solution in this aspect. Secondly, few existing topic modeling based approaches incorporate additional features such as network based features and temporal features into the topic modeling process to represent an author's authority. Thirdly, most of the existing work on expert ranking rely on carefully designed ranking models based on heuristics or traditional probabilistic principles, rather than applying machine learning techniques to learn ranking functions automatically.

To fulfill the challenges mentioned above, we propose in this paper a supervised learning scheme by incorporating the prior knowledge of the different importance over topics between pairs of authors into the topic modeling process, which results in a framework integrating the pair-wise learning-to-rank algorithm into topic modeling. We name this novel model as LtoR topic modeling (abbreviated as **LtoRTM**). In the training process, we can not only infer the authors' distribution over topics and topics' distribution over words, but also the coefficient representing the different weights of topics. In the testing process, we can infer the topic proportion of new authors. Furthermore, based on the new authors' topic distributions, and the learned coefficient in the training process, we can generate a ranked list of authors in terms of their different importance (authority) across topics.

We go beyond pure contextual information by incorporating additional features into the LtoRTM model, such as the number of publications or citations of authors. To evaluate the effectiveness of our proposed model, we apply the model to two expert ranking related applications: the task of predicting community-based future award winners and predicting future PC members of several significant conferences in computer science disciplines. To sum up, our paper has made the following contributions:

- We propose a supervised learning scheme by distinguishing the different importance of pairs of authors into author topic modeling process. To our best knowledge, this results in the first framework integrating pair-wise learning-to-rank into topic modeling.

- We identify additional features besides the pure contextual information, and integrate them into the proposed model framework.

- We evaluate the effectiveness of our model by applying it to two applications measuring author authorities: the tasks of predicting future award winners and future PC members. Experiments have been conducted on real-world data sets to test the performance of the proposed model and compare it with several other state-of-the-art topic modeling or learning-to-rank algorithms.

The rest of the chapter is organized as follows. We present the model design for **LtoRTM** and **LtoRTMF** in section 4.2, and introduce the model inference, parameter estimation and ranking process in section 4.3. Evaluation tasks, experiment settings and results are discussed in section 4.4. We review related work in section 4.5 and conclude this chapter in section 4.6.

4.2 Model Design

This novel topic model we develop is a hierarchical probabilistic model, where each document is associated with attribute information. In this section, before introducing the two models (**LtoRTM** and **LtoRTMF**) we proposed, we provide a brief overview of the basic Latent Dirichlet Allocation (LDA [18]); we then introduce the **LtoRTM** model where only pure contextual attributes, i.e., the words of the documents, are considered, and finally the **LtoRTMF** model where additional features with regards to authors' expertise are incorporated.

LDA considers each document d_i in the data collection to be a mixture of T topics, each of which is a mixutre of W words, where W here is the total number

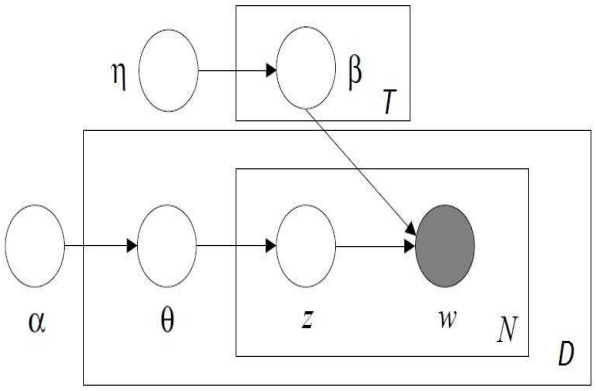

Figure 4.1: Graphical Model for original LDA

of distinct words in the entire data collection. Each document d_i of length N_{d_i} is modeled by the following generative process:

- draw $\theta_{d_i}|\alpha \sim Dir(\alpha)$, a multinomial distribution over T topics;
- for each word position k in document d_i:
  - draw a topic $z_{d_i,k} \sim Multi(\theta_{d_i})$
  - draw a word $w_{d_i,k} \sim Multi(\beta_{z_{d_i,k}})$

Using the original LDA, documents' topic proportion θ_{d_i} indicating documents' relevance to topics is only learned from their individual content. This may not well represent the real situation when we have prior knowledge that document d_i is more relevant than document d_j to topic k. We hythposize that by integrating this prior knowedge into topic modeling process, more accurate topic proportion is likely to be achieved. This basic idea stimulates the proposal of the **LtoRTM** model, which can be further extended into **LtoRTMF** model by incorporating additional features.

4.2.1 Model Description and Generative Process

The **LtoRTM** model builds upon the previous works, including [28, 46], which extend the original LDA model by incorporating linkage between pairs of documents

Table 4.1: Notation

Symbol	Size	Description
W	scalar	size of word vocabulary
D	scalar	number of papers
T	scalar	number of latent topics
N_d	scalar	the number of words in paper d
N	scalar	the number of words in corpus
		Observed Data
$\boldsymbol{w_d}$	$\lvert\boldsymbol{w_d}\rvert$	the words lists of paper d
$\boldsymbol{w}$	N	the set of word tokens in corpus
y^c_{didj}		binary indicator
		Hyper-Parameters
α	$1 \times T$	Dirichlet prior for θ
η^c	$1 \times T$	coefficient
η^c_1 η^c_2	$1 \times (T + \lvert F\rvert)$	coefficient
		Random Variables
θ	$A \times T$	distribution of authors over topics
β	$T \times V$	distribution of topics over words
$\boldsymbol{z_{di}}$	$1 \times T$	topic assignments for ith word in paper d

into topic modeling process. However, two characteristics distinguish our model from previous work. Firstly, we focus on modeling author interests and influence, Therefore, instead of modeling individual documents, we construct a virtual profile to represent each author (researcher) by concatenating all his/her publications. As a result, the topic proportion we derive for each virtual profile represent authors' distribution (authority) over topics. In the following part of this chapter, we use document and virtual profile interchangeably. Secondly, we model the difference between pairs of author virtual profiles in terms of their topic distribution rather than the linkage information which measures the similarity between two connected documents.

We depict the graphical model of LtoRTM in Figure 4.2, which is a segment of the complete model consisting of only two connected virtual profiles. As indicated, it is a concatenation of two original LDA graphical plates, each of which represent one author virtual profile, connected by a binary variable indicator y^c_{didj}, which

represents the authority preference between author d_i and d_j in community c.

Similar to the original LDA, each author virtual profile is represented by a plate, in which the shaded circle $\boldsymbol{w_d}$ is the observed data, representing each position-based word appearing in the profile, and the un-shaded circle $\boldsymbol{z}$ is the random variable representing the topic assignment for one particular word. θ_d is a multinomial random variable, indicating the distribution of author virtual profile d over topics. β is global multinomial random variable, indicating the topic distribution over words in the whole corpus. Suppose that W, D, T are the number of distinct word (word vocabulary), the number of author virtual profiles and the number of topics respectively. We can represent θ as a $D \times T$ matrix, where each row represents one θ_d. Similarity, β can be represented as a $T \times W$ matrix. There also exists a T dimensional Dirichlet prior hyper-parameter α, which determines θ. Since our model is built upon the non-smoothed LDA, we do not introduce the Dirichlet prior for β. Additional details of the model parameters are illustrated in Table 4.1.

Given a collection of author virtual profiles, one essential target of our topic modeling is to discover the semantically coherent clusters of words (known as topics) to represent the profiles. Until now, we have introduced the model that can fulfill the task. Moreover, in order to model the authority preference over topics between author profiles, we further introduce a binary variable indicator $y^c_{d_i d_j}$, named as the **binary preference indicator**, to indicate the authority preference between author d_i and d_j. We have $y^c_{d_i d_j} = 1$ if author d_i is believed to be more prestigious than author d_j in domain (community) c. This binary indicator is distributed according to a distribution that depends on the topic assignments for the two participating author profiles, and a domain (community)-specific regression parameter η^c.

The generative process of this model is divided into two periods, and can be described as follows:

- Stage 1: For each author virtual profile d_i:
  - Draw the topic proportion $\theta_{d_i}|\alpha \sim Dir(\alpha)$
  - For each word at position n in profile d_i: $w_{d_i,n}$
    - Draw the topic assignment $z_{d_i,n}|\theta_{d_i} \sim Multi(\theta_{d_i})$

* Draw word $w_{d_i,n}|z_{d_i,n},\beta \sim Multi(\beta_{z_{d_i,n}})$

- Stage 2: For each pair of author profiles d_i and d_j with known preference:
  - Draw the binary preference indicator, satisfying:

$$y^c_{d_i,d_j}|\boldsymbol{z_{d_i}}, \boldsymbol{z_{d_j}} \sim \psi(\cdot|\boldsymbol{z_{d_i}}, \boldsymbol{z_{d_j}}, \eta^c) \tag{4.1}$$

where, $\boldsymbol{z_{d_i}} = z_{d_i,1}, z_{d_i,2}, \ldots, z_{d_i,n}$.

To note that $\boldsymbol{z_{d_i}}$ can be represented as a matrix, where each $z_{d_i,n}$ is a vector with only one element set to be 1 and the other elements set to be 0. It indicates the specific topic assignment for the n^{th} word $w_{d_i,n}$ in author profile d_i.

ψ represents the distribution function that $y^c_{d_i d_j}$ depends on. In order to model the difference in terms of authors' authority over topics, we assume that $y^c_{d_i d_j}$ depends on the difference between $\boldsymbol{z_{d_i}}$ and $\boldsymbol{z_{d_j}}$. In addition, since it is a binary indicator, we suppose that it follows the Bernoulli distribution, in which:

$$\begin{aligned} & y^c_{di,dj}|\boldsymbol{z}_{d_i}, \boldsymbol{z}_{d_j}, \eta^c, \upsilon^c \\ \sim \;& Bernoulli(\sigma(\eta^T_c(\overline{\boldsymbol{z}}_{d_i} - \overline{\boldsymbol{z}}_{d_j}) + \upsilon_c)) \end{aligned}$$

in which, $\sigma(\cdot)$ is the sigmoid function. This function models each per-pair binary variable y^c_{didj} as a logistic regression with hidden co-variates, parametrized by coefficient η^c and the intercept υ^c. We further represent the original matrix $\boldsymbol{z_{d_i}}$ as a T dimensional vector $\overline{\boldsymbol{z}}_{d_i}$, where $\overline{\boldsymbol{z}}_{d_i} = \frac{1}{N_{d_i}} \sum_{n=1}^{n=N_{d_i}} z_{di,n}$.

4.2.2 Incorporating Features

In the model we introduced in Section 4.2.1, authors' different preferences over topics are only determined by their associated contextual information, i.e., the papers they have published. As we can see from the generative process of the model, the binary preference indicator only depends on authors' topic assignments which are derived from author profiles. However, to measure an author's authority is a complicated process, as authors' expertise is not only determined by the papers they have written,

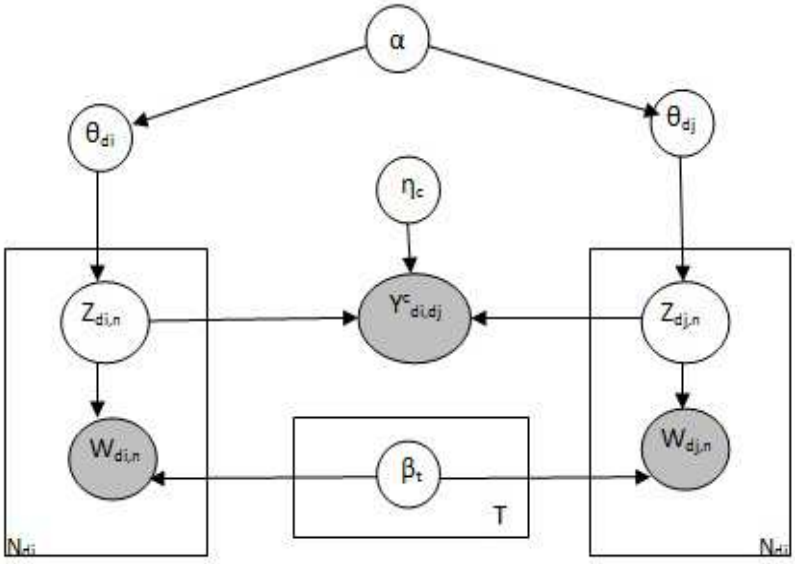

Figure 4.2: Graphical Model for LtoRTM

but also by several other factors, such as their collaboration with other researchers, the influence of their published works, and some temporal characteristics of the authors, such as, how many years have they devoted into research, how frequently do they publish, etc. To better model how authors' authority is differentiated, we extend the **LtoRTM** model by introducing an additional factor representing features.

LtoRTM with features

We depict the extended graphical model of LtoRTM in Figure 4.3. We name it as the $LtoRTMF$ model. As indicated, we represent each author A by an oval, in which, the author's virtual profile generated by the concatenation of his/her publications is still represented by a plate. In addition to that, we introduce a shaded circle $\boldsymbol{F_{d_i}}$ to represent the features associated with this author. Features are assumed to be observed data. Under this scheme, the authority preference between author A_i and A_j is not only determined by the topic assignments of their virtual contextual profiles, but jointly determined by both the content information

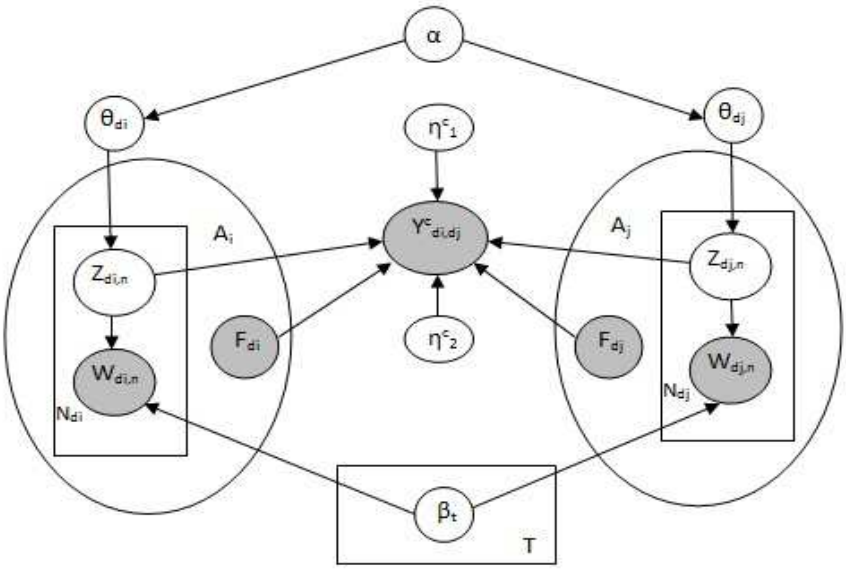

Figure 4.3: Graphical Model for LtoRTMF

and additional features. Correspondingly, we introduce two coefficients: η_1^c, a T dimensional vector, which is the regression parameter for topic assignment $\boldsymbol{z}$, and η_2^c which is the regression parameter for feature set. The size of η_2^c would be determined by the number of features we identify. Now, the binary preference indicator y_{didj}^c would be determined by following the distribution as:

$$\begin{aligned} & y_{di,dj}^c | \boldsymbol{z}_{d_i}, \boldsymbol{z}_{d_j}, \boldsymbol{f}_{d_i}, \boldsymbol{f}_{d_j}, \eta_1^c, \eta_2^c \\ & \sim \quad Bernoulli(\sigma(\eta_{c1}^T(\overline{\boldsymbol{z}}_{d_i} - \overline{\boldsymbol{z}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j}) + \upsilon^c)) \end{aligned}$$

Features

To represent authors (researchers)' authority, we identify several groups of features, each of which measures the expertise of an author from one aspect. Generally speaking, the features we consider reflect the overall expertise of an author (e.g., the total number of publications of an author) as well as his/her expertise in a specific domain or community (e.g., the author's number of publications in one domain). The whole feature set can be divided into four groups: 1) content profile

based features; 2) simple bibliographic based features; 3) network based features; and 4) temporal features.

Content profile based features: Even though we directly model the contextual virtual profile of an author by discovering its coherent clusters of words and representing it by a distribution over topics, we are also interested in measuring the content profiles by other widely-used IR metrics. Here we compute the traditional BM25 score of each author virtual profile, as well as the relevance score using standard language models. Both of these features are domain-based.

Simple bibliographic based features: We adopt a set of simple bibliographic features. These include:

total publication number (totalPubNo): which indicates the total number of publications of one author, across different research domains.

total citation number (totalCitNo): which indicates the total number of citations an author received from other papers published in different domains.

H-index[70]: H-index is the most well-known measurement in evaluating a researcher's expertise. A researchers is said to have an H-index with size h if h of his or her total papers have at least h citations each. This index is affected by the number of citations that a researcher has and the citation distribution among a researcher's various papers.

G-index[47]: G-index is another popular instrument. The G-index value is the highest integer (g) such that all the papers ranked in Position 1 to g in terms of their citation number have a combined number of citations of at least g^2.

Rational H-index distance (HD-index)[148]: this variant of H-index calculates the number of citations that are needed to increase the H-index by 1 point.

Rational H-index X (HX-index)[148]: the original H-index indicates the largest number of papers an author has with at least h citations. However, a researcher may have more than h papers, for example, n papers, that have at least h citations. If we define $x = n-h$, then the HX-index is calculated by $HX = h+x(s-h)$, where s is the total number of publications an author has.

E-index[202]: the original H-index only concentrates on the set of papers an author published, each of which has at least h citations. This set of papers is often referred to as the *h-core* papers of an author. By using this measurement, the only citation information that can be retrieved is h^2, i.e., at least h^2 citations of an author can be received. However, the additional citations for papers is the $h-core$ would be completely ignored. To complement the H-index for the ignored excess citations, $E-index$ is proposed, which can be computed by $e^2 = \sum_{j=1}^{h}(cit_j - h) = \sum_{j=1}^{h} cit_j - h^2$, where cit_j are the citations received by the j^{th} paper in the $h-core$ set. We can further have $E-index = sqrt(e^2)$.

Individual H-index IH-index[13]: this measurement is proposed to reduce the effects of co-authorship. It can be computed by dividing the standard H-index by the average number of authors in the *h-core* set: IH-index= h^2/N_a^T, N_a^T is the total number of authors in *h-core* set.

Normalized Individual H-index NIH-index[65]: this measurement is also proposed to reduce the coauthor's effect, but is much finer-grained than the previous one. To compute it, we can firstly normalize the number of citations for each paper in the *h-core* by dividing the number of its citation by its number of authors. Then we compute the H-index score based on these normalized citation counts.

It is noticeable to mention that we calculate all the features mentioned above from all its publications, as well as only those publications from a specific research domain. For example, we can compute the overall H-index of an author, by doing that, all the papers written by that author would be considered. However, when computing the H-index of an author in a specific domain c, we would only consider those papers published in that domain, and compute its citations only based on other papers that are also from that domain.

Network based features: this group of features measures how well an author collaborates with other authors, and how their publications influence other authors. We construct two types of networks, and apply the PageRank algorithm to compute the authors' authority scores. The networks we considered are:

Coauthor Network: this network is generated by connecting authors by their

coauthor-relationships. For the sake of PageRank algorithm, we convert each undirected edge into two directional edges. As a result, one non-weighted edge would exist from author a_i to author a_j and from author a_j to author a_i if they have written at least one paper together.

Citation Network: this directed network is generated by connecting authors by their citations. One non-weighted edge would point from author a_i to a_j if at least one publication of author a_i cites one paper of author a_j.

We also generate such two kinds of networks for each research community we considered.

Temporal features: this group of features measures authors' authority by some temporal characteristics associated with them. These include:

CareerTime: this measures how long a researcher has devoted into academic research? We assume that the longer career time a researcher has, the higher authority he may have.

LastRestTime: this indicates how many years have passed since the last publication of a researcher. We assume that a long time rest without academic output will negatively affect a researcher's academic reputation.

PubInterval: this measures how many years on average would a researcher take between every two consecutive publications. We assume that more frequent publication indicates more active academic participation.

Citation Influence ratio: we define and consider one other temporal factor which tests the long time influence of a researcher's publication, and thus indirectly represents the influence of the researcher. We assume that if a paper continues to be cited a long time after its publication, it brings higher prestige to its author (e.g., the paper PageRank [132] is frequently and persistently cited by the following papers). To model this temporal factor, we first introduce a decay function to differentiate the weight between a pair of paper citations. If paper p_j published in year y_j cites another paper p_i published in year y_i $(y_j - y_i) \geq 0$, we define a probability as the *citation influence ratio* of paper p_j on p_i as: $CIR(p_{ji}) = \beta_1(1 - \beta_2^{y_j - y_i})$, where β_2 $(0 < \beta_2 < 1)$ is the decay base. We now define the *citation influence* between a pair

of authors as: $CI(a_{ji}) = \sum CIR(p_{ji})$, where p_j is any paper of author a_j, p_i is any paper of a_i, and p_j cites p_i.

Contemporary h-index CH-index[158]: this index adds an age-related weighting to each paper. The basic assumption is that the older the paper, the less the weight. The new citation count for each paper of an author can be computed as $S^c(i) = \gamma \times (Y(now) - Y(i) + 1)^{-\delta} \times |C(i)|$, where $Y(i)$ is the year when paper i is published, and $|C(i)|$ is the set of paper citing paper i. In computation, δ is often set to be 1, and γ is set to be 4. After computing this new citation count for each paper, we can compute the H-index as the standard one based on the new citation count of each paper.

AR-index[80]: it is also an age-weighted index. The citation count of each paper would be divided by the age of that paper, and then the AR-index is the square root of the sum of all the papers in the *h-core* of an author.

AWCR-index[65]: This is the basically the same with the AR-index, but it sums over the weighted citation count of all the papers of an author rather than only the papers in the *h-core* set.

AvgPubNo: this is computed by dividing the total publication number of an author by the $CareerTime$ of this author.

AvgCiteNo: this is computed by dividing the total number of citations of an author by his/her $CareerTime$.

These features are also computed either based on all publications across domains or on those domain-specific publications. Overall, we have identified 42 distinct features.

4.3 Model Estimation and Ranking Scheme

To solve the **LtoRTM** and **LtoRTMF** model, we need to conduct model inference and estimation. This includes the model inference for 1) topic assignment ($\boldsymbol{z}$), 2) θ (virtual-profile-topic distribution), and 3) β (the topic-word distribution), as well as the parameter estimations for 1) α (the Dirichlet prior) and 2) η^c (the regression

coefficient). Based on the variables and parameters learned from the training set, we also introduce how to achieve the topic assignment and topic proportions for test authors, and how to rank them.

4.3.1 Inference and Estimation

Given a collection of author virtual profiles $\boldsymbol{D}$, in order to solve the topic model as we proposed, we would like to find parameters α, β, η^c, that can maximize the (marginal) log likelihood of the data:

$$
\begin{aligned}
& l(\alpha, \beta, \eta^c) \\
&= \log(p(\boldsymbol{W}, \boldsymbol{Y}|\alpha, \beta, \eta^c)) \\
&= \log([\prod_{d:1\to D} p(\boldsymbol{w}|\alpha, \beta)][\prod_{(di,dj)\in \boldsymbol{E}} p(y_{ij}|\eta^c)]) \\
&= \log\left(\prod_{d=1}^{D} \int p(\theta|\alpha)(\prod_{n=1}^{Nd} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta)) d\theta \right. \\
&\times \left. \prod_{(di,dj)\in \boldsymbol{E}} \sum_{\overline{z}_{d_i}} \sum_{\overline{z}_{d_j}} p(y_{ij}|\overline{z}_{d_i}, \overline{z}_{d_j}, \eta^c) \right)
\end{aligned}
$$

where, we denote E as the set of pairs of author profiles with known preferences. In our model, we would only model those pairs of author profiles with explicitly known preferences.

However, to maximize such log likelihood is intractable due to the problematic coupling between θ and β, which is caused by the existing edges between θ, z and β. Even though exact inference is intractable, there exist a wide variety of approximate inference algorithms, including including variational inference [18], expectation propagation [124], and Markov chain Monte Carlo (MCMC) schemes. In our work, we take use of the variational inference for approximating the posterior inference, and apply this procedure in a variational EM algorithm for parameter estimation.

The basic idea of variational inference is to make use the Jensen's inequality to obtain an adjustable lower bound on the log likelihood. A simple way to obtain a tractable family of lower bounds is to consider simple modifications of the original graphical model in which some of the edges and nodes are removed, and the

resulting graphical model is endowed with free variational parameters as follows in equation 4.2:

$$q(\theta, \boldsymbol{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n) \tag{4.2}$$

where, γ and ϕ are two free variational parameters. γ is a Dirichlet parameter, which similar to θ, can be represented by a $D \times T$ matrix; and ϕ is a multinomial parameter, which similar to $\boldsymbol{z}$, can also be represented as of $D \times N \times T$ tensor, where D is the number of author profiles in corpus, N is the number of position-based word tokens, and T is the number of pre-defined topics. Note that, $E_q[z_{d,n}] = \phi_{d,n}$.

With γ and ϕ, and integrating over the two random variables θ and z, the log of the marginal probability can be represented as:

$$\begin{aligned} & log(p(w, y|\alpha, \beta, \eta^c)) \\ = \; & log(\int \sum_z p(w, y, \theta, z|\alpha, \beta, \eta^c) d\theta) \\ = \; & log(\int \sum_z \frac{p(w, y, \theta, z|\alpha, \beta, \eta^c) q(\theta, z)}{q(\theta, z)} d\theta) \end{aligned}$$

According to Jensen's inequality $log(E(a)) \geq E(log(a))$, we can further have:

$$\begin{aligned} & log(E_q[\frac{p(w, y, \theta, z|\alpha, \beta, \eta^c)}{q(\theta, z)}]) \\ \geq \; & E_q[log(\frac{p(w, y, \theta, z|\alpha, \beta, \eta)}{q(\theta, z)})] \\ = \; & E_q[log(p(w, y, \theta, z|\alpha, \beta, \eta^c))] - E_q[log(q(\theta, z))] \end{aligned}$$

This is the lower bound of the original log likelihood, and is the goal probability we need to maximize.

To denote $E_q[log(p(w, y, \theta, z|\alpha, \beta, \eta))] - E_q[log(q(\theta, z))]$ as $L(\gamma, \phi; \alpha, \beta, \eta)$, we can expand it as:

$$
\begin{aligned}
& L(\gamma, \phi; \alpha, \beta, \eta^c) \\
&= \sum_{(di,dj)\in \boldsymbol{E}} E_q[log(p(y_{ij}|\overline{z}_{d_i}, \overline{z}_{d_j}, \eta^c))] \\
&+ \sum_d E_q[log(p(\theta_d|\alpha))] + \sum_d \sum_z E_q[log(p(z_{dn}|\theta_d))] \\
&+ \sum_d \sum_z E_q[log(p(w_{dn}|z_{dn}, \beta))] \\
&- E_q[log(q(\theta|\gamma))] - E_q[log(q(z|\phi))]
\end{aligned}
$$

Each element on the right-hand side of the above equation can be further expanded. Due to space limit, here we only present the expansion of the first element, which represents the primary contribution of our model. The expansions of the other elements are the same with the original LDA model.

In our $LtoRTM$ model, y^c_{didj} follows the Bernoulli distribution, taking η^c, $\boldsymbol{z_{d_i}}$, $\boldsymbol{z_{d_j}}$ as parameters. In the extended $LtoRTMF$ model, it further depends on the feature set of authors: $\boldsymbol{f_{d_i}}$, $\boldsymbol{f_{d_j}}$.

By representing Bernoulli distribution as a generalized linear model, we can have in the $LtoRTM$ model, the probability:

$$
p(yij|\overline{\boldsymbol{z}}_{d_i}, \overline{\boldsymbol{z}}_{d_j}, \eta^c) = exp\{y\eta_c^T(\overline{\boldsymbol{z}}_{d_i} - \overline{\boldsymbol{z}}_{d_j}) - \log(1 + exp(\eta_c^T(\overline{\boldsymbol{z}}_{d_i} - \overline{\boldsymbol{z}}_{d_j})))\} \tag{4.3}
$$

and in the $LtoRTMF$ model:

$$
\begin{aligned}
& p(yij|\overline{\boldsymbol{z}}_{d_i}, \overline{\boldsymbol{z}}_{d_j}, \overline{\boldsymbol{f}}_{d_i}, \overline{\boldsymbol{f}}_{d_j}, \eta_{c1}, \eta_{c2}) \\
&= exp\{y(\eta_{c1}^T(\overline{\boldsymbol{z}}_{d_i} - \overline{\boldsymbol{z}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})) \\
&- \log(1 + exp(\eta_{c1}^T(\overline{\boldsymbol{z}}_{d_i} - \overline{\boldsymbol{z}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})))\}
\end{aligned}
$$

By taking log of the probability, and using first-order approximation to compute their expectations, we can finally have:

in the $LtoRTM$ model:

$$E[\log(p(yij|\overline{\boldsymbol{z}}_{d_i}, \overline{\boldsymbol{z}}_{d_j}, \eta^c))] = y\eta_c^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) - \log(1 + exp(\eta_c^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}))) \tag{4.4}$$

and in the $LtoRTMF$ model:

$$\begin{aligned} & E[\log(p(yij|\overline{\boldsymbol{z}}_{d_i}, \overline{\boldsymbol{z}}_{d_j}, \overline{\boldsymbol{f}}_{d_i}, \overline{\boldsymbol{f}}_{d_j}, \eta_{c1}, \eta_{c2}))] \\ = \; & y(\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})) - \\ & \log(1 + exp(\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j}))) \end{aligned}$$

We have until now expanded $L(\gamma, \phi; \alpha, \beta, \eta^c)$. We then show how to maximize L with respect to ϕ, γ, α, β and η^c.

Inferring ϕ

To maximize L with respect to ϕ, we can collect the terms associated ϕ. Since y^c_{didj} depends on the difference between $\boldsymbol{z_{d_i}}$ and $\boldsymbol{z_{d_j}}$, which have been represented by ϕ_{d_i} and ϕ_{d_j}, we need to take derivatives with respect to ϕ_{d_i} and ϕ_{d_j} respectively.

In the $LtoRTM$ model, we have

$$\begin{aligned} & \phi_{din} \\ \propto \; & \log\beta\cdot, w_{dn} + \Gamma(\gamma_d) - \mathbf{1}\Gamma(\mathbf{1}^T\gamma_d) \\ + \; & \sum_{(di,dj)\in\boldsymbol{E}} (\frac{y}{N_{d_i}}\eta_c^T - \frac{\eta_c^T}{N_{d_i}} \frac{exp\{\eta_c^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j})\}}{1 + exp\{\eta_c^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j})\}}) \end{aligned}$$

$$\begin{aligned} & \phi_{djn} \\ \propto \; & \log\beta\cdot, w_{dn} + \Gamma(\gamma_d) - \mathbf{1}\Gamma(\mathbf{1}^T\gamma_d) \\ - \; & \sum_{(di,dj)\in\boldsymbol{E}} (\frac{y}{N_{d_j}}\eta_c^T + \frac{\eta_c^T}{N_{d_j}} \frac{exp\{\eta_c^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j})\}}{1 + exp\{\eta_c^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j})\}}) \end{aligned}$$

where

$$\overline{\boldsymbol{\phi}}_{d_i} = \frac{1}{N_{d_i}} \sum_n \phi_{dn} \tag{4.5}$$

and in the $LtoRTMF$ model with additional features, we have:

$$\begin{aligned}
&\phi_{din} \\
&\propto \quad \log \beta \cdot, w_{dn} + \Gamma(\gamma_d) - \mathbf{1}\Gamma(\mathbf{1}^T \gamma_d) + \sum_{(di,dj)\in \boldsymbol{E}} \Bigg(\frac{y}{N_{d_i}} \eta_{c1}^T \\
&- \quad \frac{\eta_{c1}^T}{N_{d_i}} \frac{exp\{\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})\}}{1 + exp\{\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})\}}) \Bigg)
\end{aligned}$$

$$\begin{aligned}
&\phi_{djn} \\
&\propto \quad \log \beta \cdot, w_{dn} + \Gamma(\gamma_d) - \mathbf{1}\Gamma(\mathbf{1}^T \gamma_d) - \sum_{(di,dj)\in \boldsymbol{E}} \Bigg(\frac{y}{N_{d_j}} \eta_{c1}^T \\
&+ \quad \frac{\eta_{c1}^T}{N_{d_j}} \frac{exp\{\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})\}}{1 + exp\{\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})\}}) \Bigg)
\end{aligned}$$

Inferring η In the $LtoRTM$ model,

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \eta^c} \\
&= \quad \sum_{(di,dj)\in \boldsymbol{E}} (y(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) - (\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) \frac{exp\{\eta_c^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j})\}}{1 + exp\{\eta_c^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j})\}})
\end{aligned}$$

and in the $LtoRTMF$ model, where we consider two coefficients η_1^c and η_2^c, we have:

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \eta_{c1}} \\
&= \quad \sum_{(di,dj)\in \boldsymbol{E}} \Big(y(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) - \\
&\qquad (\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) \frac{exp\{\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})\}}{1 + exp\{\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})\}} \Big)
\end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial \eta_{c2}} = \sum_{(di,dj)\in \boldsymbol{E}} \Big(y(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j}) - (\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j}) \frac{exp\{\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})\}}{1 + exp\{\eta_{c1}^T(\overline{\boldsymbol{\phi}}_{d_i} - \overline{\boldsymbol{\phi}}_{d_j}) + \eta_{c2}^T(\overline{\boldsymbol{f}}_{d_i} - \overline{\boldsymbol{f}}_{d_j})\}} \Big)$$

We leave the updating rule for α, β and γ for readers' reference, since they would be the same as the original LDA model[18].

4.3.2 Ranking Scheme

In the training process, we have approximated the posterior distribution of γ (representing θ), ϕ (representing the the topic assignments $\boldsymbol{z_d}$), β, as well as α and η^c. In the testing phase, a set of new author virtual profiles would be given. The words in those profiles are the observed data, but we would not know the preference between every pair of the profiles. In the testing phase, the α, η^c and β variables would be regarded as the known parameters, as their value have been estimated during the training process. As a result, what we need to approximate for the new author profiles are 1) the topic assignments for their word tokens (the γ), and 2) the author-profile-topic assignments (the ϕ):

$$p(\gamma, \phi | D^{test}, \alpha, \beta, \eta^c) \tag{4.6}$$

We would leave the inference process as an excercise for the readers, as without incorporating the pair-wise preference information between author profiles, our model would retreat to the original LDA model [18].

After approximating the γ and the ϕ variables for author profiles in testing set, we can compute the authority score of each author (represented by his/her author profile d_i) and rank them by:

$$P(di|c) = \eta_c^T \overline{\boldsymbol{\phi}}_{d_i} \tag{4.7}$$

or, with additional features:

$$P(di|c) = \eta_{c1}^T \overline{\phi}_{d_i} + \eta_{c2}^T \boldsymbol{f_{d_i}} \tag{4.8}$$

4.4 Experimental Evaluation

To demonstrate the effectiveness of our LtoRTM and LtoRTMF model, we conducted experimental studies comparing them with several state-of-the-art topic models and learning-to-rank algorithms. Particularly, we apply our model to two applications, which evaluate the expertise of researchers from two aspects: the prediction of SIG-community award winners and the prediction of PC members of the main conference of several research communities.

4.4.1 Experiments Setup

Data Set

We conducted experiments on the ACM and ArnetMiner data set (see introdcution in Section 2.4). There are 172,890 papers, 170,897 authors and 2,197 venues in the ACM data set, and 1,558,415 papers, 795,385 authors and 6,010 venues in the ArnetMiner data set. Due to computational efficiency concern, for papers in each data set, we further filter out the stop words in paper content, and collect the words that appear more than 10 times in the entire corpus. We finally retrieve 43,748 and 107,576 distinct words in the ACM and ArnetMiner data sets respectively.

Research Domain Identification

To identify a research community, we first manually cluster papers into different domains, and further group their associated authors. We choose six research communities as our targeting communities (see Table 4.2). For each such research community, we collected and merged the Top 20 venues identified by the Microsoft

Table 4.2: Community, Query and Award Winners ground truth. Numbers out side of the parentheses or in the parentheses indicate the number of winners available in ACM and ArnetMiner data set respectively

Community	Corresponding Query	SIG award winners
		(1990-2009)
sigarch	hardware architecture	27(27)
sigsoft	software engineering	15(15)
sigkdd	data mining	7(7)
sigir	information retrieval	9(9)
sigcomm	network communication	18(18)
sigmod	database	18(18)

academic search[2] and ArnetMiner search engine[3] for that research community respectively. Papers that are published in those venues are considered to be domain-specific papers of that community, and the authors of these papers are considered to be the domain-specific authors of that community. We collect the domain-specific features based on the domains we identified.

4.4.2 Application

Task description and Ground Truth generation

Both LotRTM and LtoRTMF are especially designed for modeling author's authority (interests or influence).

In this work, we focus on two applications that are closely related to expert ranking: predicting future award winners of a specific research community (the ACM SIG community), and predicting PC members of a main conference in research domain. We choose these two applications for two reasons: 1) they evaluate the expertise of a researcher from two different points of view; 2) we can retrieve excellent objective ground truth for both of them, which can avoid human labeling, which is assumed to be biased and subjective.

[2]http://research.microsoft.com/en-us/projects/academic/
[3]http://arnetminer.org/

Award Winner Prediction: Each year, in many ACM SIG communities, some outstanding researchers will be granted an award in honor of his or her profound impact and numerous research contributions. For example, in 2012, Prof. Norbert Fuhr was granted the 'Salton Award' in 'SIGIR' community for his 'pioneering, sustained, and continuing contributions to the theoretical foundations of information retrieval and database systems'.

It would be an interesting research task to predict the future award winners given historical information. To be more specific, the task of predicting award winners can be described as: *Given a specific research community c, and all its historical award winners before year Y, can we successfully predict its award winner on year Y?*. Normally, only one researcher would be granted the award each year.

From the ACM SIG official web site, we selected six SIG communities (shown in Table 4.2), and collected their historical award winners from 1990 to 2009, out of which, 2000-2009 is the period of time that we intend to predict. We generate the corresponding query for each community based on the main research area of that community; for example, the query for SIGIR community is 'information retrieval'. We also check the generated queries with the 23 categories provided by Microsoft Academic search engine, and make sure that each query corresponds to one category. We set the number of topics to be 20 for this task.

Conference PC member Prediction: Working as a PC member of the main conference in a research community is an important indicator of a researcher's expertise. This task of PC member prediction can be described as *Given a conference (representing a research community c), and all its PC members before year Y, can we successfully predict its PC members on year Y?*

For three SIG communities (SIGKDD, SIGIR, SIGMOD), we choose one main conference for each of them as our targeting conference, and collected its PC members from its official website between 2000 and 2009. 2005-2009 is the period of time that we intend to predict. Table 4.3 shows the community, the chosen conferences, as well as the number of PC members (also in our data corpus) for that conference between 2000-2009. For this task, we set the number of topics to be 10.

Table 4.3: Community, Conference, and PC member ground truth

Cmnty. (Conf.)	Years				
KDD (kdd)	2000	2001	2002	2003	2004
	55(57)	74(78)	73(78)	113(116)	124(127)
	2005	2006	2007	2008	2009
	129(130)	178(184)	210(219)	235(241)	230(247)
IR (sigir)	2000	2001	2002	2003	2004
	78(81)	41(43)	189(197)	38(38)	33(33)
	2005	2006	2007	2008	2009
	24(24)	114(114)	352(367)	365(381)	569(590)
MOD (sigmod)	2000	2001	2002	2003	2004
	14(14)	52(52)	65(65)	102(103)	136(136)
	2005	2006	2007	2008	2009
	135(140)	42(44)	4(4)	126(128)	126(129)

Training and Testing set generation

Both the training and testing sets are generated on per-community and per-year basis. Since we have few positive samples, as compared to a much larger set of negative samples, we pre-set a pos-neg ratio λ to randomly select negative samples. The process of generating the training set is as follow: suppose we intend to predict the award winner (or PC member) for community SIGKDD on year Y_i, we retrieve and regard all award winners (or PC members) of SIGKDD on year Y_j ($1990 \leq Y_j \leq Y_i - 1$) as positive samples, and for each positive sample, we randomly choose λ times other authors which are not SIGKDD award winners (or PC members) of on year Y_j. Such a process would be repeated 100 times, and all positive and negative samples would then form the training set of community SIGKDD on year Y_i. λ can be a tuned parameter, and in our current experiments, we set it to be 2.

For generating the testing set, for each community c on year Y_i, we retrieve the Top 1000 authors in terms of their in-domain(c) publication number as the testing set. We have also tried to generate the testing set by retrieving the Top 1000 authors in terms of their BM25 scores or a pool list of the merged Top 200 authors across all features, however, working on testing samples retrieved by their in-domain

publication number gives the best performance.

Baseline Algorithms

We choose two widely used learning-to-rank algorithms RankSVM and AdaRank, and one state-of-the-art topic models sLDA as our comparison baseline algorithms.

RankSVM (rSVM) [81] is a pair-wise learning-to-rank algorithm, which is designed to maximize the margin between positively and negatively labeled documents in the training set by minimizing the number of discordant pairs. Its learning task can be defined as the following quadratic programming problem.

$$\min_{\omega,\xi_{q,i,j}} \frac{1}{2}\|\omega\|^2 + c\sum_{q,i,j}\xi_{q,i,j} \qquad \textit{subject to}$$
$$\omega^T X_i^q \geq \omega^T X_j^q + 1 - \xi_{q,i,j},$$
$$\forall X_i^q \succ X_j^q, \xi_{q,i,j} \geq 0$$

where X_i^q represents the query-document feature vectors for document i. $X_i^q \succ X_j^q$ implies that document i is ranked higher than document X_j^q with respect to query q in the training set. $\xi_{q,i,j}$ denotes the non-negative slack variable. c is the parameter determining the trade-off between the training error and margin size. $\|\omega\|^2$ represents the structural loss.

AdaRank [186] is a list-wise learning-to-rank algorithm. Instead of training ranking models by minimizing the loss function loosely related to the performance measures (e.g., minimizing classification error on instance pairs), AdaRank is proposed to minimize the loss function directly defined on the performance measures (i.e., MAP, MRR, NDCG) by repeatedly constructing 'weak rankers' on the basis of re-weighted training data, and finally linearly combines the learned weak rankers to make predictions over testing data.

Supervised LDA[17] extends the original LDA model by adding a response variable connected to each document. Its ultimate goal, correspondingly, is to infer the latent topic structure of an unlabeled document, and then generate a prediction of its response. Supervised LDA is especially designed for applications like predicting

the ratings of movie reviews and the category of a document. Even though it is also a supervised learning algorithm, it does not explore the difference between every pair of documents. The response is only determined by the topic assignment of individual document.

For all three baselines, we feed them the same training data and testing data as we generated for running our LtoRTM and LtoRTMF model. We choose the average rank ($avgRank$) and MAP as the evaluation metric for predicting award winners and PC members respectively.

Prediction Results

We report the predicting results for award winners and PC members as compared with the baseline algorithms in both ACM and ArnetMiner data sets respectively in Table 4.4 to Table 4.7. We chose to use 'avgRank' as the evaluation metric for predicting award winners, and MAP as the metric for PC member prediction. 'avgRank' is defined to be the average rank of all winners in the testing community. Prediction results are reported for each community as well as the overall average rank across communities.

Predicting Award winners We test on RankSVM with pure content as well as additional features. For sLDA, we only work on word count features. AdaRank applies a different learning mechanism, where we took each of the 42 distinct features as one 'weak learner'. Several observations can be made from the results in Table2 4.4 and 4.5: 1) RankSVM still performs the best in terms of overall performance, however, this is not always true looking at individual communities. For example, our LtoRTM model can achieve better results than RankSVM for the 'sigsoft' community. 2) LtoRTM works better than AdaRank and sLDA in terms of overall performance under most circumstances (except the sigmod community on ArnetMiner data set, where AdaRank works the best); 3) incorporating additional features does not guarantee improved performance on individual communities. This is true not only for our LtoRTM vs LtoRTMF model, but also for RankSVM. However, we always obtain improved overall performance with additional features. 4)

we can achieve similar results on both data sets.

Table 4.4: Award winner prediction: ACM avgRank

Algorithm	arch	soft	kdd	ir	comm	mod	Overall
rSVM (C)	**35.0**	123.7	120.0	6.7	80.3	**49.3**	75.22
rSVM (C+F)	41.4	121.1	119.0	**5.7**	48.6	49.7	**70.03**
AdaRank	43.7	201.1	161.0	36.7	113.2	78.6	113.19
sLDA (C)	137.7	126.2	98.5	42.3	**35.8**	129.4	104.5
LtoRTM	108.2	**95.7**	82.6	22.3	109.8	136.0	97.05
LtoRTMF	120.0	101.0	**81.7**	24.8	98.2	87.4	90.86

Table 4.5: Award winner prediction: ArnetMiner avgRank

Algorithm	arch	soft	kdd	ir	comm	mod	Overall
rSVM (C)	**37.0**	122	138.0	**5.7**	**46.0**	49.7	69.67
rSVM (C+F)	69.3	**56.3**	67.1	97.8	109.7	39.2	**63.89**
AdaRank	194.8	127.4	63.9	22.4	52.2	65.7	96.35
sLDA (C)	99.7	105.9	105.3	166.0	149.4	108.9	115.12
LtoRTM	141.9	76.2	**47.8**	117.3	91.4	128.4	103.31
LtoRTMF	118.5	74.9	48.2	138.9	204.4	**34.0**	91.21

Predicting PC members Results on predicting PC members are reported in Tables 4.6 and 4.7 for ACM data set and ArnetMiner data set respectively. For the ACM data set, we can see that RankSVM still works the best; Our LtoRTM model outperforms AdaRank and shows competitive results with sLDA. For the ArnetMiner data set, however, our LtoRTMF model can outperform that of RankSVM

Table 4.6: PC member prediction: ACM MAP

Algorithm	sigkdd	sigir	sigmod	Overall
RankSVM (C)	0.5966	**0.5952**	**0.2303**	0.4740
RankSVM (C+F)	**0.6110**	0.5942	0.2267	**0.4773**
AdaRank	0.5997	0.2168	0.0261	0.2808
sLDA (C)	0.3358	0.4150	0.1814	0.3107
LtoRTM	0.3201	0.5146	0.0958	0.3102
LtoRTMF	0.4909	0.3372	0.1738	0.3340

Table 4.7: PC member prediction: ArnetMiner MAP

Algorithm	sigkdd	sigir	sigmod	Overall
RankSVM (C)	0.0692	0.0590	0.0479	0.0586
RankSVM (C+F)	0.0742	0.0632	**0.0513**	0.0629
AdaRank	0.1075	0.0411	0.0130	0.0539
sLDA (C)	0.0489	0.0809	0.0418	0.0571
LtoRTM	0.0496	**0.0821**	0.0424	0.0580
LtoRTMF	**0.1200**	0.0545	0.0393	**0.0712**

with additional features. We observe that performance varies across different communities. Incorporating features can provide performance improvement for some communities (sigkdd and sigmod on the ACM data set; sigmod on the ArnetMiner data set), but not all communities.

Feature Analysis

In LtoRTMF model, η_2^c is the coefficient vector associated with the feature vector. By checking the coefficient value associated with each feature, we can determine its contribution(importance) to the overall performance. Figures 4.4 and 4.5 illustrate the results for predicting award winners and PC members for the SIGKDD community on ACM data set respectively. In both of these figures, we use different colors to represent features' importance. Compared with the right-side indicator bar, colors more closer to '0' indicate less important features. 'Red' colors indicate positive correlations, and 'Blue' colors indicate 'negative' correlations.

We can observe that most of the features perform consistently across different years. Some features (i.e., feature #4: overall average citation number) keep on contributing positively, while others contribute (in-domain pub-interval (#40)) negatively. in-domain avgPubNo (#24), in-domain avgCiteNo (#25), and in-domain citation-network based PageRank (#26) are the three most important features in award winner prediction. Similar trends can be observed in Figure 4.4, where features show even more consistent performance than in award winner predictions.

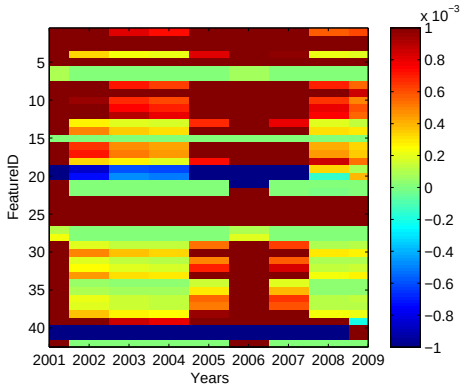

Figure 4.4: Feature Analysis (SIGKDD 2009) for award winner prediction

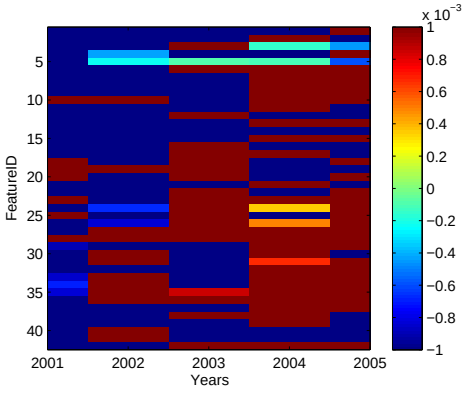

Figure 4.5: Feature Analysis (SIGKDD 2009) for PC member prediction

4.4.3 Qualitative Topic Modeling Results

We are also interested in evaluating the ability of our model in discovering latent topics in the author profile collections. Based on the learned results from the training set of predicting 2009 award winners for the sigir community(working on ACM data set), we generally retrieve the Top 10 returned words for two identified topics, and compare them with the results obtained from the original LDA.

As shown in Table 4.8, we intend to retrieve more coherent topic-related words. For example, for topic 'information retrieval', we can identify words like 'search', 'terms', which are relevant words but not ranked with Top 10 using LDA. On topic

Table 4.8: Topic Modeling Results

LDA	LtoRTM	LDA	LtoRTM
Topics: Information retrieval		Topics: Hardware	
information	information	design	hardware
retrieval	retrieval	hardware	circuit
systems	query	level	circuits
query	document	architecture	delay
based	language	processor	architecture
model	model	paper	processor
document	text	data	routing
database	search	computer	bounds
language	terms	based	clock

'hardware', we can retrieve some relevant words as 'circuit' and 'clock'.

Perplexity [33] is a standard measure to estimate the performance of topic modeling. Lower perplexity score indicates better generalization performance. Given a set of test words, perplexity can be defined as the exponential of the negative normalized predictive likelihood as follows:

$$P(d_i^{test}|\theta,\beta) = \prod_{w=1}^{V}(\sum_{z=1}^{K}\theta_{iz}\beta_{zw})^{s_{iw}^{test}} \tag{4.9}$$

$$Perplexity = \exp -\frac{\sum_{i=1}^{M^{test}}\log(P(d_i^{test}|\theta,\beta))}{\sum_{i=1}^{M^{test}}N_i^{test}} \tag{4.10}$$

where, M^{test} is the number of author profiles in testing set, and N_i^{test} is the number of words in profile d_i^{test}. s_{iw}^{test} indicates the word frequency of word w in testing profile i.

In order to test the generalization performance of our topic model, we vary the number of topics from 10 to 50, and compute the perplexity score for SIGKDD community on predicting award winners for year 2009 and 2006 on ACM data set. We compared our performance with that of sLDA.

As shown in Figure 4.6, our LtoRTM model can achieve lower perplexity score, and therefore better generalization performance than sLDA for both year 2009 and 2006 under all different topic numbers.

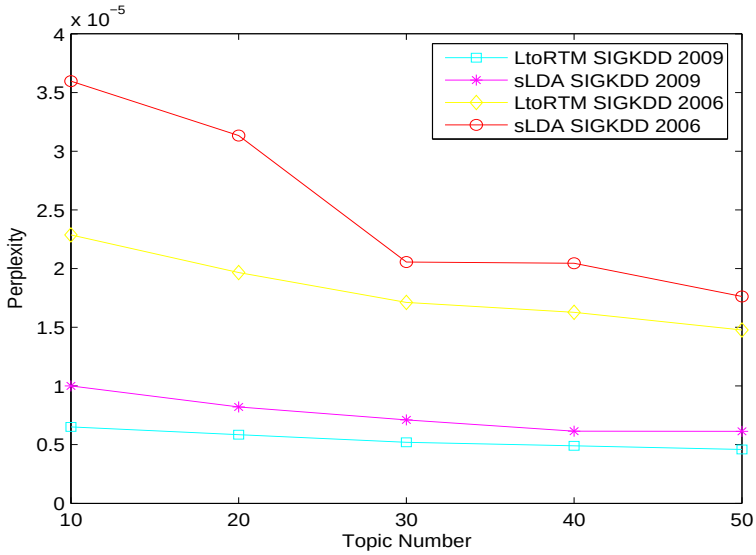

Figure 4.6: Perplexity

4.5 Bibliographic Notes

In this section, we review three lines of research work that are related to our work, and discuss the novelty of our work from them.

Topic Modeling Generative topic modeling has become a popular machine learning technique for topic-related content representations. Ever since the success of the original two representative topic models, pLSA[72] and LDA[18], which focus on pure content analysis by discovering the latent topics from large document collections, a large body of literature on topic models has been established, mostly by incorporating additional contextual information, such as time[16], authorship[147, 168, 173], geographical locations[195], or integrating linkage or social network information[28, 46, 128]. The linkage information being modeled, often represents the similarity between two linked documents, rather than the difference between documents, which is the focus of our work in this paper.

Blei and McAulliffe proposed a supervised LDA model[17] in 2010, which is a promising improvement over the original LDA, as it converts the topic modeling

approach, which is traditionally believed to be an unsupervised learning technique into a supervised one. Several other works[138, 208] have been proposed, following this direction. However, in these works, the labels are often attached to individual documents rather than every pair of documents to distinguish their different preference over topics. Our work, however, borrows the idea of pair-wise learning-to-rank into the topic modeling process.

Duan et al. proposed a ranking-based topic modeling[44], which utilizes the importance of documents and incorporates the TopicalPageRank[130] into topic modeling. Compared with our work, their documents' importance is not defined upon pairs of documents. Moreover, their model is built upon pLSA instead of LDA, and the model is designed for document clustering and classification applications, which are all different from our model.

Learning-to-Rank Learning-to-rank (LtoR for short)[104] is a recent trend of applying machine learning techniques to learn ranking functions automatically. In the standard LtoR setting, a typical training set is composed of queries, documents (represented by a feature set) and their associated labels. A machine learning algorithm would be employed to learn the ranking model, with the goal to predict the ground truth label in the training set as accurately as possible in terms of a loss function. In the test phase, when a new query comes in, the learned model is applied to rank the documents according to their relevance to the query. Depending on different hypotheses, input spaces, output spaces and loss functions, approaches to LtoR can be loosely grouped into three categories: point-wise, pairwise, and list-wise.

Expertise Ranking Expert ranking has been a promising research focus with the rapid development of on-line academic search engines, such as ArnetMiner and Microsoft Academic Search. Given a user query, the task of expert ranking basically involves identifying and ranking a list of researchers based on their expertise in that query-specific domain. Two categories of approaches have been focus of research in the past years: the pure content analysis based approach [8, 108, 50], which emphasizes evaluating authors' expertise by measuring the relevance between their associated documents and the query, and the social network based approach

[39, 170], which evaluates authors' expertise by exploiting the social interaction of authors and other scientific facets, such as their co-authorships, their citations to other papers/authors and more. Balog et al. [10] made a survey on the current main approaches for expertise retrieval, in which they more emphasized on summarizing the content-based approaches and divide them into probabilistic generative and discriminative model based approaches.

The topic modeling approach is one important group of probabilistic generative models for expert ranking. Typical works in this category include the models of CAT [173], ACT [168], ACTC [180], ALT [88]. However, none of them combine topic modeling with learning-to-rank approaches.

Fang et al. [50] proposed a probabilistic discriminative model for expert ranking, which is essentially a learning-to-rank method. Two other representative approaches using learning-to-rank for expert ranking include the work conducted by Moreira et al. [126] and the work done by MacDonald et al. [110], both of which applied several existing learning-to-rank algorithms for ranking experts (bloggers). None of these models integrate the advantage of topic modeling though, and the latter two are applications of existing algorithms.

4.6 Summary

In this chapter, we propose a novel topic model that incorporates the preference between pairs of authors in terms of their authority in a specific domain into topic modeling process. It borrows the essential idea of pair-wise learning-to-rank algorithms and is particularly designed for modeling authors' authority (interests or influence) in the academic environment. We further extend the model by introducing additional features related with authors' expertise beyond pure content. We provide introduction on model inference, parameter estimation, as well as the ranking scheme on new authors. Experiments conducted on two real world data sets have demonstrated our model to be either competitive or better than some state-of-the-art algorithms.

Chapter 5

Writing with style: venue classification and recommendation

In this chapter, we focus on the problem of publishing venue classification and recommendation, two tasks which have applications in the academic environment but are seldom investigated by previous research. Particular attention has been paid on discovering and making use of the stylometric features of publishing venues. For venue recommendation, an enhanced collaborative filtering method is proposed. Comprehensive experiments over real world data sets demonstrate the effectiveness of our methods.

5.1 Introduction

As early as the late nineteenth century, the research scientist T. C. Mendenhall conducted his pioneering studies in authorship attribution among Bacon, Marlowe, and Shakespeare. More than half a century later, another two scientists, Mosteller and Wallace, carried out their famous study on the mystery of the authorship of the Federalist papers [127]. They examined 146 political essays from the late eighteenth century, of which most are acknowledged to have been written by John Jay, Alexander Hamilton and James Madison; however, twelve of them are claimed to be

co-authored by Hamilton and Madison. By extracting function words as one of the most important stylometric features and making use of Bayesian statistical analysis, Mosteller and Wallace assigned all twelve disputed papers only to Madison.

These early studies initiated research in author attribution, also known as author verification or identification, and demonstrated that writing style is a key feature in distinguishing among authors. Today we not only have many more authors writing and publishing papers, but also have many different kinds of publications, covering different topics, with different genres and requiring different writing formats. In this chapter, we regard the publishing venues of all kinds of publications as venues. We have different venues for different research domains; for example, the 'SIGIR' conference for Information Retrieval (IR) research, and the 'VLDB' conference for database research. Moreover, even in one research domain, we also have multiple venues. To take the 'IR' research domain as an example, we have journals such as *Information Retrieval* and *J.ASIST*, as well as conferences, such as SIGIR, JCDL, WWW, CIKM and more. We also have posters, workshops, technical reports and patents. With so many different kinds of venues provided, a straightforward question may arise: how can they be distinguished from each other? Besides their topic-related differences, are they also distinguishable in writing styles?

A writing style, according to Karlgren [86], is a consistent and distinguishable tendency in making some linguistic choices. Compared to the content of a paper, writing style more reflects the preferences of authors in organizing sentences and choosing words. Even though no work has been carried out, to the best of our knowledge, investigating whether venues are also distinguishable by their writing styles, some brief statistical analysis can easily show that there exist obvious differences in terms of the probability distributions of papers published in different venues over stylometric features. In Figure 5.1 and Figure 5.2, we illustrate the distributions over two context-free features: the number of words and the number of sentences, for all papers we collect from the CiteSeer digital library that are published in two distinct venues: the JCDL venue and the WWW venue. We can observe distinguishable differences from both of the figures: papers published in JCDL have more diverse number of words and sentences than WWW papers,

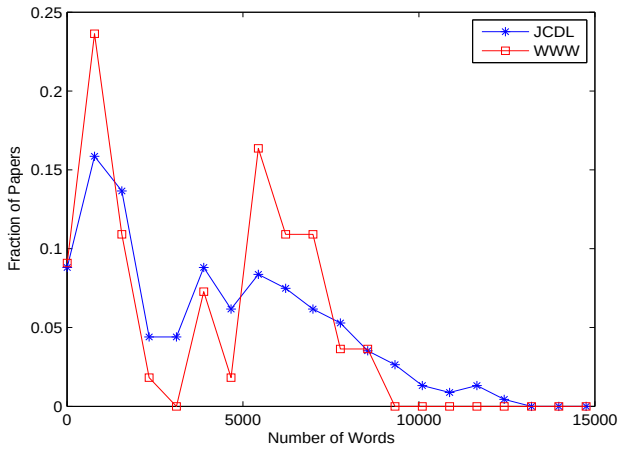

Figure 5.1: JCDL and WWW paper distribution over Number of Words

and more papers in WWW are written in fewer total number words and sentences. Such kind of observations further initiate the exploration into the task of classifying venues by their writing styles, a task that is actually equivalent to the question as whether the papers published in one specific venue share common characteristics in writing styles, and how are they distinguishable from papers published in other venues. We approach this problem by using a classification-based method.

Besides the task of venue classification, we are also interested in the work of venue recommendation or prediction, since some researchers, especially when they are new to a specific domain, may find it difficult to choose an appropriate place to submit their papers. There have been provided many recommendation systems, such as movie recommendation, merchandise recommendation, tag recommendation and citation recommendation; however, little work has been proposed for venue recommendation. We propose in this chapter a collaborative-filtering-based recommendation mechanism, in which we consider both content and writing style (stylometric) features of papers. Furthermore, we differentiate the importance of neighboring papers, those that are similar to the target paper, to better improve recommendation performance.

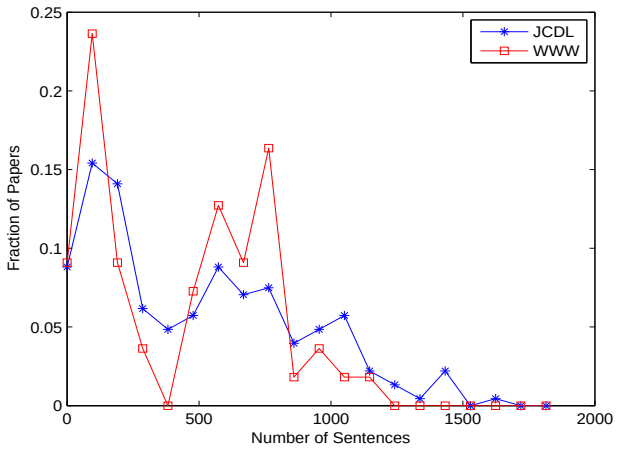

Figure 5.2: JCDL and WWW paper distribution over Number of Sentences

In this chapter, we first investigate the importance of writing styles in determining venue classification performance via comprehensive experimental studies, and then incorporate our observations into developing an automatic venue recommendation system. In summary, the main contributions we have made in this chapter include:

- the first exploration into classifying venues by their writing styles;
- a novel collaborative filtering based mechanism for automatic venue recommendation that have two distinctive characteristics: incorporating stylometric features to measure the similarity between papers, and differentiating the different contributions of neighboring nodes via tuning and optimization;
- empirical experimental studies which demonstrate the effectiveness of both venue classification and recommendation on two real-world data sets.

The rest of this chapter is organized as follows. We introduce the venue classification task and report our experimental results in section 5.2. We address the problem of venue prediction and provide the experimental results in section 5.3. Related work is reviewed in section 5.4. Section 5.5 concludes this chapter.

5.2 Venue Classification

5.2.1 Problem Identification

Given a set of papers, with their full or partial content provided, the task of venue classification is to determine the likelihood of a paper to be published in a particular venue. We can approach the task using traditional classification techniques, where a set of papers with known venue information are used for training, and the ultimate goal is to automatically determine the corresponding publishing venue of a paper whose venue information is missing. In particular, we are interested in exploring the following research questions:

- How well can venues be distinguishable from each other in terms of writing styles?
- What are the valuable features to represent writing styles?
- How sensitive is venue classification to classifier choice?
- Compared with using content-based features, can we improve classification results using stylometric features?
- Are topically-similar venues distinguishable by writing styles?
- Are venues of different genres distinguishable by writing styles?

5.2.2 Features

Since we focus on writing-style based venue classification, one of the main concerns is to define an appropriate quantitative text representation that captures the writing style of scientific papers. To avoid the influence from paper content, the features we employed need to be unrelated to topic and context-free. Based on previous studies and analyses in the task of author attribution, we incorporated three types of features into the feature set: lexical features, syntactic features and structural features. The entire set of features is listed in Table 5.1.

Lexical Features: Lexical features can be further divided into character-based or word-based features. It reflects a paper's preference for particular character or word usage. In our work, we included character-based features like number of terms, number of distinct terms, and more. The number of Hapax terms, one of the features we used, is defined to be the number of distinct terms that appear only once in the paper. We also used vocabulary richness as defined in [209]. In total, we have 66 lexical features.

Syntactic Features: Compared to lexical features, the discriminating power of syntactic features is derived from different formats and patterns in which sentences of a paper are organized. They are more likely to be content-independent. One of the most important syntactic features is the set of short yet all-purpose words, which are often referred to as function words, such as 'the', 'a', 'and', and 'to'. Research in author attribution demonstrated that function words play an important role in identifying authors, since their frequency of usage are often unaffected by papers' subjective topics. We adopted a set of 298 function words. Another example of a syntactic feature is punctuation. We count the sum of appearances of eight predefined punctuation symbols that appear in the paper.

Structural Features: Structural features represent the layout of a piece of writing. De Vel [37] introduced several structural features specifically for email. In our work, we adopted five structural features specifically for scientific papers: the number of sections, figures, equations, tables, and bibliographic references. Due to the fact that the original paper content available is in raw text format, in order to retrieve the number of figures in one specific paper, we simply count the number of times the word 'figure' or 'Figure' appears in the paper. We did the same for number of sections, number of tables and number of equations. We add number of references as an extra feature, not only because it is available in our data set, but also because this kind of feature is important for scientific papers. We can retrieve all of these five features for the papers in the CiteSeer data set, where the full paper content is available. For papers in the ACM data set, we can only retrieve the feature for the number of references.

In summary, we have 371 features for papers in the CiteSeer data set, and 367

features for papers in the ACM data set. The data sets are described below.

5.2.3 Experimental Evaluation

Data Collection

In order to test whether we can successfully classify venues by their writing styles, we perform experiments on two real world data sets: the ACM data set and the CiteSeer data set (see introduction in Section 2.4). The ACM data set consists of 172,890 papers, 170,897 authors, and 2,197 venues, and the CiteSeer data set consists of 510,231 papers, 48,797 authors and 65,441 venues.

For the CiteSeer data set, we further collected 119,727 papers published between 1949 and 2010 that have both abstracts and full content information, and 48,797 publising venues (out of 65,441) that have at least one paper with full content provided. We choose to use this smaller subset of the CiteSeer data set as our working data set.

Overall Classification Results

In the first analysis, we determine whether venues are distinguishable by their writing styles under general circumstances, regardless of content, topic and genre effects.

For all experiment settings, we make use of 10-fold cross validation, and adopt Accuracy and F_1 score, the two traditional classification metrics for performance evaluation.

Multi-Class Classification Results

To examine multi-class classification results, we randomly choose K venues, where K indicates the number of venues on which we tested. In our experiments, we change the value of K among 2, 5, 10, 30, 50, 100 and 150. For each value of K, we randomly choose K venues that have at least 100 papers for the ACM data set (at least 50 papers for the CiteSeer data set). We collect all the papers published in those chosen venues to construct the training/testing sets. The same process is

repeated ten times for each particular K, and the results are an average of all the iterations.

We construct RandomForest classifiers **Stylometric(A)** and **Stylometric(F)** for the CiteSeer data set, since we have both abstract and full content information for papers in this data set. Stylometric features are extracted from either abstract content or paper full content respectively. For the ACM data set where the full content of papers is missing, we work only on papers' abstracts to generate the stylometric features. Table 5.2 shows some brief statistics over the randomly chosen venues we tested. In order to demonstrate the effectiveness of the classification results, we further construct a **Baseline Classifier** for comparison, which randomly guesses the venue label for paper instances in the testing set.

As shown in Table 5.3 and Table 5.4, our stylometric classifier can outperform the baseline classifier under all circumstances. Based on the p value computed from the students' t test, all improvement over the Baseline classifier is statistically significant ($p \leq 0.05$), which confirms that venues are distinguishable by their writing styles. Moreover, there exists a tendency to achieve greater improvement over the random guessing baseline as the number of venues tested increased. Working on CiteSeer data with paper full content, there is a 70.25% improvement for 2-venue classification, and the performance is 7.45 times over random guessing for 30-venue and 8.86 times for 150-venue respectively. We also notice from the experiment results in CiteSeer data that we can achieve better performance working on the full paper content to retrieve the stylometric features than just from paper abstracts. The improvement is statistically significant when 30 or more venues are taken as testing venues.

Comparison of Classification Techniques

To evaluate the classification results of different classifiers, we repeat the same experimental process as described above using three state-of-the-art classifiers: RandomForest (RF), NaiveBayes (NB), and Support Vector Machines (SVM). For the CiteSeer data set, experiments were carried out for both paper abstract (A) and

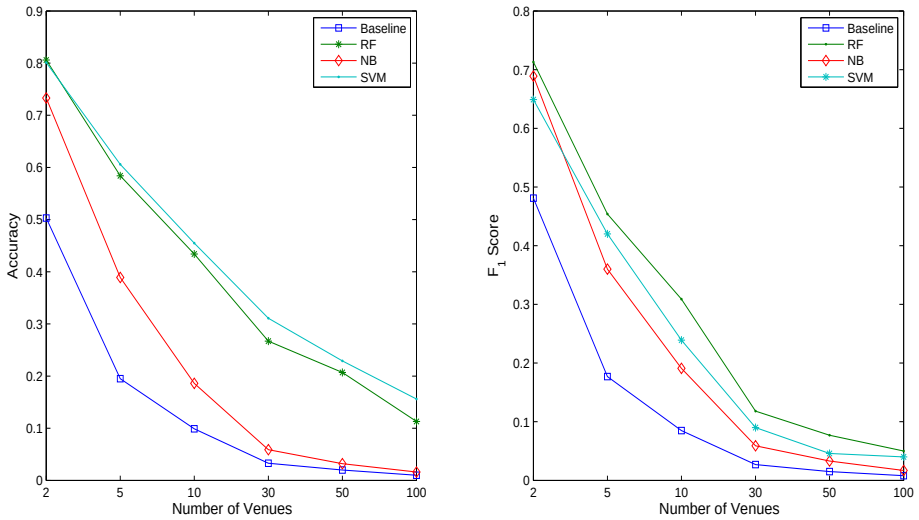

Figure 5.3: Comparison of Classifiers: Accuracy and F_1 Score for ACM data

full content (F) separately. We report experimental results in Figure 5.3 and Figure 5.4. We can see that all classifiers achieve better performance than random guessing; however, different classifiers have different impacts on the performance over the two data sets.

For ACM data set, RandomForest and SVM work better than NaiveBayes for both Accuracy and F_1 Score. SVM outperforms RandomForest in terms of Accuracy, however, RandomForest can achieve higher F_1 Score than SVM.

For CiteSeer data set, all three classifiers can achieve better performance working with paper full content than paper abstract. For both working with paper abstract and full content, RandomForest performs the best with small number of testing venues, and is then outperformed by SVM when the number of venues exceeds 30 and 50 respectively. NaiveBayes is the worst in general in terms of Accuracy, however, it gradually catches up with the performance of RandomForest and SVM when the number of venues tested is increased. In terms of F_1 Score, RandomForest is the best classifier working on both data sets. NaiveBayes shows comparable performance as RandomForest. SVM turns out to be the worst of the three, whose

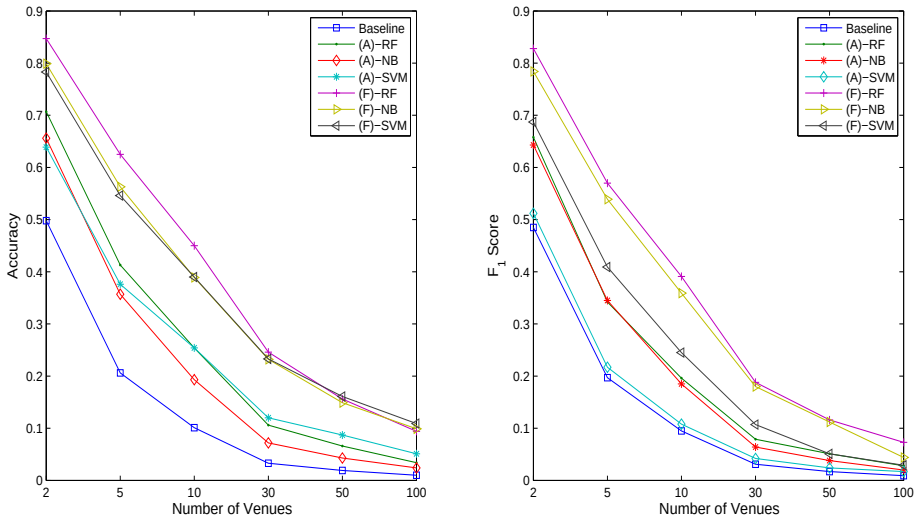

Figure 5.4: Comparison of Classifiers: Accuracy and F_1 Score for CiteSeer data

performance is only slightly better than random guessing when working on paper abstracts.

Contribution of Features

Comparison of feature types

As introduced in Section 5.2.2, we have three groups of stylometric features: lexical, syntactic and structural. To examine the contribution of different feature sets, we first test the performance on each individual group, and then add them one by one to test the changes in performance. We fix the number of venues tested to be 10. Performance in terms of Accuracy and F_1 Score are summarized in Tables 5.5 and 5.6 respectively.

We can see that lexical features still play the most important role in venue classification. Structural features are the least useful, probably due to our rough calculation method for collecting number of sections and number of figures. However, we can also find that each group of features contributes positively to the overall

performance, since when we add them together, performance is better than each individualy.

We further conducted five individual pairwise t tests in order to examine the significance of improvement. Table 5.7 shows the p value of the t tests for feature comparison for both ACM and CiteSeer data sets. Both lexical and syntactic features work significantly better than structural features. Combining lexical and syntactic features can provide significant improvement over pure lexical features, however, the improvement is not significant when we further add structural features. The results are consistent across the two data sets.

Contribution of individual features

To investigate the contribution of individual features, we adopted the leave-one-out scheme to test the classification performance when one targeted feature is not incorporated. The more the performance drops, the more positive contribution the targeted feature would make, and therefore, it would be more important.

Experiments were conducted for 5-classes (5-venues); RandomForest works as the classifier. The following Table 5.8 shows the ranked results on CiteSeer data set in terms of F_1 score and Accuracy respectively. The stylometric features are extracted from papers full content.

As indicated from the results, number of tables, number of refereces and number of word tokens are the three most important features in classifying venues by their writing styles.

Content vs. Writing Styles

Under all experimental settings in previous sections, we work on pure stylometric features. Besides the difference in writing styles, venues also differ in their content. In order to compare the classification performance between writing-style based features and topic/content based features, we further construct the RandomForest-based **Content Classifier**, in which we represent each paper by the TF-IDF scores of the Top 500 most frequent appearing terms in the whole corpus, and the **Combine Classifier**, where we combine both stylometric and content-based features.

As shown in Tables 5.9 and 5.11, the Content Classifier works better than the Stylometric Classifier. It indicates that topic-related difference is more distinguishable than writing styles for venues. When combining both stylometric and content features, the performance is not improved on the ACM data set; however, we can get improved performance on CiteSeer data set when features over full content are integrated.

Topics vs. Writing Styles

Working on CiteSeer data set, we randomly select 100 papers published in the venue 'SIGIR'. We would like to test whether papers in this venue can be successfully distinguished from papers published in other venues, either with more or less similarity with the venue 'SIGIR' in terms of venue topics. We select six other venues, and randomly select 100 papers for each of them. RandomForest is used as the classifier. Table 5.12 shows the result.

We can find that papers published in similar venues can also be successfully distinguished with high probability (e.g., 73% for papers in SIGIR and WWW) based on writing style features. There shows an increase in classification accuracy when venues are talking about different topics than similar topics.

Genres vs. Writing Styles

We are also interested in discovering the impact of different genres of venues on similar topics in terms of their writing styles. As we already know, there exist many different genres of venues even for the same topic. For example, the journal of SIGMOD Record compared with the conference of SIGMOD in database research domain. In this group of experiments, we collect papers published in journals and conferences, and show their classification results. RandomForest is used as the classifier. As shown in Table 5.10, we first test on the overall performance for all journals and conferences regardless of topic difference. For doing this, we randomly select 1000 journal venues and 1000 conferences venues, collect all their published papers, and carry out the classification. As indicated, we can retrieve an accuracy over 76%.

We further choose three different research domains; for each of them, we collected 100 papers published in their corresponding journal venues and conference venues respectively. Results show that in database and computer architecture domain, the classification results are better than that in the graphics domain. Even though we cannot determine exactly the effect of research topics on the classification results between journals and conferences, we can still see that on a general basis, these two are distinguishable.

Improving Classification Results

To further improve the accuracy of our classifier, two popular techniques, Boosting [151] and Bagging (Bootstrap aggregating) [22], have been adopted, both of which essentially construct a set of classifiers which are then combined to form a composite classifier. The composite classifier is generally believed to perform better than the individual classifiers.

We apply both Bagging and Adaboost, provided by WEKA[2], on both ACM and CiteSeer data sets. We experimented on different numbers of venues (2, 5, 10, 30 and 50). For venues in CiteSeer data set, we also test the performance by either using only paper abstract or full content respectively. RandomForest is used as the basic classifier, and the results are also evaluated using 10-fold cross validation. We report results in terms of accuracy and F_1 score in Figure 5.5 and and Figure 5.6.

Both Bagging and Boosting provide significant improvement over the original classification results. Bagging shows better ability in improving accuracy. The improvement increases when more venues are tested. Working on 10-venue task, the improvement of Bagging is 12.44% for ACM data set, 27.56% for CiteSeer abstract and 16.4% for CiteSeer full paper content. AdaBoost, however, works better for improving the performance in terms of F_1 Score: it improves performance by 10.36% for ACM, 10.71% for CiteSeer abstract and 15.09% for CiteSeer full paper content.

[2]http://www.cs.waikato.ac.nz/ml/weka/

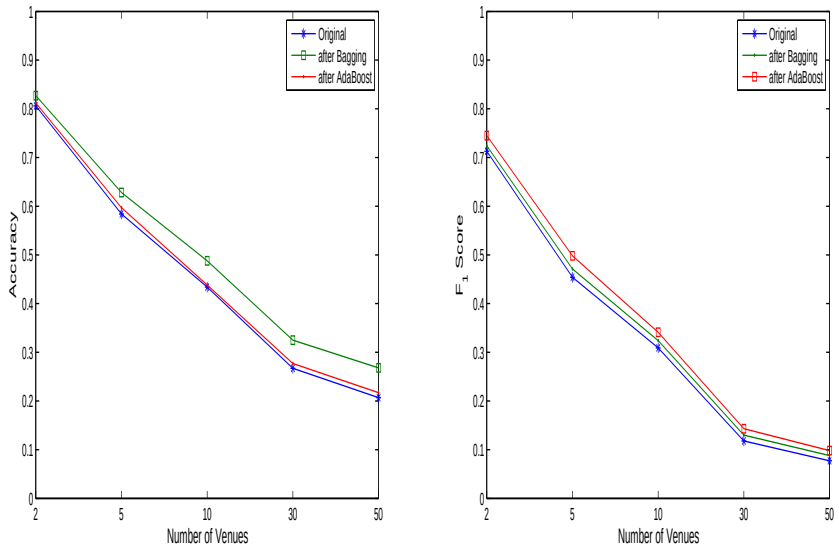

Figure 5.5: Bagging and Boosting: Accuracy and F_1 Score for ACM data

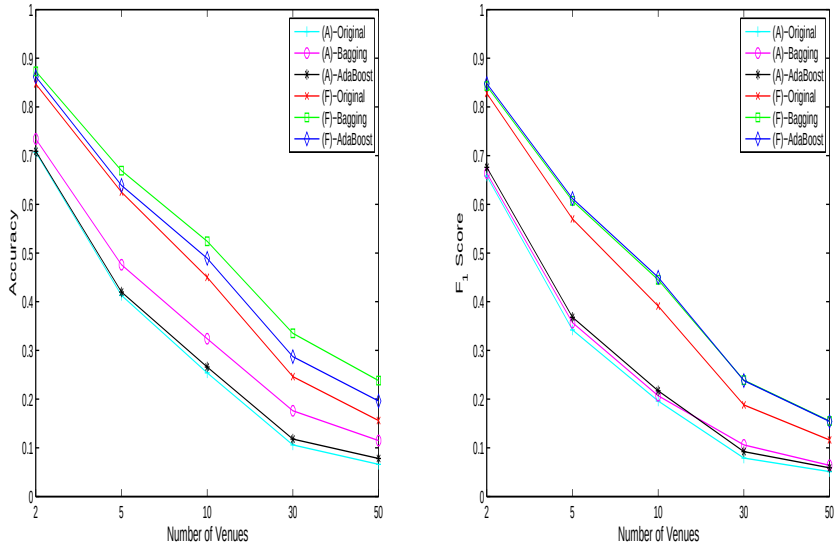

Figure 5.6: Bagging and Boosting: Accuracy and F_1 Score for CiteSeer data

5.3 Venue Recommendation

It is well understood that one of the fundamental tasks for most research scientists is to publish their work. However, many research scientisits occasionally have a difficult time in determining where to submit their papers. Even though some experienced researchers may have a target venue in mind before they finish their work, many others, especially new researchers in a domain, prefer to finish their papers first, and then to decide where to submit. Similarly, if the paper is completed after the deadline or not accepted at the target venue, another choice may be needed.

It is not a trivial task to make such a choice, however, due to the rapid growth in both the quantity and variety of publication venues in recent decades, making us have many different kinds of venues, with different topics and genres and requiring different writing formats.

Recommender systems have emerged as a good solution for helping people deal with the rapid growth and complexity of information. The technique was first introduced to generate suggestions (e.g., for movies and merchandise) to users, and then introduced in social network analysis and has been widely used in many applications, including tag recommendation, link recommendation, and citation recommendation. However, little effort has been employed to tackle the problem of venue recommendation, where given a paper, with its authors, content, and references provided, a list of venues are recommended for submission of this chapter.

A number of challenges arise in this task. First of all, the recommended venue should have a good match with the topics discussed in the paper. Venues have their own topic focus, as we have mentioned before, like information retrieval for SIGIR and databases for SIGMOD. Secondly, venues often have their specific writing format requirement. *As we have demonstrated in the task of venue classification, different venues do have their distinguishable writing styles*, an interesting question would therefore rise as whether papers with similar writing styles can more easily get accepted in similar venues. Finally, a good venue recommendation should match with the research profiles (e.g., historical venues) of the authors of the paper. We are interested in examining how the previous publication history of an author, along

with the relationship between the target paper and other papers will be useful to affect the recommendation results.

Collaborative Filtering (CF for short) is the predominant approach in current recommender systems. It can be further divided into memory-based CF and model-based CF. Memory-based CF is widely used due to its simplicity and efficiency, and it provides a good framework for venue recommendation as both papers' inter-similarity and inter-relationships can be incorporated for recommendation. In this work, we introduce the memory-based CF into venue recommendation, and particularly provide two extensions to the basic algorithm. For the first extension, we consider papers' similarity in terms of their writing style, an importance feature whose effectiveness has been desmstratated in our venue classification task; For the second extension, we divide the neighboring papers of the target paper into several groups, each of which represents a certain scientific relationship with the target paper. Contributions from each sub-group of neighbors can be differentiated and optimized.

5.3.1 Problem Identification

Let p be any given paper, and v be any candidate venue in the data corpus. The venue recommendation task can be defined as follows:

Given a paper p, what is the probability of it being published in venue v ?

It is essentially a ranking problem. Namely, we need to determine $p(v|p)$, to rank the candidate venues according to this probability, and to return the ranked list of candidate venues as recommendations of venues to which this chapter could be submitted.

In order to compute this probability, we adopt the basic idea of collaborative-filtering, and utilize other papers with known venues to predict or recommend venues for the target paper. Moreover, we make two extensions to the original traditional collaborative-filtering based approach: one is to incorporate stylometric features to better measure the similarity between papers; the other is to differentiate the

importance of those papers that share some similarity with the target paper, to further improve recommendation performance.

5.3.2 Approach

CF-based method

In a traditional user-item recommendation system, when the memory-based collaborative-filtering approach is used to predict the ratings of users over items, the user-item relationship is often represented as a two-dimensional matrix. Similarly, we can represent the relationship between papers and venues in a two-dimensional matrix, where the rows represent papers, and the columns represent venues. For each particular paper-venue pair (p, v), the corresponding entry on matrix represented as $I(p, v)$ indicates whether paper p is published in venue v.

We can apply the memory-based CF into our paper-venue matrix, with the underlying assumption that it would have a higher probability for a paper to get published in venues in which other similar papers have been published. However, the paper-venue matrix is different from the user-item matrix in that one paper can only be published in one venue, and thus it is unsuitable to use the item-based method, where the similarity between items (venues) rated (published) by the target user (paper) is going to be compared. We therefore choose to apply the user-based CF.

Formally, the process of applying user-based CF to the venue recommendation task can be described as follows.

- Given a target paper p_i, we first compute its similarity with all other papers in the data set, and collect the K most similar papers to target paper p_i. The collection of these top K papers is indicated as $S(p_i)$. K is a system parameter, and can be tuned via experiments.

- We collect all publishing venues of the papers in $S(p_i)$, and denote the collection as $V(p_i)$. For each venue v_j in collection $V(p_i)$, we predict the probability of having p_i published in v_j by computing $P(v_j|p_i)$ by

$$P(v_j|p_i) = \frac{\sum_{p_k \subset S(p_i)} s(p_i, p_k) I(p_k, v_j)}{\sum_{p_k \subset S(p_i)} I(p_k, v_j)} \tag{5.1}$$

where $s(p_i, p_k)$ is the similarity score between paper p_i and p_k, and $I(p_k, v_j)$ is an indicator function. We have: $I(p_k, v_j) = 1$, if p_k is published in v_j; otherwise, $I(p_k, v_j) = 0$.

- Rank all candidate venues in $V(p_i)$ by $P(v|p)$.

Extension 1: Stylometric Features

As indicated in Equation 5.1, one crutial component in this CF-based method for venue recommendation is the paper-paper similarity measurement. Dominant similarity measures in the traditional CF method include the Pearson Correlation Coefficient and Vector Space Cosine Similarity measurement. We make use of the latter method.

Papers differ in their content and topics. Moreover, papers as well as venues are also distinguishable by their writing styles. To better measure papers' similarity, we need to consider both the content and stylometric features. To represent papers' content, we take use of Mallet [115], which is an open source software implementing LDA [18], to retrieve the papers' content distribution over 100 topics; To capture the writing styles of papers, we made use of the identified over 300 distinct features in the task of venue classification. Table 5.1 indicates all the stylometric features we adopted, which can be grouped into three categories each of which measure a paper's writing style from lexical, syntactic and structural aspects.

Lexical features [127] reflect a paper's preference for particular character or word usage. Typical features within this category include number of distinct terms, number of alphabetic and digital characters, average sentence length, and more. Syntactic features [165], however, focus on extracting the different formats and patterns in which sentences of a paper are organized. The most representative syntactic features include function words, punctuation and part-of-speech tags. In our work, we make use of the first two syntactic features. Structural features [37] represent the layout

of a piece of writing. We adopt in our work five structural features specifically for scientific papers, including the number of sections, figures, tables, equations and references. The entire feature sets is presented in Table 5.1.

Extension 2: Neighbor Differentiation

Another crucial component in the memory-based CF model is to retrieve proper neighbors that share similarity with the target paper. Normally, this is done by finding the top K neighboring papers in terms of their cosine similarity score with the target paper. However, papers do not only differ in the value of the similarity scores, but also in their different relationships with the target paper. For example, given a paper, we can find other papers that are written by the same authors (authorship), papers that are cited by the target paper, and papers that share the same citations with the target paper (bibliographic coupling). All of these kinds of papers should play different roles in their influence on the target paper in selecting future venues in which to publish.

We divide the top K similar papers into four categories. The first category is called 'author-neighbors', which are papers written by at least one author in common with the target paper. The second category is referred to as 'reference neighbors', which are the papers that have been cited by the target paper. The third category is named as 'sibling neighbors', which are papers that have at least one common reference paper with the target paper. All other papers that share similarity with the target paper, yet do not fall into any of the three categories mentioned above are referred to as 'other neighbors'. Since we rely on the historical data for prediction or recommendation, for any given paper p which is finished in year y_1, and is to be predicted, we would only consider neighboring papers that have been published before y_1.

To differentiate their influence on the target paper, we introduce four parameters, each of which indicates the importance of neighbor papers of one category. To

compute $P(v_j|p_i)$, the updated CF model can then be indicated as:

$$P(v_j|p_i) = \sum_{c:1\rightarrow 4} \alpha_c \frac{\sum_{p_k \subset N_c(p_i)} s(p_i, p_k) I(p_k, v_j)}{\sum_{p_k \subset N_c(p_i) I(p_k, v_j)}} \tag{5.2}$$

where $N_c(p_i)$ $(1 \leq c \leq 4)$ indicates the four categories of neighbor papers of the target paper p_i. $\alpha_c \in [0, 1]$ is the parameter that needs to be tuned to reflect the influence of neighbor papers of category c.

5.3.3 Evaluation

Experimental Setup

We introduce in this section the experiments we carried out for the task of venue recommendation. In particular, we wish to explore the following questions:

- What would venue recommendation results be if we utilize stylometric features alone to measure paper similarity?
- Can we achieve improved performance if we combine both the content and stylometric features for paper similarity measurement?
- Which category of paper neighbors would play the most important role in helping to predict publishing venues?
- Under what combination of the four categories can the best recommendation performance be achieved?

We conducted experiments on the **ACM data set** and **CiteSeer data set**, the same two data sets as we conducted experiments for venue classification. We further select 35,020 papers published across 739 venues, each of which has at least 20 papers published in it, to serve as the experimental papers for the CiteSeer data set. We randomly choose 10,000 papers from ACM and CiteSeer data sets respectively as our target papers whose venues are to be predicted.

We identified three categories, and 25 different types of stylometric features. For papers in the CiteSeer data set, where the full content of papers is available in pure

text format, we can simply count the number of times the word 'figure' or 'Figure' appears in the paper to obtain the number of figures. We did the same for number of sections, number of tables and number of equations. Finally, we extracted 371 stylometric features for papers in the CiteSeer data set, and 367 features for papers in the ACM data set.

To test venue recommendation performance, we match the predicted venues with the real publication venues of the target papers. Two standard metrics: **Accuracy@N** (N varies among 5, 10, and 20) and **MRR** are adopted for evaluation.

Results Analysis: Stylometric Features

We first examine whether paper similarity based on stylometric features can lead to good recommendation performance. By doing this, we represent each paper by a vector composed of only the stylometric features of that paper, and compute papers' similarity based on those paper vectors.

For comparison, we construct paper vectors by only making use of their paper content information, that is, the paper's content distribution over 100 topics learned from LDA. We also combine both content and stylometric features to get merged features for paper similarity measurement. In all the experiments, we set the parameters $\alpha_c (1 \leq c \leq 4)$ to be 0.25.

We collect the top K most similar papers with known venues to predict the possible publishing venue of the target paper. K is a system parameter, whose value might affect the prediction performance. To examine its effect, and varied the value of K among 500, 1000, 2000, 5000, 10000. We also experimented with using all neighboring papers of the target paper. Experimental results for ACM and CiteSeer data sets are described in Table 5.13.

Several observations can be found from the results on the ACM data set. First of all, there is a significant improvement as we combine both stylometric and content-based features as compared to working on either stylometric or content-based features separately, whose performance is competitive with each other. The improvement is nearly or more than 50% when a subset of paper neighbors are considered,

and is 10.92% working on all paper neighbors in terms of Accuracy@5. Secondly, there is no obvious increase in terms of Accuracy@5, Accuracy@10 and Accuracy@20 as the value K (the top K most similar papers to the target paper) increases from 500 to 10000 working on either stylometric or content features seperately. However, we achieved consistent improvement on the average MRR value. When working on combined features, performance in terms of all metrics also obtained constant improvement. We achieve significant improvement when we collect all paper neighbors for consideration. The best performance is achieved when working on all neighbors with combined features. Over 55.72%, 69.81% and 78.32% papers can have their publishing venues be correctly predicted within Top 5, Top 10 and Top 20 results respectively.

We noticed consistent performance when working on the CiteSeer data set, where paper's full content is used for generating both content and stylometric features. Content-based features work better than stylometric features when a small set of top-returned paper neighbors are adopted; however, the performance on using stylometric features gradually outperform that of content-based features when more top-returned paper neighbors are considered. When combining both stylometric and content-based features, there is no improvement as compared to using pure content-based features, however, we observe improved performance for such a combination when more than 2000 top neighbors are considered. The best performance is also achieved when all paper neighbors and all features contribute, where 23.87%, 28.99% and 33.74% papers can have their venues correctly predicted within Top 5, Top 10 and Top 20.

Results analysis: Weights among neighbors, Parameter tuning

We expect that different categories of neighboring papers can have different contributions when making venue recommendations.

We gradually change the weight for each particular type of neighbors from 0 to 1, and let the other three kinds of neighboring papers share the remaining weight. Results are reported in Figure 5.9 and 5.10.

When the weight for a particular type is set to be 1, it actually indicates the individual contribution of that type of neighbors. We particularly show the individual contribution of each type of neighbor papers for ACM and CiteSeer data set in Figure 5.8 and 5.9. As indicated, author neighbors contribute the most in both data sets, while the other neighbors are less important. It indicates that when authors finish their work, they often submit the paper to those venues in which they have had a previous paper successfully published. This is on one hand due to researchers continuing to focus on similar or related topics, at least within similar research domains. On the other hand, authors will gain more reputation and thus confidence in certain venues, so that they are always willing to submit to those venues, and it also has higher probability to have their work accepted. Reference neighbors and Sibling neighbors are competitive with each other, which matches our initial expectation, as reference neighbors and sibling neighbors both are topic-related with the target paper.

We also notice from the results shown in Figure 5.9 and 5.10 that we need to incorporate all types of neighbors, since we can retrieve better performance when all four categories of neighboring papers contribute rather than giving any of them zero weight. Moreover, even though the author neighbors are the most important source of information, when giving extra weight to them, predictive performance decreased.

Results analysis: Weights among neighbors, Parameter Optimization

Parameter tuning, as we addressed in Section 5.3.3, tells us the different importance of different categories of neighboring papers. We are more interested, however, to find parameter settings that can give us the best recommendation performance. To implement that, we can apply parameter optimization approaches.

Given a paper p_i, which is the target paper, and any candidate venue v_j in the data set, we can compute the probability $P(v|p)$ based on formula (4). Suppose A_j, R_j, S_j and O_j represent the normalized accumulated similarity score between the target paper and author neighbors, reference neighbors, sibling neighbors, and other

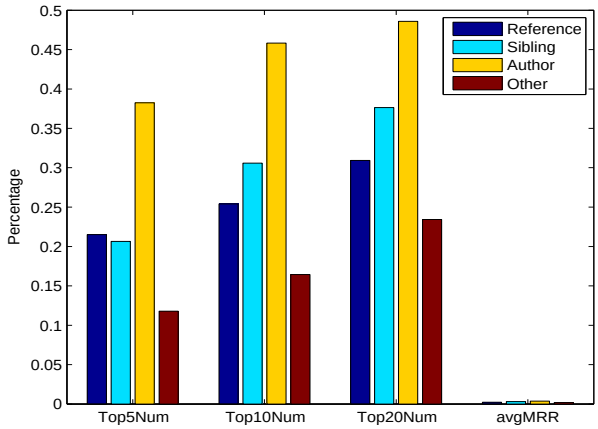

Figure 5.7: ACM data set: Individual contribution of types of neighbors

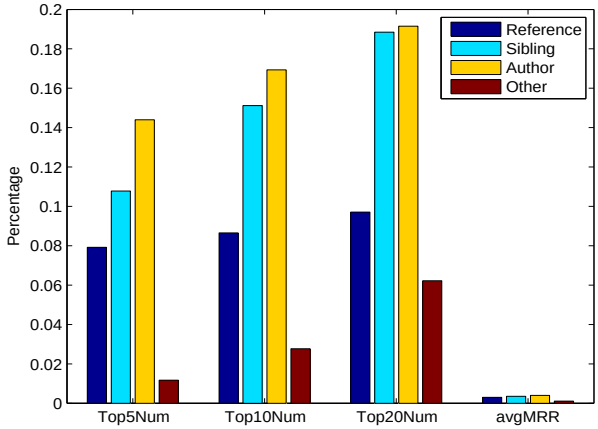

Figure 5.8: CiteSeer data set: Individual contribution of types of neighbors

neighbors respectively; the formula can be re-written as: $P(v_j|p_i) = \alpha_1 A_j + \alpha_2 R_j + \alpha_3 S_j + \alpha_4 O_j$

Let us suppose the real publishing venue for the target paper p_i is venue v_j, then in an ideal venue recommendation system, for any other venue candidate v_k rather than v_j, the computed probability score $P(v_k|p_i)$ should be less or at

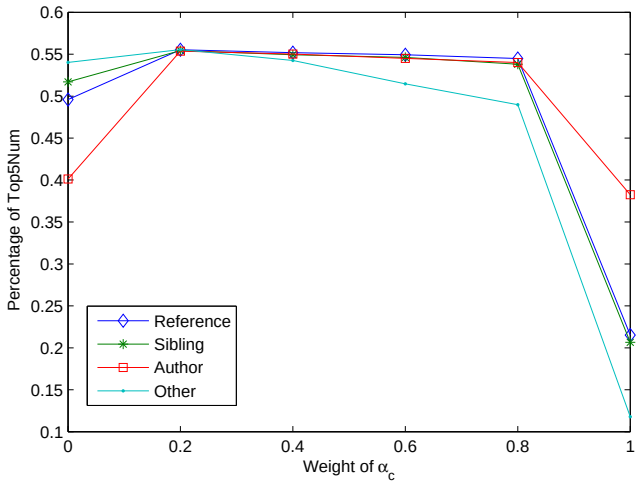

Figure 5.9: ACM data set: Weight of Neighbors

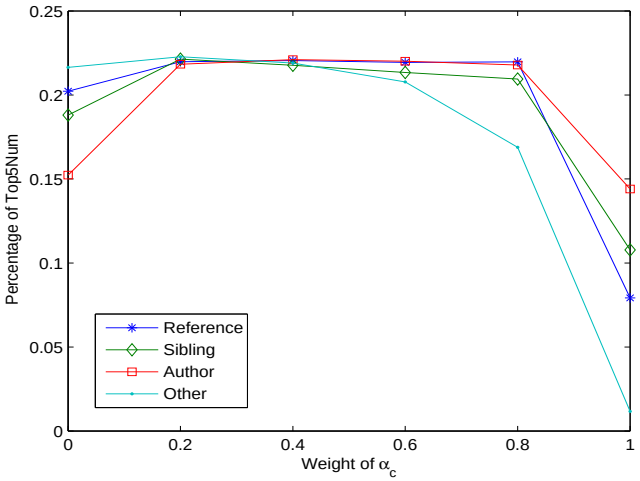

Figure 5.10: CiteSeer data set: Weight of Neighbors

most equal to $P(v_j|p_i)$; that is, we need to have $P(v_j|p_i) - P(v_k|p_i) \geq 0$ for all v_k $(k \neq j)$. Naturally, our goal is to learn the values of the four parameters α_c $(1 \leq c \leq 4)$, such that $\sum_{k:1 \to V}(P(v_j|p_i) - P(v_k|p_i))$ can be maximized, where V is the number of candidate venues. Therefore, we introduce our objective function

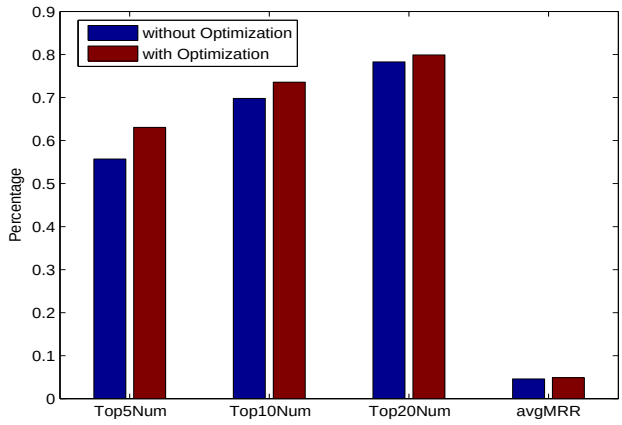

Figure 5.11: ACM data set: Parameter Optimization

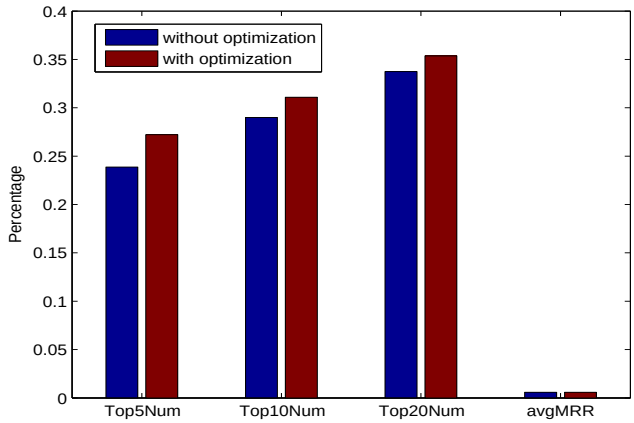

Figure 5.12: CiteSeer data set: Parameter Optimization

as: $h = argmax \sum_{k:1 \to V} s(P(v_j|p_i) - P(v_k|p_i))$ where $s(x)$ is the sigmoid function: $s(x) = \frac{1}{1+e^{-x}}$.

To achieve the optimal combination of weights, we use gradient descent in which the four parameters are updated in each iteration until they converge.

As shown in Figure 5.11 and 5.12, we achieved more than a 13% improvement in Accuracy@5 for both ACM and CiteSeer.

Results analysis: Comparisons with other approaches

In order to demonstrate the effectiveness of our proposed approach, we compare results across several baseline algorithms:

Simple Counting: For each target paper p_i, we simply count the occurring frequency of venues of three kinds of neighboring papers of paper p_i, i.e., the reference neighboring papers (papers cited by p_i, referred as SimpleCount-Ref), sibling neighboring papers (papers that share at least on citation with p_i, referred as SimpleCount-Sibling) and author neighboring papers (other papers written by authors of p_i, referred as SimpleCount-Author). We also count the frequency of venues of the combination of all three kinds of neighboring papers (referred as SimpleCount-All). We would then rank and return the venues in terms of their frequency.

Content-based LDA: We construct a profile for each venue by concatenating all the papers published in it. We use LDA topic model implemented by Mallet [115] to retrieve the topic distribution for each paper and venue over 100 topics. We then compute and rank venues by their similarities with the target paper.

Traditional memory-based CF: We use the original traditional memory-based CF approach, in which we do not incorporate stylometric features of papers to compute their similarity, nor do we categorize neighboring papers and differentiate their different contributions. Under this scheme, $P(v_j|p_i)$ can be computed as: $P(v_j|p_i) = \sum_{p_k \subset S(p_i)} s(p_i, p_k) I(p_k, v_j)$, where papers' similarity is determined by their topic distribution obtained from LDA.

Graph-based FolkRank algorithm: We used the FolkRank algorithm [76], which is an adaptation of PageRank, and has been shown empirically to generate high quality recommendations in tag recommendation systems. The basic idea of this approach is to run PageRank algorithm twice, giving uniform initial weights to all nodes in the first time, and giving higher weight to targeted nodes in the second time. The difference in terms of the weight of the nodes is then used to generate the final ranking.

We compare the results using our proposed approach with the baseline algorithms, and show the results in Table 5.14. The results we report under our method

are the best results we can achieve when both stylometric and content features are combined and all neighboring papers are considered. As indicated from the results, our approach outperforms the baseline algorithms under all evaluation metrics. The content-based approach works the worst. TraditionalCF can work better than the graph-based FolkRank algorithm; moreover, we can achieve better performance when no normalization is introduced. The SimpleCount-based method can provide surprisingly good results, and is the second best algorithm among all compared algorithms. However, our model can improve performance over SimpleCount-All by 18.53% (on ACM) and 19.77% (on CiteSeer) in terms of Accuracy@5.

Case Study Example

We show in this section several recommendation examples using our proposed approach. We report in Table 5.15 the Top 5 returned venues for three randomly chosen papers in our system. Venue names written in bold indicate the actual publishing venue of that paper. We observed that for each target paper, under most circumstances, the top five returned venues share similarity in topics, and are content-related to the target paper. They are all reasonable candidate venues to which the paper could have been submitted. For papers that concentrate on topics within specific subset of a wide research domain, or discussed topics covering interdisciplinary domains, we can also provide proper recommendation. For example, paper 1 focuses on modeling language, and therefore some computational linguistics related venues are ranked highly, such as ACL. Paper 2 discussed database integrated view design, and therefore venues in the database domain like SIGMOD and VLDB are returned. We also notice that some paper may have other appropriate choices when considering submitting; for example, for paper 3, even though its actual publishing venue is only ranked 8th, several other venues ranked higher than the actual venue are also good places to submit.

5.4 Bibliographic Notes

There is a lack of prior work exploring the problem of classifying venues by their writing styles. However, there has been a long history in the research of author attribution, also known as author identification or verification, whose main task is to determine the author of a piece of work, mostly by identifying the unique writing styles of authors. Author attribution has been used in a small yet diverse number of applications, such as authorship verification for literature and published articles, for online messages [205, 6], plagiarism detection and forensic analysis for criminal cases.

One of the most important components for author attribution is to identify representative stylometric features, which compared to the features used in text content classification, are assumed to be topic-independent and context-free. Stylometric features used in early author attribute studies are lexical [127] (i.e., character and word) based, such as number of words and characters, word length, vocabulary richness [201, 174]. Further study then began to make use of syntactic features [165]. The three most representative syntactic features are function words [24, 75], punctuation [29] and part-of-speech tags [165]. More recently, structural features [37], such as number of paragraphs, use of indentation, use of signature, have attracted attention, especially for online message authorship identification. Other useful stylometric features include character-based n-grams [92] and POS-based n-grams [52]. However, due to different applications, no set of significant stylometric features have been identified to be the most discriminative.

Just as there are a range of stylometric features, there are also many techniques for author attribution. In most cases, this task has been treated as a single-label multi-class classification task, and therefore many classification techniques have been considered [205]. Besides that, there are other techniques such as statistical approaches [51], neural networks [107], genetic algorithms [75], and principle component analysis approaches [24]. Most recently, researchers have started to use latent factor models into author attribution task [154, 7]. However, there is no consensus on which particular approach can perform the best due to different applications.

In this chapter we conduct a detailed study of venue classification by adopting a set of stylometric features that have been demonstrated useful in author attribution. Unlike most author attribution experiments, we test large numbers of classes (venues). We work on real data sets, collecting paper instances according to the actual distributions of venues in the data corpus. Moreover, we compare classification results using different feature sets and classifiers and further examine the distinguishing power between creating style-based classifiers and content-based classifiers. We further explore the relationship between writing styles and topics and genres respectively.

For the task of venue recommendation, two previous works have been proposed that consider this problem. Lau and Cohen [99] develop a combined path-constraint random walk-based approach, not only for venue recommendation, but also for citation recommendation, gene recommendation and expert finding. In their work, they would present each term in the paper title as a node, combined with other entities, like author names and venue names to construct a big graph. Complex optimization approaches are carried out to learn the weights on each edge of the graph. Pham et al. [135, 136] define the task of venue recommendation as predicting the participating venues of users, and therefore their input is users instead of papers, which is different from our work. They use a clustering-based approach to group users that share similar patterns in choosing publishing venues.

5.5 Summary

We first addressed in this chapter the task of venue classification, for which we tested whether venues are distinguishable by the writing styles of papers published in them. We applied the traditional classification approach for this task, and identified over 300 stylometric features for representing papers' writing styles. Experiments on both ACM and CiteSeer data sets demonstrated that venues can be distinguished by their writing styles. By combining both stylometric features with traditional content-based features using papers' full content, we can get improved performance

for venue classification. We examined the impact of three different classifiers: RandomForest, NaiveBayes and SVM. Even though they perform differently on different experimental settings, RandomForest, however, turns out to work the best in general. We further examined the contribution of different feature sets in which lexical features were found to be the most valuable. Moreover, we carried out experiments to test the relationship between venues topics and writing styles as well as venue genres and writing styles, both of which achieved positive results on the tested venues.

We then applied the memory-based collaborative filtering approach for venue recommendation, and in particular, we updated the original CF based approach by applying two extensions. The first extension is to incorporate papers' stylometric features to better measure the similarity between papers, and the second one is to divide the neighboring papers into four categories. By tuning or optimizing the different contributions of four categories of neighboring papers, we succeeded in obtaining better recommendation performance. Experiments demonstrate our approach to be an effective method for venue recommendation, which outperformed several baseline algorithms. By differentiating the four categories of neighboring papers' contributions, we also find that papers that are published by the same authors are the most reliable source of information for the venue recommendation task.

Table 5.1: Features

Type	Features	Description
Lexical	TokenNum	Total number of words
	TypeNum	Total number of distict words
	CharNum	Total number of characters
	SentenceNum	Total number of sentences
	AvgSenLen	Average sentence length
	AvgWordLen	Average word length
	ShortWordNum	Total number of short words (less than 3 characters) normalized by TokenNum
	HapaxVSToken	Frequency of once-occuring words normalized by TokenNum
	HapaxVSType	Frequency of once-occuring words normalized by TypeNum
	ValidCharNum	Total number of characters excluding the non-digital, non-alphabetical and non-white-space characters
	AlphaCharNum	Total number of alphabetic characters normalized by CharNum
	DigitalCharNum	Total number of digital characters normalized by CharNum
	UpperCaseNum	Total number of characters in upper-case normalized by CharNum
	WhiteSpaceNum	Total number of white-space characters normalized by CharNum
	SpaceNum	Total number of space characters normalized by CharNum
	TabSpaceNum	Total number tab spaces normalized by CharNum
	Vocabulary Richness	A vocabulary richness measure defined by Zipf
Syntactic	FuncWordNum	Total number of function words
	PunctuationNum	Total number of punctuation characters (‘.’, ‘?’, ‘!’, ‘,’, ‘:’, ‘;’, ‘”, ‘ / ’)
	FuncWordFreq	Frequency of function words normalied by FuncWordNum (298 features)
Structural	SectionNum	Total number of sections
	FigureNum	Total number of figures
	EquationNum	Total number of equations
	TableNum	Toatl number of tables
	ReferenceNum	Total number of references

Table 5.2: Statistics over Chosen Venues

	Avg. No. of Papers Per Venue	Avg. length of Papers per Venue (Abstract)	Avg. length of Papers per Venue (Full Paper)
ACM	415	105 words	N/A
CiteSeer	98	140 words	6490 words

Table 5.3: Multi-Class Venue Classification for ACM Data Set. Value* is significantly better than the Baseline Classifier. The 'Baseline' algorithm here means 'random guessing'

		Accuracy	F_1 Score
2-Venue	Baseline	0.503	0.481
	Stylometric	**0.806***	**0.713***
5-Venue	Baseline	0.195	0.177
	Stylometric	**0.584***	**0.454***
10-Venue	Baseline	0.099	0.085
	Stylometric	**0.434***	**0.309***
30-Venue	Baseline	0.033	0.027
	Stylometric	**0.267***	**0.118***
50-Venue	Baseline	0.020	0.015
	Stylometric	**0.207***	**0.077***
100-Venue	Baseline	0.010	0.008
	Stylometric	**0.113***	**0.050***
150-Venue	Baseline	0.007	0.005
	Stylometric	**0.099***	**0.040***

Table 5.4: Multi-Class Venue Classification for CiteSeer Data Set. Value * is significantly better than the Baseline Classifier. Value † is significantly better than the Stylometric(A) Classifier. The 'Baseline' algorithm here means 'random guessing'

		Accuracy	F_1 Score
2-Venue	Baseline	0.498	0.485
	Stylometric(A)	0.707*	0.658*
	Stylometric(F)	**0.847***	**0.828***
5-Venue	Baseline	0.206	0.197
	Stylometric(A)	0.413*	0.342*
	Stylometric(F)	**0.625***	**0.570***
10-Venue	Baseline	0.101	0.095
	Stylometric(A)	0.254*	0.196*
	Stylometric(F)	**0.450***	**0.391***
30-Venue	Baseline	0.033	0.031
	Stylometric(A)	0.106*	0.079*
	Stylometric(F)	**0.246***†	**0.188***†
50-Venue	Baseline	0.019	0.017
	Stylometric(A)	0.066*	0.051*
	Stylometric(F)	**0.156***†	**0.116***†
100-Venue	Baseline	0.010	0.009
	Stylometric(A)	0.034*	0.028*
	Stylometric(F)	**0.094***†	**0.044***†
150-Venue	Baseline	0.007	0.007
	Stylometric(A)	0.022*	0.018*
	Stylometric(F)	**0.062***†	**0.044***†

Table 5.5: Accuracy for Different Feature Sets and Techniques

	ACM			CiteSeer		
	RF	NB	SVM	RF	NB	SVM
Lexical	**0.425**	0.170	0.403	**0.435**	0.315	0.355
Syntactic	0.382	0.165	**0.402**	**0.416**	0.366	0.267
Structural	**0.304**	0.131	0.291	**0.294**	0.265	0.221
Lexi+Syn	0.429	0.177	**0.433**	**0.447**	0.383	0.388
Lexi+Str	**0.423**	0.173	0.414	**0.441**	0.329	0.357
Syn+Str	0.386	0.165	**0.410**	**0.436**	0.372	0.269
Lexi+Syn+Str	0.434	0.186	**0.455**	**0.450**	0.389	0.390

Table 5.6: F_1 Score for Different Feature Sets and Techniques

	ACM			CiteSeer		
	RF	NB	SVM	RF	NB	SVM
Lexical	**0.273**	0.132	0.146	**0.382**	0.257	0.203
Syntactic	**0.224**	0.158	0.151	**0.354**	0.339	0.076
Structural	**0.109**	0.105	0.100	**0.247**	0.199	0.038
Lexi+Syn	**0.298**	0.182	0.224	**0.389**	0.349	0.240
Lexi+Str	**0.285**	0.173	0.147	**0.376**	0.274	0.207
Syn+Str	**0.247**	0.165	0.149	**0.373**	0.347	0.089
Lexi+Syn+Str	**0.309**	0.191	0.239	**0.391**	0.359	0.245

Table 5.7: P-values of pairwise t tests on Accuracy for different types. Symbol * indicates statistical significance

Feature Sets	ACM	CiteSeer
Lexical vs. Syntactic	0.2179	0.1264
Lexical vs. Structural	0.0018*	0.0005*
Syntactic vs. Structural	0.0035*	0.0012*
Lex vs. Lex+Syn	0.0482*	0.0407*
Lex+Syn vs. Lex+Syn+Stru	0.2210	0.1987

Table 5.8: CiteSeer Data Set: Contribution of individual features

Feature	Accuracy	Feature	F_1 Score
tableNum	0.5972	**tableNum**	0.5416
RefNo	0.6041	**RefNo**	0.5483
TokenNum	0.6072	**TokenNum**	0.5492
TabSpaceNo	0.6084	AlphaCharNo	0.5511
AlphaCharNo	0.6090	AvgWordLen	0.5559
FuncWordDis	0.6096	TabSpaceNo	0.5563
figureNum	0.6100	FuncWordNum	0.5581
TypeNum	0.6125	SentenceNum	0.5586
CharNum	0.6127	FuncWordDis	0.55927
upperCaseNo	0.6129	DigitalCharNo	0.55930
punctuNo	0.6137	figureNum	0.5594
equationNum	0.61376	equationNum	0.55956
AvgWordLen	0.61377	SpaceNo	0.55962
SpaceNo	0.6143	AvgSentenceLen	0.5598
DigitalCharNo	0.6148	TypeNum	0.5607
FuncWordNum	0.6157	upperCaseNo	0.5609
HapaxVSType	0.61585	CharNum	0.5615
AvgSentenceLen	0.61588	sectionNum	0.56355
SentenceNum	0.6162	ValidCharNo	0.56361
ShortWordNum	0.6177	HapaxVSType	0.5647
sectionNum	0.6179	punctuNo	0.5648
ValidCharNo	0.6185	ShortWordNum	0.5662
HapaxVTToken	0.6195	HapaxVSToken	0.5669
whiteSpaceNo	0.6211	whiteSpaceNo	0.5683
VocRichness	0.6228	VocRichness	0.5695
All	0.6245	All	0.5701

Table 5.9: Content vs. Writing Style: ACM data set. Value* is significantly better than Stylometric Classifier

		Accuracy	F_1 Score
2-Venue	Stylometric	0.806	0.713
	Content	**0.916**	**0.888**
	Combine	0.884	0.836
5-Venue	Stylometric	0.584	0.454
	Content	**0.798**	**0.706**
	Combine	0.742	0. 636
10-Venue	Stylometric	0.434	0.309
	Content	**0.657**	**0.528**
	Combine	0.595	0.444
30-Venue	Stylometric	0.267	0.118
	Content	**0.491***	**0.302***
	Combine	0.419*	0.227*
50-Venue	Stylometric	0.207	0.077
	Content	**0.407***	**0.216***
	Combine	0.342*	0.155*
100-Venue	Stylomeric	0.113	0.050
	Content	**0.280***	**0.141***
	Combine	0.217*	0.101*
150-Venue	Stylometric	0.099	0.040
	Content	0.135*	**0.085***
	Combine	**0.179***	0.074*

Table 5.10: Writing Styles vs. Genres

Conference vs. Journal	Accuracy	F_1 Score
Overall	0.7680	0.7679
Database	0.7965	0.7949
Computer Graphics	0.5887	0.5885
Computer Architecture	0.7670	0.7668

Table 5.11: Content vs. Writing Style: CiteSeer Data Set. Value* is significantly better than Stylometric classifier. Value† indicates that 'Combine' Classifier is significantly better than 'Content' Classifier

		Accuracy	F_1 Score
2-Venue	Stylometric(F)	0.847	0.828
	Content	0.885	**0.868**
	Combine	**0.886**	0.866
5-Venue	Stylometric(F)	0.625	0.570
	Content	0.687	0.638
	Combine	**0.691**	**0.645**
10-Venue	Stylometric(F)	0.450	0.391
	Content	0.504	0.442
	Combine	**0.516**†	**0.458**†
30-Venue	Stylometric(F)	0.246	0.188
	Content	0.270	0.211
	Combine	**0.286**	**0.225**
50-Venue	Stylometric(F)	0.156	0.116
	Content	0.187*	0.141*
	Combine	**0.191***	**0.145***
100-Venue	Stylometric(F)	0.094	0.044
	Content	0.111*	0.086*
	Combine	**0.116***†	**0.087***†
150-Venue	Stylometric(F)	0.062	0.044
	Content	0.075*	0.059*
	Combine	**0.079***	**0.060***

Table 5.12: Writing Styles vs. Topics

		Accuracy	F_1 Score
SIGIR	WWW	0.730	0.729
SIGIR	CIKM	0.660	0.659
SIGIR	SIGKDD	0.755	0.755
SIGIR	JCDL	0.690	0.688
SIGIR	computer architecture	0.855	0.855
SIGIR	parallel computing	0.895	0.895
SIGIR	graphics	0.845	0.844

Table 5.13: Venue Recommendation Results on ACM and CiteSeer data

*T*op K=500						
	ACM			CiteSeer		
	Style	Content	S+C	Style	Content	S+C
Accuracy@5	0.084	0.103	**0.150**	0.065	**0.125**	0.108
Accuracy@10	0.150	0.190	**0.291**	0.086	**0.172**	0.148
Accuracy@20	0.265	0.352	**0.526**	0.141	**0.251**	0.231
MRR	0.002	0.003	**0.005**	0.010	0.013	0.013
*T*op K=1000						
	Style	Content	S+C	Style	Content	S+C
Accuracy@5	0.081	0.081	**0.150**	0.086	**0.122**	0.116
Accuracy@10	0.138	0.151	**0.286**	0.105	**0.157**	0.152
Accuracy@20	0.239	0.272	**0.504**	0.137	**0.212**	0.209
MRR	0.003	0.004	**0.009**	0.008	0.009	0.009
*T*op K=2000						
	Style	Content	S+C	Style	Content	S+C
Accuracy@5	0.079	0.071	**0.166**	0.114	0.122	**0.130**
Accuracy@10	0.128	0.124	**0.319**	0.131	0.156	**0.162**
Accuracy@20	0.224	0.221	**0.520**	0.158	0.197	**0.209**
MRR	0.005	0.006	0.013	0.006	0.007	**0.008**
*T*op K=5000						
	Style	Content	S+C	Style	Content	S+C
Accuracy@5	0.080	0.075	**0.214**	0.153	0.117	**0.158**
Accuracy@10	0.128	0.124	**0.375**	0.177	0.148	**0.196**
Accuracy@20	0.220	0.217	**0.559**	0.203	0.197	**0.236**
MRR	0.009	0.008	**0.022**	0.006	0.006	**0.007**
*T*op K=10000						
	Style	Content	S+C	Style	Content	S+C
Accuracy@5	0.086	0.082	**0.249**	0.190	0.118	**0.195**
Accuracy@10	0.134	0.140	**0.422**	0.221	0.161	**0.231**
Accuracy@20	0.230	0.241	**0.604**	0.250	0.227	**0.272**
MRR	0.011	0.009	**0.027**	0.006	0.006	**0.007**
*A*ll Neighbors						
	Style	Content	S+C	Style	Content	S+C
Accuracy@5	0.502	0.367	**0.557**	0.238	0.124	**0.239**
Accuracy@10	0.623	0.492	**0.698**	0.286	0.178	**0.290**
Accuracy@20	0.716	0.600	**0.783**	0.332	0.250	**0.337**
MRR	0.032	0.016	**0.046**	0.006	**0.007**	0.006

Table 5.14: ACM and CiteSeer: Comparison with baseline algorithms

ACM Data	Accuracy@5	Accuracy@10	Accuracy@20	MRR
SimpleCount-Ref	0.203	0.212	0.212	0.0006
SimpleCount-Sibling	0.252	0.307	0.344	0.0008
SimpleCount-Author	0.377	0.430	0.446	0.0008
SimpleCount-All	0.470	0.566	0.603	0.0013
contentLDA	0.010	0.018	0.024	0.0008
traditionalCF	0.317	0.467	0.608	0.0283
FolkRank	0.102	0.184	0.252	0.0087
Our method	**0.557**	**0.698**	**0.783**	**0.0459**
CiteSeer	Accuracy@5	Accuracy@10	Accuracy@20	MRR
SimpleCount-Ref	0.096	0.099	0.099	0.0001
SimpleCount-Sibling	0.112	0.141	0.161	0.0001
SimpleCount-Author	0.129	0.157	0.176	0.0001
SimpleCount-All	0.199	0.239	0.277	0.0002
contentLDA	0.008	0.016	0.022	0.0005
traditionalCF	0.095	0.015	0.224	0.0040
FolkRank	0.037	0.068	0.113	0.0037
Our method	**0.239**	**0.290**	**0.337**	**0.0058**

Table 5.15: Venue Recommendation Results: Examples

Papers and their Top 5 Predicted Venues
1. corpus structure language models and ad hoc information retrieval (SIGIR 2004) *predicted:* 1) annual meeting acl 2) **annual intl acm sigir conf on research and development in information retrieval** 3) journal machine learning research 4) computational linguistics 5) acm ieee cs joint conf on digital libraries
2. induction of integrated view for xml data with heterogeneous dtds (CIKM 2001) *predicted:* 1) acm sigmod intl conf on management data 2) **intl conf on information and knowledge management** 3) acm symposium on applied computing 4) communications acm 5) vldb journal mdash intl journal on very large data bases
3. multi resolution indexing for shape images (CIKM 1998) *predicted:* 1) acm intl conf on multimedia 2) intl conf on very large data bases 3) annual acm siam symposium on discrete algorithms 4) conf on visualization 5) annual conf on computer graphics and interactive techniques (rank 8) **intl conf on information and knowledge management**
4. video suggestion and discovery for youtube taking random walks through the view graph (WWW 2008) *predicted:* 1) intl conf on human computer interaction with mobile devices and services 2) annual sigchi conf on human factors in computing systems 3) acm sigkdd intl conf on knowledge discovery and data mining 4) **intl conf on world wide web** 5) annual meeting on association for computational linguistics

Chapter 6

Academic Network Analysis: a Joint Topical Modeling Approach

Generative topic modeling provides an extensible platform to integrate multiple types of entities and discovery their underlying semantics (topics) over words. In this chapter, we continue on developing enhanced topic modeling approach for expert ranking. Compared to the work conducted in chapter 4, we integrate two more important factors: the publishing venues and cited authors into topic modeling process. Experiments show that additional information can improve ranking performance. We also demonstrate the capability of the model in predicting publishing venues and cited authors via experimental studies.

6.1 Introduction

Social network research has attracted the interests of many researchers, not only in analyzing online social media applications, such as Facebook and Twitter, but also in providing comprehensive services in the domain of scientific research. We define an *academic network* as a kind of social network which integrates scientific factors, such as authors, papers, publishing venues, and their relationships. With

the rapid development of online digital libraries, the proliferation of large quantities of scientific literature provides us abundant opportunity to extract the textual content of scientific factors (i.e., publishing papers) as well as their mutual relationships (citation, coauthorship), and therefore stimulates the emergence of many applications that are particularly important in academic domain (in mining and analyzing academic networks), such as expert ranking, citation prediction, cited author prediction, venue prediction, etc.

Generative topic modeling has emerged as a popular unsupervised learning technique for content representation in large document collections. This kind of generative model was first envisioned for pure contextual analysis while ignoring the linkage structure among text data. Representative models of this type of analysis (e.g., [72, 18]) exploit the co-occurrence patterns of words in documents and unearth the semantically meaningful clusters of words (as topics). Researchers have since added extensions to model authors' interests [147], providing a framework for answering questions and making predictions at the level of authors rather than documents, and in a variety of other aspects, such as incorporating link structures and integrating additional context information.

Despite such recent developments (which we review in Section 2), limitations are still present. It is widely acknowledged that one of most prominent advantages of generative topic modeling is that it provides us a flexible and extensible framework to exploit the underlying latent structures over text data as well as their mutual connections. In the academic network, we have multiple kinds of scientific factors and connections; however, most of the previous work considers one aspect of several factors while ignoring some others.

In this chapter, we provide a framework that can jointly model authors, papers, cited authors, and venues in one unified model. We name the model as the Author-Citation-Venue topic model (abbreviated as ACVT model), in which we link authors to observed words, cited authors and venues via latent topics. We hypothesize that such a joint modeling has multiple advantages. First of all, this model provides a more comprehensive framework to fully utilize the content words of documents and combines them with other useful contextual information: authors, cited authors

and venues. It therefore directly models documents' content relevance, authors' interests, authors' influence, and venues' influence in one model, all of which are important instructive evidence in supporting academic network based applications, such as expert ranking, cited author prediction, and venue prediction. Missing the integration of one sort of contextual information, some certain kind of application would become impossible; for example, if the topic-venue association is not explored, we cannot make valid venue predictions. Our model therefore can be applied in a wider range of applications than previous work. Moreover, incorporating additional contextual and linkage information can help to identify more coherent and complete latent structures over multiple facets. In the ACVT model, we assume that we can achieve better topic-related associations for authors, cited authors and venues when we simultaneously model them together, and such associations with greater coherency are believed to be able to further improve the performance of multiple applications.

In summary, we make the following contributions in this chapter:

- We propose a generative model that incorporates multiple facets of academic network: authors, papers, venues and cited authors in an integrated fashion.
- We apply our model, and provided solutions to three tasks in the academic domain: expert ranking, cited author prediction and venue prediction.
- Experiments based on two real world data sets demonstrate our model to be effective on all three tasks, significantly outperforming several state-of-the-art algorithms.

The rest of this chapter is organized as follows. We introduce the ACVT model as well as the parameter estimation methodin section 6.2. Three applications of this model are introduced in section 6.3, with their experimental results discussed in section 6.4. We review related work in section 6.5 and conclude this chapter in section 6.6.

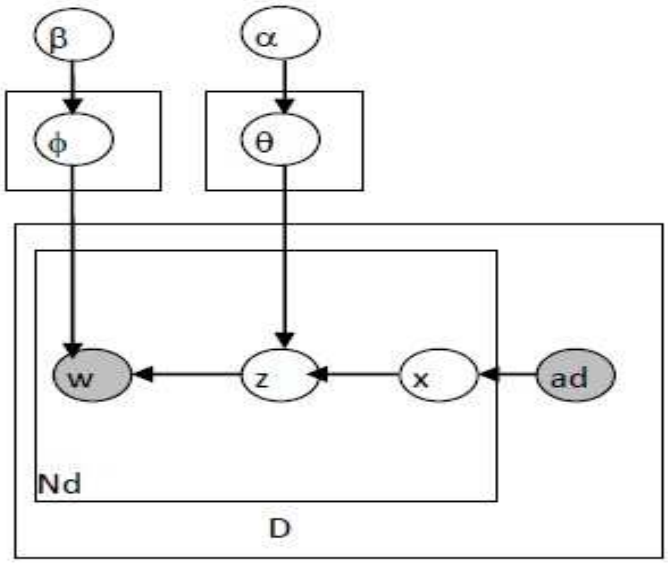

Figure 6.1: Graphical Model for the original Author-Topic Model

6.2 Model

Before presenting the model, we first introduce some notation. Suppose W, D, A, V indicate the size of the word vocabulary, the number of papers, the number of authors (cited authors), and the size of venues in the corpus respectively. $\boldsymbol{a_d}$, $\boldsymbol{c_d}$ and N_d denote the set of authors, the set of cited authors, and the number of position-based words in paper d. T denotes the number of latent topics predefined. We further suppose that there exists a $A \times T$ author-topic distribution matrix θ indicating the distribution of authors over topics, a $T \times W$ topic-word distribution matrix ϕ indicating the probability distribution of topics over words, an $T \times A$ distribution matrix φ indicating the distribution of topics over cited authors, and a $T \times V$ distribution matrix ϑ indicating the distribution of topics over venues. $\boldsymbol{z}$, $\boldsymbol{x}$, $\boldsymbol{m}$, $\boldsymbol{s}$ are random variables, representing the topic assignment, author assignment, cited author assignment and venue assignment for each word. α, β, γ, and λ are the Dirichlet prior hyper-parameters that determine θ, ϕ, φ, and ϑ respectively. We list the detailed notation in Table 6.1.

Table 6.1: Notation

Symbol	Size	Description
W	scalar	size of word vocabulary
D	scalar	number of papers
A	scalar	number authors (cited authors)
V	scalar	number of venues
T	scalar	number of latent topics
N_d	scalar	the number of words in paper d
A_d	scalar	the number of authors of paper d
C_d	scalar	the number of cited authors of paper d
N	scalar	the number of words in corpus
		Observed Data
$\boldsymbol{a_d}$	$\lvert\boldsymbol{a_d}\rvert$	the set of authors of paper d
$\boldsymbol{c_d}$	$\lvert\boldsymbol{c_d}\rvert$	the set of cited authors of paper d
$\boldsymbol{w_d}$	$\lvert\boldsymbol{w_d}\rvert$	the words lists of paper d
v_d	1	the publishing venue of paper d
$\mathcal{A}$	A	the set of authors (cited author) in corpus
$\boldsymbol{w}$	N	the set of word tokens in corpus
$\mathcal{V}$	V	the set of venues in corpus
		Hyper-Parameters
α	$1 \times T$	Dirichlet prior for θ
β	$1 \times T$	Dirichlet prior for ϕ
γ	$1 \times T$	Dirichlet prior for φ
λ	$1 \times T$	Dirichlet prior for ϑ
		Random Variables
θ	$A \times T$	distribution of authors over topics
ϕ	$T \times V$	distribution of topics over words
φ	$T \times A$	distribution of topics over cited authors
ϑ	$T \times C$	distribution of topics over venues
$\boldsymbol{z_{di}}$	$1 \times T$	topic assignments for i^{th} word in paper d
$\boldsymbol{x_{di}}$	$1 \times \lvert a_d\rvert$	author assignments for i^{th} word in paper d
$\boldsymbol{m_{di}}$	$1 \times \lvert c_d\rvert$	cited author assignments for i^{th} word in paper d
$\boldsymbol{s_{di}}$	scalar	venue assignments for ith word in paper d

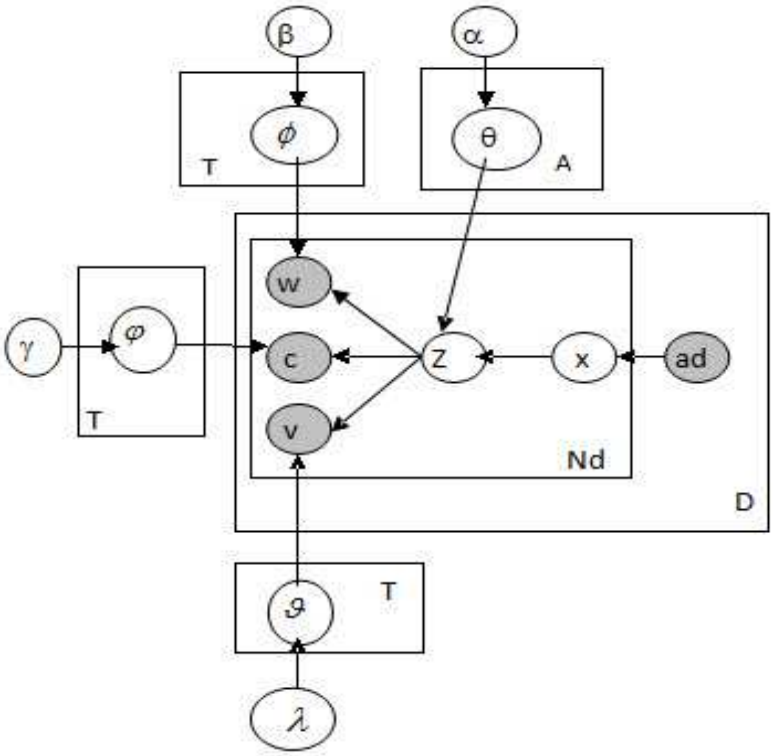

Figure 6.2: Graphical Model for the Author-Citation-Venue-Topic Model

6.2.1 Model Description / Generative Process

We depict the graphical model of ACVT in Figure 6.2 as compared to the original Author-Topic Model shown in Figure 6.1. As indicated, the graphical model is composed of six plates. Besides the four plates representing Topics, Authors, Documents and words in each document, ACVT introduces two additional plates, representing the topic-cited author association and topic-venue association respectively. As we can see, authors, words, cited authors and venues are all connected via the latent topics. Note that even though the author list and cited author list for any given paper d are assumed to be known, the exact author and cited author assignment for each particular word in paper d are unknown.

Within ACVT, each author is associated with a multinomial distribution over topics θ, and each topic is associated with a multinomial distribution over words

ϕ, a multinomial distribution over cited authors φ, and a multinomial distribution over venues ϑ. Moreover, θ, ϕ, φ and ϑ follow a Dirichlet distribution with respect to the Dirichlet prior α, β, γ, and λ respectively.

The design of the ACVT model captures the intuition of people writing a paper. Normally, when authors start to write a paper, they should have known what they are going to write about, namely, the topics of their paper. Based upon the chosen topics, they will then choose the exact words to use to represent their intended topics, figure out other related works and their corresponding authors to cite, and determine where to submit this chapter. We assume that one paper may address multiple topics, and can be co-authored by more than one author, and that each of the co-authors may have different weights of contributions to a specific topic.

The generative process of the ACVT model can be described as follows. We first sample the author-topic, topic-word, topic-cited author and topic-venue distributions based on the four Dirichlet prior hyper-parameters. Suppose we know the author lists of papers; then for each word in a given paper, we would first draw an author from its author list, then conditioned on this drawn author and his associated author-topic distribution, we sample one topic, based upon which, we further sample the cited author, venue and word according to their topic-related distributions independently.

Under this generative process, the likelihood of the corpus $\boldsymbol{w}$, conditioned on θ, ϕ, φ, and ϑ is:

$$
\begin{aligned}
& p(\boldsymbol{w}|\theta,\phi,\varphi,\vartheta,\mathcal{A},\mathcal{V}) \\
&= \prod_{d=1}^{D} p(\boldsymbol{w_d}|\theta,\phi,\varphi,\vartheta,\boldsymbol{a_d},\boldsymbol{c_d},v_d) \\
&= \prod_{d=1}^{D}\prod_{i=1}^{N_d}\frac{1}{A_d}\sum_{a\in\boldsymbol{a_d}}\sum_{t=1}^{T}\sum_{c=1}^{C_d}\varphi_{tc}\vartheta_{tv_d}\phi_{tw_{di}}\theta_{at}
\end{aligned}
$$

6.2.2 Parameter Inference and Estimation

The primary inference goal of our ACVT model is to estimate the posterior distribution of two sets of unknown random variables: (1) the distribution of θ, ϕ, φ and ϑ, and (2) the topic, author, cited author and venue assignments for each word w_{di}: z_{di}, x_{di}, m_{di}, s_{di}.

$$p(\theta, \phi, \varphi, \vartheta, \boldsymbol{z}, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{s} | D^{train}, \alpha, \beta, \gamma, \lambda) \tag{6.1}$$

where, $\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{s}$ indicate the topic, author, cited author and venue assignments for all word tokens in corpus.

Even though calculating these posterior distributions is intractable for exact inference, various approximate inference models have been employed to estimate these posterior distributions in hierarchical Bayesian models, including variational inference [18], expectation propagation[124], and Markov chain Monte Carlo (MCMC) schemes. In this chapter, we use Gibbs Sampling [63], a special case of the MCMC approximation scheme, which is not necessarily as computationally efficient as variational inference and expectation propagation, but is unbiased and simple to implement.

The entire inference process involves two steps. Firstly, we obtain an empirical sample-based estimate of $p(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{s} | D^{train}, \alpha, \beta, \gamma, \lambda)$ using Gibbs Sampling, and then secondly, we infer the posterior distribution of θ, ϕ, φ, and ϑ based upon $\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{s}$, by exploiting the fact that the Dirichlet distribution is conjugate to the multinomial distribution.

1). Gibbs Sampling for $\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{s}$

Using Gibbs Sampling, we construct a Markov chain, in which the transition between two successive states results from repeatedly drawing the four-tuple $< z, x, m, s >$, i.e., the assignment of topic, author, cited author, and venue for each word as a block from its distribution, conditioned on all other variables. Such a sampling process would be repeated until it finally converges to the posterior distribution of $\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{s}$. The corresponding updating equation for this blocked Gibbs

Sampler can be defined as:

$$
\begin{aligned}
& p(x_{di} = a, z_{di} = t, m_{di} = c, s_{di} = v | \boldsymbol{U_{known}}) \\
\propto & \frac{C^{AT}_{at,-di} + \alpha}{\sum_{t'} C^{AT}_{at',-di} + T\alpha} \frac{C^{TW}_{tw,-di} + \beta}{\sum_{w'} C^{TW}_{tw',-di} + N\beta} \\
\times & \frac{C^{TC}_{tc,-di} + \gamma}{\sum_{c'} C^{TC}_{tc',-di} + A\gamma} \frac{C^{TV}_{tv,-di} + \lambda}{\sum_{v'} C^{TV}_{tv',-di} + V\lambda}
\end{aligned}
$$

$$
\begin{aligned}
& \boldsymbol{U_{known}} \\
= & \{w_{di} = w, \boldsymbol{z_{-di}}, \boldsymbol{x_{-di}}, \boldsymbol{m_{-di}}, \boldsymbol{s_{-di}}, \boldsymbol{w_{-di}}, \boldsymbol{a_d}, v_d, \alpha, \beta, \gamma, \lambda\}
\end{aligned}
$$

where C^{AT} represents the author-topic count matrix, and $C^{AT}_{at,-di}$ is the number of words assigned to topic t for author a excluding the topic assignment to word w_{di}. Similarly, C^{TW} represents the topic-word count matrix, and $C^{TW}_{tw,-di}$ is the number of words from the wth entry in word vocabulary assigned to topic t excluding the topic assignment to word w_{di}; C^{TC} represents the topic-cited author count matrix, and $C^{TC}_{tc,-di}$ is the number of cited authors assigned to topic t excluding the topic assignment to word w_{di}, and finally, C^{TV} represents the topic-venue count matrix, and $C^{TV}_{tv,-di}$ is the number of venues assigned to topic t excluding the topic assignment to word w_{di}. Moreover, $\boldsymbol{z_{-di}}, \boldsymbol{x_{-di}}, \boldsymbol{m_{-di}}, \boldsymbol{s_{-di}}$, and $\boldsymbol{w_{-di}}$ stand for the vector of topic, author, cited author and venue assignment and the vector of word observations in the corpus except for the i^{th} word in the d^{th} document respectively.

In implementing this Gibbs Sampling, we simply need to keep track of the four matrices (C^{AT}, C^{TW}, C^{TC}, C^{TV}). By initially assigning words to randomly chosen topic, authors, cited authors and venues, we repeatedly apply this equation to each word in corpus, until finally converged.

2). The Posterior on $\theta, \phi, \varphi, \vartheta$

After we obtain the approximated estimation of $\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{s}$, the posterior distribution of $\theta, \phi, \varphi, \vartheta$ can be directly computed by exploiting the fact that the Dirichlet

distribution is conjugate to the multinomial distribution, and therefore we have:

$$\theta|\boldsymbol{z}, \boldsymbol{x}, D^{train}, \alpha \sim Dirichlet(C^{AT} + \alpha) \tag{6.2}$$

$$\phi|\boldsymbol{z}, D^{train}, \beta \sim Dirichlet(C^{TW} + \beta) \tag{6.3}$$

$$\varphi|\boldsymbol{z}, \boldsymbol{m}, D^{train}, \gamma \sim Dirichlet(C^{TC} + \gamma) \tag{6.4}$$

$$\vartheta|\boldsymbol{z}, \boldsymbol{s}, D^{train}, \lambda \sim Dirichlet(C^{TV} + \lambda) \tag{6.5}$$

We can then estimate the posterior mean of $\theta, \phi, \varphi, \vartheta$ by following:

$$E[\theta_{at}|\boldsymbol{z}, \boldsymbol{x}, D^{train}, \alpha] = \frac{C^{AT}_{at} + \alpha}{\sum_{t'} C^{AT}_{at'} + T\alpha} \tag{6.6}$$

$$E[\phi_{tw}|\boldsymbol{z}, D^{train}, \beta] = \frac{C^{TW}_{tw} + \beta}{\sum_{w'} C^{TW}_{tw'} + W\beta} \tag{6.7}$$

$$E[\varphi_{tc}|\boldsymbol{z}, \boldsymbol{m}, D^{train}, \gamma] = \frac{C^{TC}_{tc} + \gamma}{\sum_{c'} C^{TC}_{tc'} + C\gamma} \tag{6.8}$$

$$E[\vartheta_{tv}|\boldsymbol{z}, \boldsymbol{s}, D^{train}, \lambda] = \frac{C^{TV}_{tv} + \lambda}{\sum_{v'} C^{TV}_{tv'} + V\lambda} \tag{6.9}$$

6.3 Application

We introduce in this section three main applications related to academic network analysis that can be solved by applying our ACVT model.

6.3.1 Expert Ranking

The problem of expert ranking is equivalent to the problem of finding experts. The ultimate goal of an expert finding task is to identify people who have relevant expertise to a specific topic of interest. In the academic research environment, estimating a researcher's reputation (contribution) and further ranking academic researchers is of great importance as it can offer support when making decisions about researchers' job promotion, project funding approval, paper review assignment, as well as scientific award assignment.

Rank experts by Topic Models

Based on the learning results from the ACVT model, we obtain four distributions: θ, ϕ, φ and ϑ. Suppose we are given a query q, composed of a set of words $\boldsymbol{w}$, then for any given author a in the corpus, the probability of having author a being relevant to the query q, i.e, the expertise of the author a in domain q, can be computed under our ACVT model as:

$$\begin{aligned} p_{TM}(a|q) &\propto p_{TM}(q|a) \\ &= \prod_{w \in q} p(w|a) \\ &= \prod_{w \in q} p(w|a_a) p(w|a_c) \sum_{v \in V(a)} p(w|v) \end{aligned} \tag{6.10}$$

where $p(w|a_a)$ represents the probability of author a generating word w as an author; $p(w|a_c)$ represents the probability of author a being cited by word w; $p(w|v)$ represents the probability of venue v generating word w. We consider all the publishing venues $V(a)$ of a to evaluate the relevance of author a to word w from the venue aspect of view.

Based upon the learning results from ACVT, we can further have:

$$p(w|a_a) = \sum_t p(w|z) p(z|a_a) = \sum_t \phi_{tw} \theta_{a_a t} \tag{6.11}$$

$$\begin{aligned} p(w|a_c) &= \sum_t p(w|z) p(z|a_c) \\ &\propto \sum_t p(w|z) p(a_c|z) = \sum_t \phi_{tw} \varphi_{t a_c} \end{aligned} \tag{6.12}$$

$$\begin{aligned} p(w|v) &= \sum_t p(w|z) p(z|v) \\ &\propto \sum_t p(w|z) p(v|z) = \sum_t \phi_{tw} \vartheta_{tv} \end{aligned} \tag{6.13}$$

As a result, we can compute $p_{TM}(a, q)$ by:

$$p_{TM}(a|q) \propto \prod_{w \in q} (\sum_t \phi_{tw} \theta_{a_a t}) (\sum_t \phi_{tw} \varphi_{t a_c}) (\sum_{v \in V(a)} \sum_t \phi_{tw} \vartheta_{tv}) \tag{6.14}$$

Combining with Language Model and Random-walk

We are also interested in examining whether we can achieve better performance when combining the results obtained from Topic Modeling with that of using a language model based approach and a random walk based approach, the two other representative approaches in evaluating researchers' expertise.

To evaluate the relevance of an author a to a query, we can construct a virtual document F_a of author a by concatenating all the publishing papers of author a, and thus the relevance of author a to query q would be equivalent to the relevance of document F_a to query q. Under the standard language model with Jenilek-Mercer smoothing, the probability can be computed by:

$$\begin{aligned} p_{LM}(a|q) &= p_{LM}(F_a|q) \\ &= \prod_{w \in q} \{(1-\lambda)\frac{n(w,F_a)}{n(F_a)} + \\ &\quad \lambda \frac{\sum_{F_{a'}} n(w,F_{a'})}{\sum_{F_{a'}} n(F_{a'})}\} \end{aligned} \tag{6.15}$$

A random-walk based algorithm directly models the interaction among network nodes. In this chapter, we construct a heterogeneous academic network (as shown in Figure 6.3, which follows the network design mentioned in paper [168]) which is composed of three kinds of academic factors: authors, papers and venues, and their mutual relationships: $G = (V_a \cup V_d \cup V_v, E_{ad} \cup E_{dd} \cup E_{cd})$. V_a, V_d and V_v represents the collection of authors, papers and venues respectively. Based on our definition, $(d_i, d_j) \in E_{dd}$ if paper d_i cites paper d_j. We further represent each undirected edge into two directed edges in bipartite graphs, and therefore we have both $(a_i, d_j) \in E_{ad}$ and $(d_j, a_i) \in E_{ad}$ if paper d_j is written by author a_i. Similarly, $(v_i, d_j) \in E_{vd}$ and $(d_j, v_i) \in E_{vd}$ if paper d_j is published in venue v_i.

The transition probability between any two nodes in the network is determined by two parameters: the type-based transition parameter $\lambda_{t_1 t_2}$, which determines the probability when the random surfer transfers from node of type t_1 to node of type t_2. The second parameter $p(n_1|n_2)$ determines the transition probability between

any two specific nodes, no matter what type of the nodes they are. Under this definition, if the random surfer transfers from node n_1 of type t_1 to node n_2 of type t_2, the transition probability would be $\lambda_{t_1 t_2} p(n_2|n_1)$.

Given this academic network, we apply a PageRank-like [132] propagation algorithm over it to achieve the ranking score for each 'author' node. Suppose the PageRank score of each node n_i is denoted as $r(n_i)$, and then it can be computed by:

$$r(n_j) = \frac{d}{|V|} + (1-d) * \sum_{(n_i,n_j)\in E} \lambda_{t(n_i)t(n_j)} p(n_j|n_i) \tag{6.16}$$

where $|V|$ is the total number of nodes in the network, and $t(n_i)$ indicates the type of node n_i.

We adopted two methods to combine the ranking performance of topic modeling, language model and random-walk based PageRank. To linearly combine them, the final ranking score of an author a for a given query q can be computed as:

$$p_{Final}(a|q) = \alpha p_{TM}(a,q) + \beta p_{LM}(a,q) + \gamma r(a) \tag{6.17}$$

where, α, β and γ, satisfying $\alpha + \beta + \gamma = 1$, are the parameters that need to be tuned.

We can also multiply the results obtained from the three methods, which results in the final ranking score presented as:

$$p_{Final}(a|q) = p_{TM}(a|q) \times p_{LM}(a|q) \times r(a) \tag{6.18}$$

6.3.2 Cited Author Prediction

We examine in this task the capability of our model in predicting the authors that a given paper might cite in the future. Instead of predicting the cited papers directly, we predict the cited authors. This has applications in real life, since we sometimes follow some authors, especially some authors who are of high reputation in a certain field, and then by going through their publications, an author can locate the most recent and relevant papers to cite.

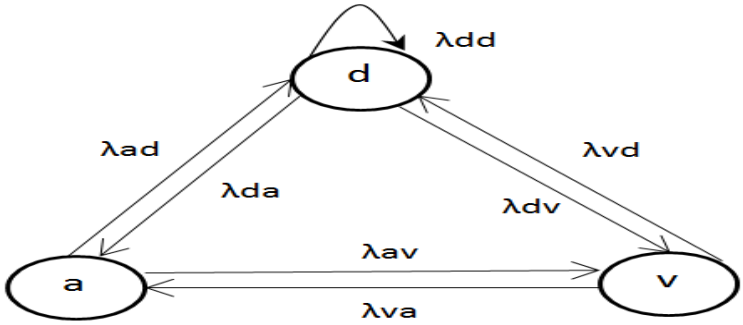

Figure 6.3: Heterogeneous Academic Network

Suppose we are now given a new document, represented by $\boldsymbol{W_d}$, and suppose we know its author lists $\boldsymbol{a_d}$. In order to predict the potentially cited authors, we need to compute the probability of $p(c|\boldsymbol{w_d})$, the probability of generating c given words $\boldsymbol{W_d}$ and author lists $\boldsymbol{a_d}$. This probability can be computed by making use of the distributions we learned from the training set. We have:

$$\begin{aligned}
p(c|\boldsymbol{W_d}) &= \sum_z \int p(c,z,\theta|\boldsymbol{W_d})d\theta = \sum_z \int \frac{p(c,z,\theta,\boldsymbol{W_d})}{p(\boldsymbol{W_d})}d\theta \\
&\propto \sum_z \int p(c,z,\theta,\boldsymbol{W_d})d\theta \\
&= \sum_z \int p(\boldsymbol{W_d}|z)p(c|z)p(z|\theta)d\theta \\
&= \sum_z p(\boldsymbol{W_d}|z)p(c|z)\int p(z|\theta)d\theta \\
&= \prod_{w\in\boldsymbol{W_d}} [\sum_z \sum_{a\in\boldsymbol{a_d}} p(w|z)p(c|z)\int p(z|\theta)d\theta] \\
&\approx \prod_{w\in\boldsymbol{W_d}} [\frac{1}{|a_d|}\sum_{k=1}^{K}\sum_{a\in\boldsymbol{a_d}} \theta_{ak}\phi_{kw}\varphi_{kc}] \qquad (6.19)
\end{aligned}$$

where, $a \in \boldsymbol{a_d}$.

6.3.3 Venue Prediction

In the task of venue prediction, we aim to predict the potential publishing venue given a paper with both its content and author lists provided. This task is of importance to some researchers, especially researchers that are new to a domain. They may find it difficult to decide where to submit after they finish their work. Similarly, in order to predict the potential venue, we need to compute the probability of $p(v|\boldsymbol{w_d})$. The derivation process is similar to that of the cited author prediction, and therefore we have:

$$\begin{aligned}
p(v|\boldsymbol{W_d}) &= \sum_z \int p(v,z,\theta|\boldsymbol{W_d})d\theta = \sum_z \int \frac{p(v,z,\theta,\boldsymbol{W_d})}{p(\boldsymbol{W_d})}d\theta \\
&\propto \sum_z \int p(v,z,\theta,\boldsymbol{W_d})d\theta \\
&= \sum_z \int p(\boldsymbol{W_d}|z)p(v|z)p(z|\theta)d\theta \\
&= \sum_z p(\boldsymbol{W_d}|z)p(v|z)\int p(z|\theta)d\theta \\
&= \prod_{w\in\boldsymbol{W_d}} [\sum_z \sum_{a\in\boldsymbol{a_d}} p(w|z)p(v|z)\int p(z|\theta)d\theta] \\
&\approx \prod_{w\in\boldsymbol{W_d}} [\frac{1}{|a_d|}\sum_{k=1}^{K}\sum_{a\in\boldsymbol{a_d}} \theta_{ak}\phi_{kw}\vartheta_{kv}] \qquad (6.20)
\end{aligned}$$

where, $a \in \boldsymbol{a_d}$.

6.4 Experimental Evaluation

6.4.1 Experimental Setup

In order to demonstrate the effectiveness of our model, we carried out a set of experiments on two real world data sets: the ACM data set and the ArnetMiner data set (see introduction in Section 2.4). The ACM data set of is composed of 172,890 papers, 170,897 authors, and 2,197 venues, and the ArnetMiner data set is composed of 1,558,415 papers, 795,385 authors and 6,010 venues.

Table 6.2: Statistics over ACM and ArnetMiner data set

Data Set	Paper	Author	Venue	Distinct Word	Word Tokens
ACM	92,708	2,965	1,816	17,341	6,224,821
ArnetMiner	165,330	14,454	2,304	18,151	13,368,826

For computational efficiency concern, we further carried out a filtering process to remove data noise, and to obtain a smaller subset of both data sets for experiments. We collect for two data sets the papers that have complete information, i.e, title, abstract and venue. Moreover, the papers we collect should have at least one available author and at least one citation. This results in a collection of 92,708 papers for the ACM data set, and 165,330 papers for the ArnetMiner data set. We further collect authors that have at least one publication and have been cited ten times as minimum, resulting in a set of 2,965 authors and 14,454 authors for ACM and ArnetMiner data sets. We finally filter out the stop words in paper content, and collect sets of 17,341 and 18,151 distinct words for ACM and ArnetMiner respectively that have a word frequency in the entire corpus greater than ten. Table 6.2 shows a brief summary of the two data sets we use for experiments.

6.4.2 Experimental Methodology and Results

We report in this section results over several groups of experiments. We compare our results with several other state-of-the-art baseline algorithms, and provide analysis for the results.

Qualitative Topic Modeling Results

We are interested in examining the modeling results in terms of the four probability distributions we define in the model. In the experiments for both ACM and ArnetMiner data set, we pre-fixed the number of topics to be 50. In this section, we report the top 10 returned words, authors, cited authors, and venues based on their topic-based distributions for one randomly chosen latent topic for ArnetMiner data set as one example.

As shown in Table 6.3, we can observe cohesive and interpretable results. For topic 12, which concerns 'information retrieval'-related research as concluded from the top returned words, we can identify several well-known scientists in this field from both the top 10 author list and cited author list. For example, the top cited author, Prof. Gerard Salton, is regarded as a founding scientist in the field of information retrieval, and the SIGIR Award outstanding contributions in IR research is named after him. The top returned author, Prof. Norbert Fuhr, was presented the Salton Award in 2012 due to "his pioneering, sustained and continuing contributions to the theoretical foundations of information retrieval and database systems."

Table 6.3: Topic Modeling Results on ArnetMiner data set

ArnetMiner data set Topic (Information Retrieval)			
Top 10 Words	Top 10 Authors	Top 10 Cited Authors	Top 10 Venues
information	Norbert Fuhr	Gerard Salton	sigir
based	Christopher Manning	W Croft	cikm
web	Jaap Kamps	Hector Molina	world wide web
paper	Kathleen Mckeown	Ricardo Baeza-Yates	acl
search	Gary Lee	Berthier Neto	inf process manage
results	Jian Nie	Justin Zobel	coling
retrieval	Eiichiro Sumita	Fernando Pereira	jcdl
model	Jamie Callan	John Lafferty	jasist
using	Jimmy Lin	Clement Yu	computational linguistics
user	Vimla Patel	Andrew Mccallum	emnlp

Expert Ranking

(1). Evaluation Ground Truth

It has long been acknowledged as one of the problems in expert ranking research that the community lacks both standard query collections and benchmarks for evaluation. Much previous research resorts to human labeling, which is naturally subjective and biased, and is also time-consuming. In this chapter, we make use of other evidence and carry out two kinds of evaluations. In the first approach

(GT1), we use historical information regarding award winners provided by 16 SIG communities as supporting ground truth. We assume that these award winners are nominated and selected by other researchers in an open and objective way. They are widely acknowledged in their community to have made outstanding contributions in their research fields, and have established world-wide reputations. The corresponding query is generated based on the main research area of that community; for example, the query for SIGIR community is 'information retrieval'. We also check the generated queries with the 23 categories provided by Microsoft Academic engine, and make sure that each query corresponds to one category. We assume that these queries cover the main disciplines of computer science research, and that they represent reasonable topics that users might use for information. These queries are intended to be broad queries.

In the second evaluation approach (GT2), we make use of a benchmark data set with seven queries and expert lists provided by Zhang et al. [204].[1] The expert lists are generated by pooled relevance judgments together with human judgments. Specially, for each query, the top 30 results from three main academic search engines (Libra, Rexa, and ArnetMiner) are collected and merged then further judged by one faculty professor and two graduate students. These queries are more specific queries.

We utilize the traditional IR evaluation metric MAP. We list the query and their corresponding number of experts in Table 6.4.

(2). Topic Modeling Results

We report the experiment results comparing the performance of our ACVT model with the ATM model [147], the CAT model [173], the ACT [168] model, and the ACTC [180] model which is the most recently published work extending ACT [168].

For ACTC [180] model, additional latent variable 'subject' is introduced, and there is no direct author-topic distributions. Instead, each author would be associated with a multinomial distribution over multiple subjects, which have distributions over topics and conferences respectively. There also exists a distribution for topics

[1]This data is available online at http://arnetminer.org/lab-datasets/expertfinding/ (the New People Lists).

Table 6.4: Evaluation Benchmark

Benchmark 1: SIG Community Award Winner	
Query	Expert No.
algorithm theory	7
security privacy	4
hardware architecture	27
software engineering	15
programming language	19
artificial intelligence	14
data mining	7
information retrieval	9
graphics	12
human computer interaction	10
multimedia	2
network communication	18
operating systems	9
database	18
simulation	3
computer education	28
Benchmark 2: ArnetMiner New Expert Lists	
intelligent agents	30
information extraction	20
semantic web	45
support vector machine	31
planning	35
natural language processing	43
machine learning	41

over words. Under this model, the expertise ranking scheme can be described as:

$$P(a|q) = \prod_{w_i} \sum_{s_j} \sum_{z_t} P(a|s_j)P(s_j|z_t)P(z_t|w_i) \tag{6.21}$$

In our experiments, we set the number of latent topics to be 50, and the number of latent subjects to 20 for the ACTC [180] model. For the four hyper-parameters, we set $\alpha = 2$, $\beta = 0.01$, $\gamma = 2$ and $\lambda = 2$. As indicated in the results, our ACVT model works the best in all scenarios and it significantly outperforms the other four models in both ACM and ArnetMiner data sets. Better results can be achieved with the ACVT model using the first benchmark than the second one in both data

sets. It can also be observed that under most circumstances, CAT, ACT and ACTC outperform the original ATM, except that working on ArnetMiner data set and using the second benchmark, ACT works slightly worse than ATM. ACTC works better than ACT, and CAT works better than both ACT and ACTC under most circumstances.

Working on ArnetMiner data set, we list in Table 6.5 the Top 10 ranked experts for query 'information retrieval' under five different topic models (ATM, ACT, CAT, ACTC and ACVT) combined with the query. As indicated in the results, we can achieve more valid results using CAT and ACVT than ATM, ACT and ACTC, since several well-known experts in information retrieval can be identified within Top 10, and ranked higher. Furthermore, ACVT can do even better than CAT, since all the returned experts are information retrieval concentrated researchers, while some of the top 10 returned experts by CAT are experts in other fields; for example, Prof.Jeffrey Ullman is famous for his research in compiler, theory of computation and database theory, and Prof.Jennifer Widom is also a well-known database researcher who has won the SIGMOD award in 2007.

Table 6.5: Comparison of Topic Modeling Results: MAP

ACM data set					
	ATM	CAT	ACT	ACTC	ACVT
GT1	0.0288	0.0688	0.0513	0.0562	**0.1802**
GT2	0.0269	0.0791	0.0780	0.0785	**0.1490**
ArnetMiner data set					
	ATM	CAT	ACT	ACTC	ACVT
GT1	0.0156	0.0919	0.0514	0.0685	**0.1485**
GT2	0.0508	0.0552	0.0673	0.0730	**0.1135**

(3). Combine with Language Model and Random-Walk methods

We examine in this section whether the performance can be improved if we combine topic modeling with a language model-based approach and a random-walk based approach. We report the results for expert ranking in terms of using a language model, a random-walk based method and topic modeling separately, as well

Table 6.6: Expert Ranking Results Comparison (on ArnetMiner data set)

Query: information retrieval				
ATM	ACT	ACTC	CAT	ACVT
Jintao Li	C Giles	Edward Fox	Gerard Salton	W Croft
Ling Duan	Wei-ying Ma	C Giles	Ricardo Baeza-Yates	Gerard Salton
Simone Tini	Ji Wen	Marcos Alves	W Croft	Ricardo Baeza-Yates
Stanley Jr	Maarten Rijke	W Croft	Hector Molina	Hector Molina
Sunil arya	Jian Nie	Berthier Neto	Jiawei Han	Berthier Neto
Karthikeyan Sankaralingam	Irwin King	Maarten Rijke	Rakesh Agrawal	Jiawei Han
Si Wu	Alan Smeaton	Jian Nie	Berthier Neto	Justin Zobel
Cleidson Soua	Chengxiang Zhai	Min Kan	Hans Kriegel	Fernando Pereira
Shi Neo	Rohini Srihari	Mounia Lalmas	Jeffrey Ullman	C Giles
Osman Unsal	W Croft	Mark Sanderson	Jennifer Widom	Wei-ying Ma

as the combined results.

As introduced in section 4.1.2, we adopted two combination methods. For linear combination, we take use of a simple greedy search method to tune the parameters. We gradually change the weight for one particular method from 0 to 1, and let the other two methods evenly share the remaining weights, i.e. ($\alpha \in [0, 1]$, $\beta = \gamma = (1 - \alpha)/2$). Figure 6.4 and Figure 6.5 depict the results working on ACM data set using GT1 as the ground truth, and ArnetMiner data set using GT2 as the ground truth respectively. Table 6.6 indicates the results by the multiplication combination method.

Several observations can be made from the results. 1) We can achieve better performance when combining the three methods by multiplication than linearly

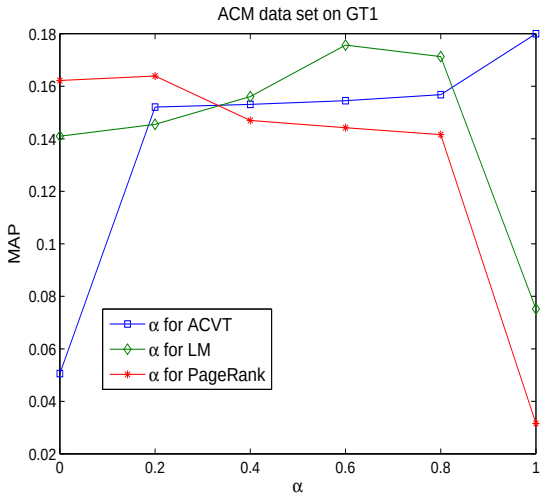

Figure 6.4: Combine ranking methods (ACM data set)

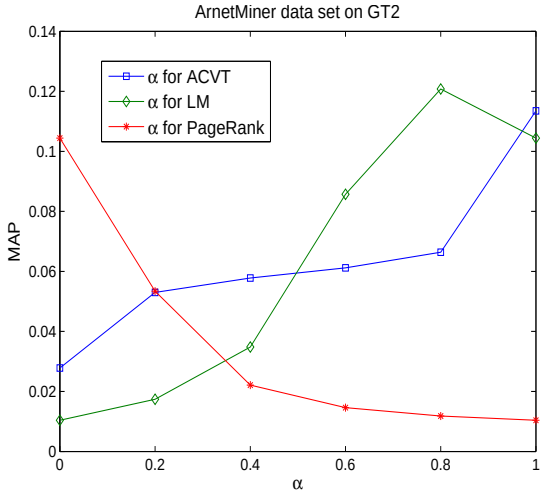

Figure 6.5: Combine ranking methods (ArnetMiner data set)

combining them. The best performance under linear combination is always outperformed by multiplication method. This is also true for working on ACM data set with GT2 ground truth, and ArnetMiner data set with GT1 as ground truth. 2) Our ACVT model works better than both the language model and random-walk

Table 6.7: Comparison of Topic Modeling Results: MAP

ACM data set				
	LM	PR	ACVT	LM+PR+ACVT
GT1	0.0752	0.0316	0.1802	**0.1863**
GT2	0.1242	0.0129	0.1490	**0.1529**
ArnetMiner data set				
	LM	PR	ACVT	LM+PR+ACVT
GT1	0.0258	0.0107	0.1485	**0.1750**
GT2	0.1044	0.0104	0.1135	**0.1676**

PageRank-based approach in all experimental scenarios. 2) The language model approach works the second best, and its performance is much better under the first benchmark than the second benchmark. 3) We can achieve improved performance when combing the three approaches together than working on any of them individually. The relative improvement over plain ACVT is 3.45% (ACM under GT1), 2.62% (ACM under GT2), 17.85% (ArnetMiner under GT1) and 47.67% (ArnetMiner under GT2) respectively.

6.4.3 Cited Author Prediction

In this section, we consider the capability of our ACVT model in predicting the authors that any given paper might cite. We take the CAT model as our baseline algorithm, in which cited author information is modeled yet the venue information is missing. In experiments, we select 10,000 papers for the ACM data set, and 18,000 papers for the ArnetMiner data set, as our two testing sets, corresponding to roughly 10% of the total papers in each data set. The criterion for such a selection is that we make sure that the authors of each paper in the testing set has at least one other paper publication in the remaining training set.

Predictions are made by following Equation 6.19. The actual set of cited authors for each test paper serves as our ground truth. We evaluate our performance in terms of MAP, as shown in Table 6.8 and Precision@K, as shown in Figure 6.6.

As shown in Table 6.8, we can achieve a 12.15% and 34.07% improvement in

Table 6.8: Comparison of Cited Author Prediction: MAP

Data Set	CAT	ACVT
ACM	0.1029	**0.1154**
ArnerMiner	0.0364	**0.0488**

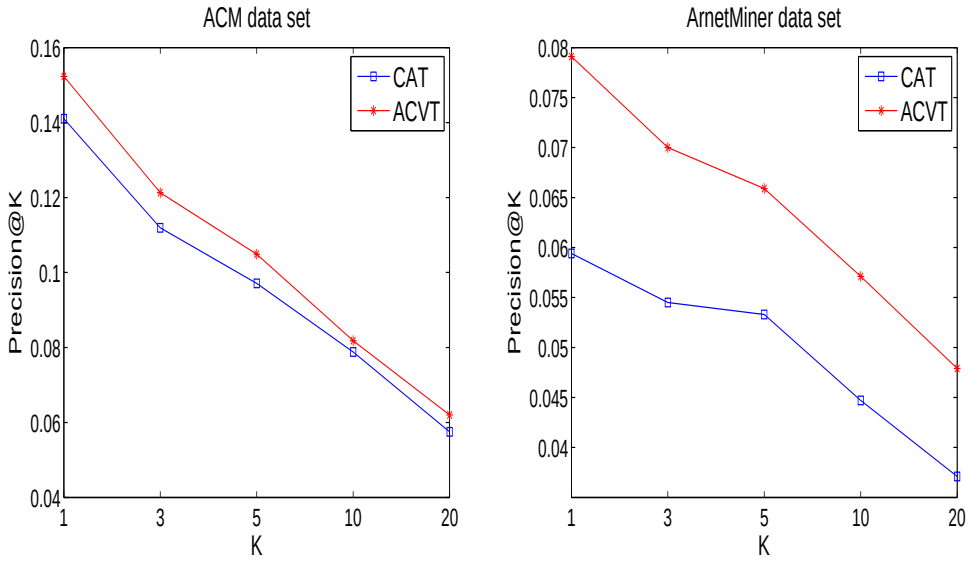

Figure 6.6: Cited Autor Prediction: Precision@K

MAP when using our ACVT model as compared to the CAT model in ACM and ArnetMiner data sets respectively. These demonstrate our model to be more effective in predicting cited authors, and indicate that jointly modeling venue information can provide more cohesive author-topic and topic-cited author associations.

We observed consistent performance in terms of Precision@K across two data sets. Even though the value of Precision@K keeps dropping when K is increased, ACVT outperforms CAT at all different K values. We further notice that there is greater improvement for ACVT over CAT on ArnetMiner data set than ACM data set. For both data sets, the improvement of ACVT over CAT decreases with larger K value.

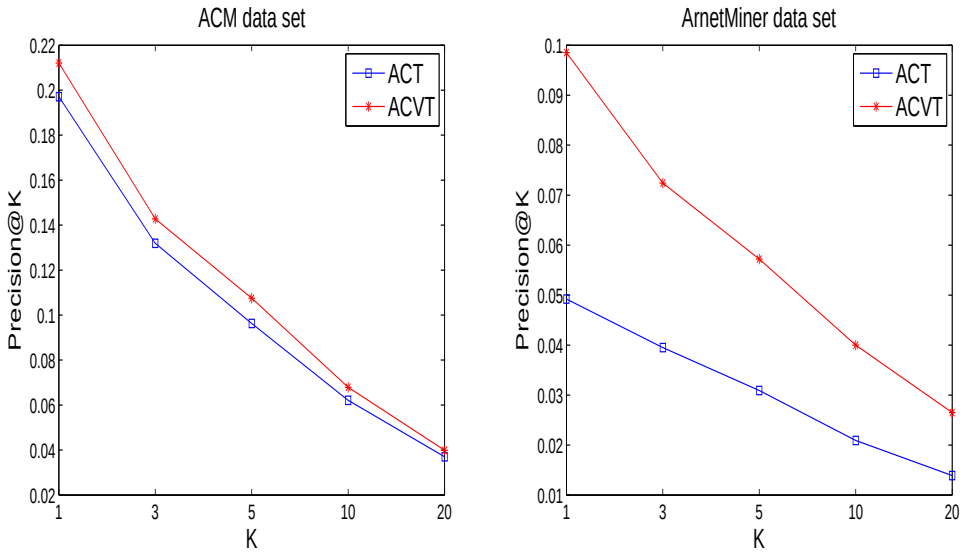

Figure 6.7: Venue Prediction: Precision@K

Table 6.9: Comparison of Venue Prediction: MAP

Data Set	ACT	ACVT
ACM	0.3226	**0.3585**
ArnerMiner	0.1151	**0.1977**

6.4.4 Venue Prediction

We now evaluate the capability of our ACVT model to predict the publishing venue of a given paper. We take the ACT model as our baseline algorithm in which the venue information is modeled yet the cited author information is missing. Similar to the experiments for cited author prediction, we select 10,000 papers and 18,000 papers from ACM and ArnetMiner data sets respectively to work as our testing sets, and make sure that the authors of those chosen papers have at least one other paper in the remaining training sets.

We can perform venue prediction by following Equation 6.20, and evaluate the results by comparing with the real publishing venue of the given paper.

As demonstrated in Table 6.9, our ACVT outperforms the ACT model in predicting the publishing venues of any given paper. The improvement of ACVT over ACT is 11.13% for ACM and 71.76% for ArnetMiner. This demonstrates the advantage of jointly modeling multiple facets.

Figure 6.7 shows the performance in terms of Precision@K. We observe similar trend as in the task of cited author prediction. Our ACVT model can outperform the ACT model under all different K values, and we can achieve greater improvement on ArnetMiner data set than on ACM data set.

6.5 Bibiliographic Notes

6.5.1 Author Topic Modeling

Generative topic modeling is a popular unsupervised learning technique for topic-related content representation. Initially, this kind of generative modeling was utilized in pure contextual analysis. Two representative models of this kind, Probabilistic Latent Semantic Analysis (PLSA) [72] and Latent Dirichlet Allocation (LDA) [18], exploit co-occurrence patterns of words in documents and unearth the semantic clusters (topics) of words. In those proposed models, each document would be regarded as a mixture over multiple latent topics.

The original PLSA and LDA idea of document topic modeling has been extended to include modeling of authors' interests. The very first work in this direction is that of Rosen-Zvi et al. [147], which simultaneously models the content of documents and the interests of authors, such that the mixture weights for different topics would be determined by the authors of the documents.

Most recently, a number of models that extend the original idea of LDA and ATM have been proposed, most of which contribute in the direction of incorporating additional contextual information and integrating linkage structures. Link-LDA [46], Pairwise-LDA and Link-PLSA-LDA [128] are three representative topic models that extend PLSA and LDA by integrating citation linkages among papers into topic

modeling. However, in these three efforts, no author information has been considered, and the citation prediction is made based upon pairs of papers, which is quite different from the model we propose in this chapter that particularly emphasizes the interests and influence of authors.

Several representative works have been proposed to extend ATM. The Author-Conference-Topic (ACT) [168] model adds contextual information, the publishing venues of papers, to represent venues' influence over topics. The Author-Conference-Topic-Connection [180] model extends [168] by introducing an additional latent variable 'subject', from which the confereneces (venues) and topics can be respectively generated. The Citation-Author-Topic (CAT) [173] model directly models the cited authors' information, such that authors' influence over other authors can be considered. As a further extension to the CAT model, the Context Sensitive Topic Models [88] introduces a learning mechanism that can dynamically determine the citation context windows, and to associate terms within citation context windows to cited authors. Our proposed model, the ACVT model, can be regarded as a further extension and combination of the ACT and CAT model, in that we jointly model both the venue and the cited authors information, as compared to ACT which only considers venues, and CAT and the Context Sensitive model that only consider citations.

There are also other topic models which emphasize different aspects of contribution. Liu et al. [106] proposed a model that can jointly model topics, author communities and link information for author community detection. Johri et al. [83] introduced a model that can relax the 'bag-of-words' assumption and can automatically identify multi-word phrases into modeling; Mei et al. [119] conducted temporal author topic analysis, and Song et al. [162] built topic models to disambiguate names. Mei et al. [118] incorporated network regularization technique into an extended version of PLSA. Our ACVT model distinguishes itself from all the work mentioned above by its model design focus and applications.

6.5.2 Applications

Expert ranking has blossomed since the advent of the TREC Enterprise Track initiated in 2005, and the rapid development of online academic search engines, such as ArnetMiner and Microsoft Academic Search. Given a user query, the task of expert ranking basically involves identifying and ranking a list of researchers based on their expertise in that query-specific domain. Two categories of approaches have been the research focus in the past years: the pure content analysis based approach [8, 108, 50], which emphasizes evaluating authors' expertise by measuring the relevance of their associated documents to the query, and the social network based approach [39, 170, 62, 79], which evaluates authors' expertise by exploiting the social interaction of authors and other scientific facets, such as their co-authorships, their citations to other papers/authors, etc. Few prior works directly make use of topic modeling results for expert ranking. The CAT, ACT and ACTC models are the three most representative works we have identified.

Citation prediction has long been a research topic as a specific application in link prediction (e.g., [69, 68]). However, most of them predict citations among papers, and few use topic modeling results. In our paper, we focus on predicting the potential cited authors given a new document, which has seldom been explored by previous work except the work of Tu et al. [173].

In venue recommendation, a ranked list of venues is generated to which a given paper could be submitted. Two prior works [99, 136] particularly address such a problem, however, none of them makes use of a topic modeling approach.

6.6 Summary

We proposed in this chapter an extended probabilistic topic model (the ACVT model) that can jointly model authors, papers, cited authors and venues in one unified model. As compared to previous work, ACVT can provide a more complete framework to incorporate additional useful contextual information. It is therefore more applicable to multiple applications related to academic network analysis. We

have considered performance in three typical applications: expert ranking, cited author prediction and venue prediction. Experiments based on two real world data sets demonstrate that our model can identify more interpretable topic-related associations in terms of authors, cited authors, and venues, and can provide better performance in all three applications as compared with several baseline algorithms.

Chapter 7

Recommendation in Academia: a Joint Multi-Relational Model

In this chapter, we present an extended latent factor model that can jointly model several relations in an academic environment. The model is specially designed for four recommendation tasks: author-paper citation prediction, paper-paper citation prediction, publishing venue prediction and author-coauthor prediction, and is proposed based upon the assumption that several academic activities are highly coupled, and that by joint modeling, we can not only solve the cold start problem but also help in achieving more coherent and accurate latent feature vectors. Moreover, to facilitate ranking, we extend an existing work which directly maximizes MAP over one single tensor into a more generalized form and is therefore able to maximize MAP over several matrices and tensors. Experiments carried out over two real world data sets demonstrate the effectiveness of our model.

7.1 Introduction

People can conduct many activities in academic environment: publishing papers, collaborating with other authors, or citing other papers/authors. Theses activities are sometimes not easy to fulfill. For example, reading and therefore citing new

published papers is one of the most important tasks that a researcher should do for research, however, to find relevant and referential scientific literature from hundreds of thousands of publications is a time-consuming and labor-intensive task especially with the rapid development of Internet which makes published papers easy to be accessed. For another example, when a researcher finished writing a paper, it may be difficult for him to decide where to submit due to the large number of possible conferences and journals. To better facilitate such activities, information needs have arisen for developing systems that can automatically help to predict or recommend proper venues to submit, papers to cite, and authors to collaborate. In this work, we focus on the prediction task in academic environment, and particularly pay attention to the following four tasks: the prediction on publishing venues, collaborators, cited papers for authors, and cited papers for papers.

Even though individual systems or algorithms have been proposed to tackle each of the four tasks separately, which we will review in later sections, limitations still remain. Most of the previous methods only focus on one single type of relationship while neglect to explore the mutual interaction among different relationships. In a real complicated academic environment, which often consists of heterogeneous nodes and links, each scientific factor can play different roles, and participate in different activities. For example, individual researcher can work as an author to write paper, as a collaborator to work with another researcher, or to be cited by another researcher. The entire academic network is composed of multiple relations that mutually affect each other.

To better model this multi-relational academic activities and to provide good recommendations, several challenges remain:

- **coupled high order data**: as mentioned above, there are multi-typed scientific entities in the academic environment, playing different roles and participating in different activities. These activities are often coupled. It is quite natural for a paper that has a large number of citations from other papers to be cited by more authors and that authors who collaborate more frequently may tend to have the same set of paper citations. It is necessary to incorporate

other correlated relations when making prediction on one specific relation.

- **cold start problem**: the cold start problem is a typical problem in recommender systems. Take the task of citation recommendation for papers as one example, some most recently published papers will hardly be cited since they have never been cited before by other papers or authors, even though they are highly relevant to a topic or may have great contribution in a certain field.

- **Personalization support for authors:** Researchers play an important role in many activities, and they may have different preferences in selecting which paper to cite, or which venue to submit, even though those papers or venues focus on similar topics.

- **Interest evolution for authors:** The interest of researchers evolves over time. Even though they keep on working in one research field, their research focus and methods may change.

To tackle these challenges, we propose a joint multi-relational model referred as the JMRM model which directly models several groups of coupled activities in the academic environment and provide a more general framework that can solve several prediction tasks simultaneously in a unified way.

Our model is fundamentally the latent factor collaborative-filtering(CF)-based model, in which each relation can be represented as a matrix or higher-dimensional matrix. However, the following three characteristics distinguish our model from previous ones. **Firstly**, our model is composed of multiple matrices or tensors, each of which indicates one relation in the academic environment, and are highly coupled with each other. **Secondly**, we integrate the temporal information into the generation of several matrices to better reflect the evolution of authors' preferences; **Thirdly**, we choose the objective function for solving the model as maximizing the mean average precision (MAP) as compared to most of the previous work minimizing the predicting error (RMSE). MAP is a standard IR evaluation metric which provides a single-value measure of quality across all recall levels. It is widely used due to its good discrimination and stability property. More important, MAP is

a ranking-based measure for which errors at top of the ranking list will lead to a higher penalty than errors at lower places of the ranking list. This top-heavy biased property makes MAP particularly suitable to work as the objective function for recommender systems, since most people will only pay attention to the top ranked results in the recommendation list. For this reason, we choose to maximize MAP as our objective function.

To sum up, the main contributions of our work are as follows:

- we propose a joint multi-relational model which integrates several coupled relations in an academic environment presented as matrix or higher dimensional matrix in a unified way. This model is particularly designed for four recommendations: the prediction task on paper submission for venues, co-authorship prediction, paper citation prediction for authors, and paper citation prediction for papers.

- we choose to maximize MAP as the objective function for solving the model, and extend the tensor factorization approach optimizing MAP into a more general framework that can maximize MAP for coupled multiple matrices and tensors.

- experimental evaluation over two real world data sets demonstrate the capability of our model in four recommendation tasks, as they show improved performance as compared to three state-of-the-art algorithms.

We report preliminary experiments in analysis of the main challenges in section 7.2. We address the model design in section 7.3, and introduce the algorithm solving the model in section 7.4. Experimental evaluation is reported in section 7.5. We review related work in section 7.6 and conclude this chapter in section 7.7.

7.2 Preliminary Experiments

In this section, we conducted some simple experiments on two real world data sets: the **ACM data set** and **ArnetMiner data set** (see introduction in Section 2.4)

to analyze the characteristics of activities and relationships among scientific factors in the academic environments.

7.2.1 Data Sets

The ACM data set is composed of 172,890 papers, 170,897 authors, and 2,197 venues. Papers within this data set are published between 1951 and 2009. The ArnetMiner data set is composed of 1,558,415 papers, 795,385 authors and 6,010 venues; papers in ArnetMiner are published between 1936 to 2011.

7.2.2 Coupled Relations

We are first interested in finding out whether multiple relations in an academic environment are coupled. As a simple test example, we compute for each author in both data sets his/her total number of publications, citations and coauthors, and evaluate the correlation between these three factors. Figures 7.1 and 7.2 show our results.

As we can see, there exists an obvious linear positive correlation between number of publications and coauthors, indicating that under most circumstances, the more coauthors you have, the more publications you can achieve. This observation is compatible with our common sense. However, the correlation between publication number and citation number is not so obvious. As shown in Figure 7.2, we have many data points scattered in the lower-left corner of the figure, indicating that some authors who do not publish many papers can also achieve high citation.

7.2.3 Cold Start Problem

We evaluate the changes in papers' attracting citation to demonstrate the existence of cold start problem in the academic environment. We average the number of citations each paper retrieves in both data sets on a yearly basis. This simple statistical result, as shown in Table 7.1, indicates that averagely a newly published paper begins to retrieve citations 2 more years later than its publication. However,

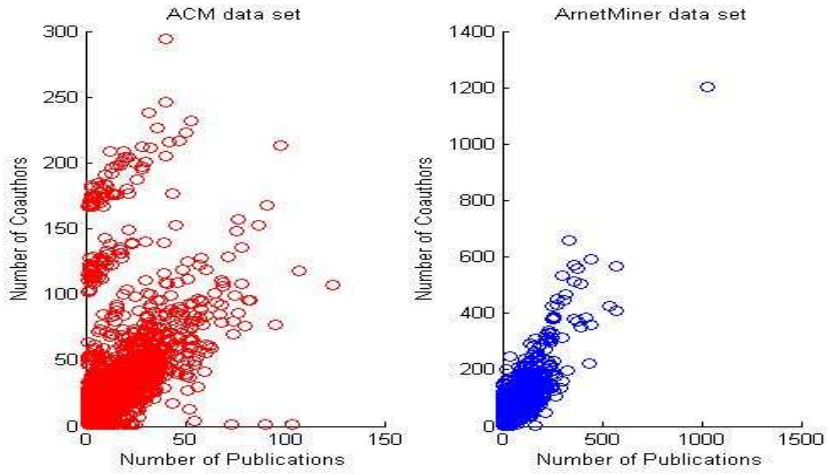

Figure 7.1: Correlation between Number of Publications and Coauthors

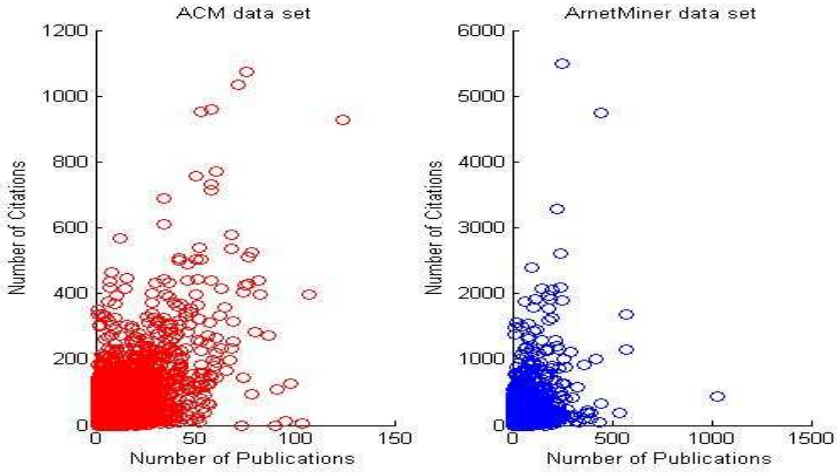

Figure 7.2: Correlation between Number of Publications and Citations

after that, it just costs around 0.97 years and 0.85 years for papers in ACM and ArnetMiner data set to retrieve one new citation. Another simple statistics, as shown in Figure 7.3, indicates that papers on average can achieve most of their citations in the following year of its publication, and that number gradually drops as time evolves.

Table 7.1: Statistics on Papers' Citations

Data set	No. of Papers	First Citation after publication	Avg. Citation Frequency
ACM	55,392	2.0350	0.9693
Arnet	315,831	2.7599	0.8528

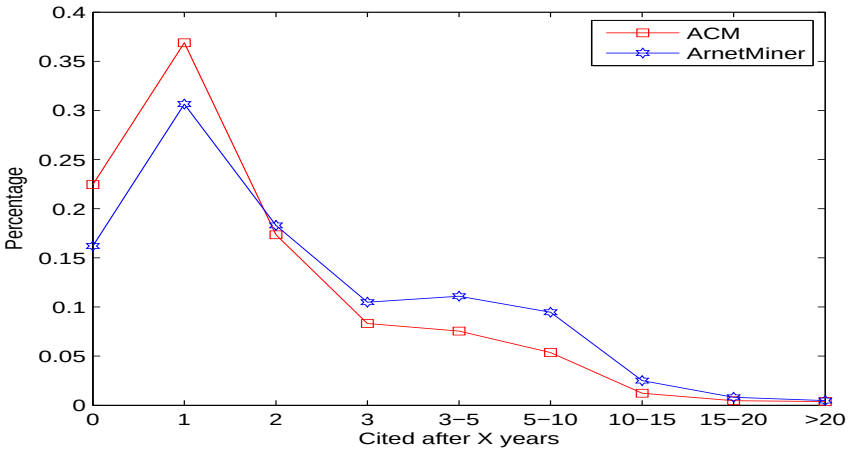

Figure 7.3: Average number of citations change over time

7.2.4 Interests Evolution

We evaluate the evolution of authors' research interests by checking the changes in their publishing venues. For ACM data set, we collect for each author the publishing venues of his papers published before 2003 and after 2003 (including 2003) as two sets, and adopt the Jaccard Similarity method to detect the similarity/difference between these two sets. For ArnetMiner data set, we set the year point as 2006. We choose the year point by guaranteeing that the average number of distinct venues of authors in each separate data set is equivalent before and after that year point. Table 7.2 shows the results.

As indicated, the average Jaccard values for both data sets are pretty small, indicating that authors have a diversified publishing venue list. Authors chose different venues to submit, indicating that their research focus may evolve over time.

Table 7.2: Statistics on Changes of Publishing Venues

	ACM ($Y = 2003$)	**ArnetMiner** ($Y = 2006$)
No. of Authors	23,358	188,143
Avg No.Venues before Y	2.73	5.14
Avg No.Venues after Y	2.75	5.09
Avg Jaccard Similarity	0.0946	0.1188

7.3 Joint Multi-Relational Model (JMRM): Model Design and Generation

Inspired by the information needs for developing recommender systems in the academic environment and in order to fulfill the challenges, we propose a joint multi-relation model. Our model is designed for four particular recommendation tasks in the academic environment, each of which represents one academic activity, and induces one relation. Therefore, we have four main relations in the model: the author-paper-venue relation (represented as the APV-tensor), author-author-collaboration relation (AA-matrix), author-paper-citation relation (AP-matrix), and paper-paper-citation relation (PP-matrix). Figure 7.4 shows the framework of the model. In order to deal with the cold start problem and better support authors' personalization, we further incorporate additional features for papers and authors. In the current work, we only consider the pure paper content as paper features, and we use the PW-matrix to represent it. We model authors and their features as the AF-matrix, and will introduce more detailed features for authors in the following section.

APV-tensor: the author-paper-venue relation This three-order relation consisting of triples <author-paper-venue> indicates the publishing venue selection for papers with known authors. We have: $APV(a_i, p_j, v_k) = 1$ if paper p_j written by author a_i is published in venue v_k; Otherwise, $APV(a_i, p_j, v_k) = 0$.

AP-matrix: the author-paper-citation relation The AP matrix models the citation relationship between authors and papers. An author may cite a paper

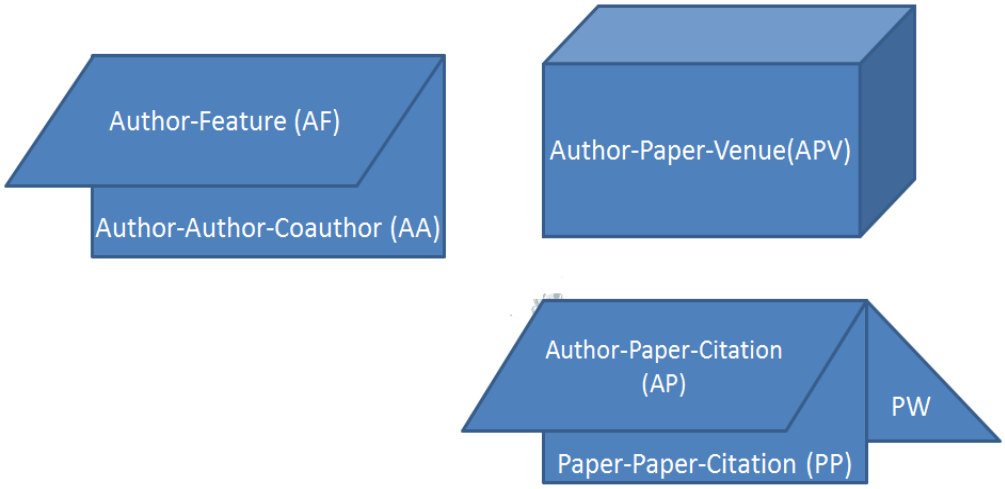

Figure 7.4: Coupled Matrices and Tensor

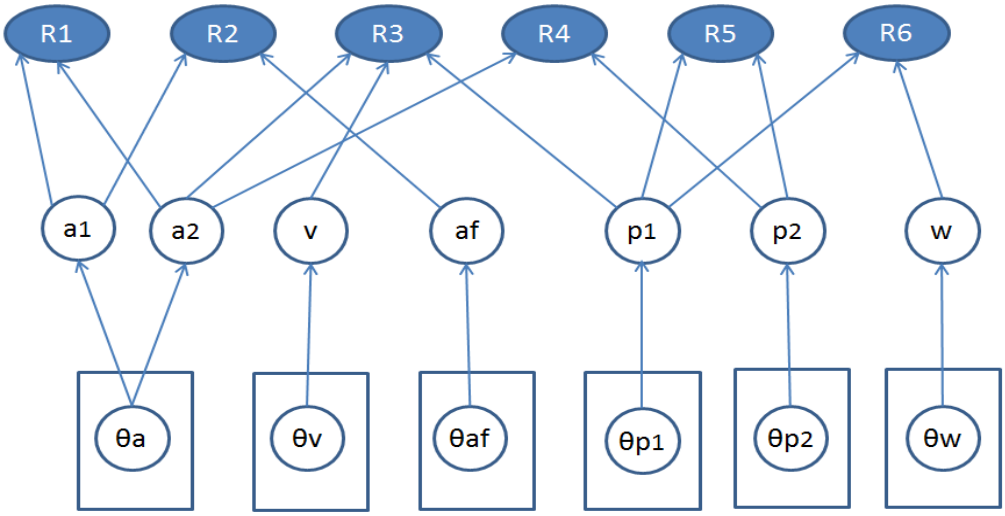

Figure 7.5: Graphical Representation of the Model. R1: AA-relation; R2: AF-relation; R3: APV-relation; R4: AP-relation; R5: PP-relation; R6: PW-relation

multiple times at different times, and the preference of the author over papers may also change over time. In order to model this temporal evolution property, we first generate a three-dimensional tensor incorporating the temporal factor as the

third dimension, and then collapse the tensor into a two-dimensional matrix by aggregating the number of citations at different years with a time decay function. Given an author a_i, and a paper p_j cited by a_i, the number of times p_j is cited by a_i on year t_k (the value for entry $< a_i, p_j, t_k >$) can be retrieved as:

$$E(a_i, p_j, t_k) = \sum_{p_{ai} \in \boldsymbol{p_{ai}}} \delta(y(p_{ai}) = t_k \wedge p_j \in \boldsymbol{c_{p_{ai}}}) \tag{7.1}$$

where p_{ai} is any paper published by a_i, $\boldsymbol{p_{ai}}$ is the publication set of a_i. $\boldsymbol{c_{p_{ai}}}$ is the set of all cited papers of p_{ai}. Function $y(p_{ai})$ retrieves the publication year of p_{ai}, and $\delta(c)$ is a function which returns 1 if condition c is satisfied and 0 otherwise. We aggregate the citations at different time points based upon the hypothesis that authors' interests decay over time, and therefore more recent citation contribute more heavily than older citation. We penalize the old citations by introducing an exponential decay kernel function. The entry $< a_i, p_j >$ for the collapsed author-paper matrix can thus be defined as:

$$E_{AP}(a_i, p_j) = \sum_{t_k=T_1}^{T_2} e^{-\beta(T_2 - t_k)} \cdot E(a_i, p_j, t_k) \tag{7.2}$$

where T_1 and T_2 are the earliest and last year for paper publication in the data set, and β is the decay rate.

AA-matrix: the author-author-collaboration relation The AA-matrix indicates the collaboration, an important social interactions between pairs of authors. Similar to the changing interests' of authors over papers, researchers may also change to work with others researchers in different time periods. We follow the same procedure as introduced for the AP-matrix generation by first constructing the author-author-time tensor, and then collapse it into author-author matrix. The entry for $< a_i, a_j >$ can thus be determined by:

$$E_{AA}(a_i, a_j) = \sum_{t_k=T_1}^{T_2} e^{-\beta(T_2 - t_k)} \cdot E(a_i, a_j, t_k) \tag{7.3}$$

where $E(a_i, a_j, t_k)$ is the number of times author a_i collaborates with a_j on year t_k.

PP-matrix: the paper-paper-citation relation The generation of the PP-matrix is different from that of the AP-matrix or AA-matrix, since each paper can only cite another paper once. However, there also exists temporal influence, as a paper may cite a paper published long time ago, or a more recent one. Suppose we have three papers $p1$, $p2$ and $p3$, published in $y1$, $y2$ and $y3$ respectively ($y1 \leq y2 \leq y3$), and we have paper $p3$ cites $p2$ and $p1$. In our work, we assume that $p2$ will have a greater contribution in presenting the topic interests or preferences for $p3$ than $p1$, since in publishing papers, we often need to review and compare with those most recently published and state-of-the-art papers. With this assumption, we have for each entry $< p_i, p_j >$ indicating that paper p_i cites p_j in the PP-matrix as:

$$E_{PP}(p_i, p_j) = e^{-\beta(y(p_i)-y(p_j))} \tag{7.4}$$

where $y(p_i)$ and $y(p_j)$ returns the publishing year for p_i and p_j respectively.

PW-matrix: the paper-word relation PW-matrix indicates the features of papers. In current work, we only consider the pure content of papers, and therefore we collect the top returned words in the data set with higher frequency. Each entry of $< p_i, w_j >$ indicates the term frequency of word w_j in paper p_i.

AF-matrix: the author-feature relation We identify 20 distinctive features for authors listed in Table 7.3 to represent the personalized property of an author from three aspects, and we introduce them as follows.

Simple bibliographic based features: We adopt a set of simple bibliographic features. These include:

- **total publication number (totalPubNo):** which indicates the total number of publications of one author, across different research domains.
- **total citation number (totalCitNo)**: which indicates the total number of

Table 7.3: Author Features

Feature Category	Feature
Simple bibliographic	total publicationNo total citationNo H-index [70] G-index [47] Rational H-index distance [148] Rational H-index X [148] E-index [202] Individual H-index [13] Normalized individual H-index [65]
Network-based	PageRank score on coauthor network PageRank score on citation network
Temporal-based	CareerTime [191] LastRestTime [191] PubInterval [191] Citation Influence Ratio [191] Contemporary H-index [158] AR-index [80] AWCR-index [65] Avg Publication number Avg Citation number

citations an author received from other papers published in different domains.

- **H-index**[70]: H-index is the most well-known measurement in evaluating a researcher's expertise. A researchers is said to have an H-index with size h if h of his or her total papers have at least h citations each. This index is affected by the number of citation that a researcher has and the citation distribution among a researcher's various papers.

- **G-index**[47]: G-index is another primarily used measurement. The G-index value is the highest integer (g) such that all the papers ranked in Position 1 to g in terms of their citation number have a combined number of citations of at least g^2.

- **Rational H-index distance (HD-index)**[148]: this variant of H-index calculates the number of citations that are needed to increase the H-index by 1 point.

- **Rational H-index X (HX-index)**[148]: the original H-index indicates the largest number of papers an author has with at least h citations. However, a researcher may have more than h papers, for example, n papers, that have at least h citations. If we define $x = n - h$, then the HX-index is calculated by $HX = h + x(s - h)$, where s is the total number of publications an author has.

- **E-index**[202]: the original H-index only concentrates on the set of papers an author published, each of which has at least h citations. This set of papers is often referred to as the *h-core* papers of an author. By using this measurement, the only citation information that can be retrieved is h^2, i.e., at least h^2 citations of an author can be received. However, the additional citations for papers is the $h - core$ would be completely ignored. To complement the H-index for the ignored excess citations, $E - index$ is proposed, which can be computed by $e^2 = \sum_{j=1}^{h}(cit_j - h) = \sum_{j=1}^{h} cit_j - h^2$, where cit_j are the citations received by the j^{th} paper in the $h - core$ set. We can further have $E - index = sqrt(e^2)$.

- **Individual H-index IH-index**[13]: this measurement is proposed to reduce the effects of co-authorship. It can be computed by dividing the standard H-index by the average number of authors in the *h-core* set: IH-index= h^2/N_a^T, N_a^T is the total number of authors in *h-core* set.

- **Normalized Individual H-index NIH-index**[65]: this measurement is also proposed to reduce the coauthor's effect, but is much finer-grained than the previous one. To compute it, we can firstly normalize the number of citations for each paper in the *h-core* by dividing the number of its citation by its number of authors. Then we compute the H-index score based on these normalized citation counts.

Network based features: this group of features measure how well an author collaborate with other authors, and how their publications influence other authors. We construct two types of network, and apply the PageRank algorithm to compute the authors' authority scores. The networks we considered are:

- **Coauthor Network**: this network is generated by connecting authors by their coauthor-relationships. For the sake of PageRank algorithm, we convert one non-directional edge into two directional edges. As a result, one non-weighted edge would exist from author a_i to author a_j and from author a_j to author a_i if they have written at least one paper together.

- **Citation Network**: this directed network is generated by connecting authors by their citations. One non-weighted edge would point from author a_i to a_j if at least one publication of author a_i cites one paper of author a_j.

Temporal features: this group of features measures authors' authority by some temporal characteristics associated with them. These include:

- **CareerTime**: this measures how long a researcher has devoted into academic research? We assume that the longer career time a researcher has, the higher authority he may have.

- **LastRestTime**: this indicates how many years have passed since the last publication of a researcher. We assume that a long time rest without academic output will negatively affect a researcher's academic reputation.

- **PubInterval**: this measures how many years on average would a researcher take between every two consecutive publications. We assume that more frequent publication indicates more active academic participation.

- **Citation Influence ratio**: we define and consider one other temporal factor which tests the long time influence of a researcher's publication, and thus indirectly represents the influence of the researcher. We assume that if a paper continues to be cited a long time after its publication, it brings higher prestige

to its author (e.g., the paper PageRank [132] is frequently and persistently cited by the following papers). To model this temporal factor, we first introduce a decay function to differentiate the weight between a pair of paper citations. If paper p_j published in year y_j cites another paper p_i published in year y_i $(y_j - y_i) \geq 0$, we define a probability as the *citation influence ratio* of paper p_j on p_i as: $CIR(p_{ji}) = \beta_1(1 - \beta_2^{y_j - y_i})$, where β_2 $(0 < \beta_2 < 1)$ is the decay base. We now define the *citation influence* between a pair of authors as: $CI(a_{ji}) = \sum CIR(p_{ji})$, where p_j is any paper of author a_j, p_i is any paper of a_i, and p_j cites p_i.

- **Contemporary h-index CH-index**[158]: this index adds an age-related weighting to each paper. The basic assumption is that the older the paper, the less the weight. The new citation count for each paper of an author can be computed as $S^c(i) = \gamma \times (Y(now) - Y(i) + 1)^{-\delta} \times |C(i)|$, where $Y(i)$ is the year when paper i is published, and $|C(i)|$ is the set of paper citing paper i. In computation, δ is often set to be 1, and γ is set to be 4. After computing this new citation count for each paper, we can compute the H-index as the standard one based on the new citation count of each paper.

- **AR-index**[80]: it is also an age-weighted index. The citation count of each paper would be divided by the age of that paper, and then the AR-index is the square root of the sum of all the papers in the *h-core* of an author.

- **AWCR-index**[65]: This is the basically the same with the AR-index, but it sums over the weighted citation count of all the papers of an author rather than only the papers in the *h-core* set.

- **AvgPubNo**: this is computed by dividing the total publication number of an author by the $CareerTime$ of this author.

- **AvgCiteNo**: this is computed by dividing the total number of citations of an author by his/her $CareerTime$.

Table 7.4: Notations

K	Number of entity types ($K = 6$)
a, p, p_c, v, w, a_f	represents author, citing paper cited paper, venue, word and feature entity type respectively.
a_i	entity of type a with index i
k	entity type. $k \in a, p, p_c, v, w, a_f$
N_k	Number of entities of type k in data corpus
D	Dimension for latent vector
V	Number of relations ($V = 6$)
θ_k	Latent matrix for entities of type k
θ_{k_t}	Latent feature vector for the t^{th} entity of type k

7.4 Joint Multi-Relational Model: Algorithm

7.4.1 Preliminary Notations

As shown in Figure 7.4, our joint multi-relational model consists of six relations generated by authors, papers, venues, words and features entities. It is noticeable to mention that we distinguish the 'paper' entity into two different entities types: the citing papers and cited papers, and therefore we altogether have six entity types. Even though to fulfill more applications, more complicated model frameworks can be generated by increasing the dimension of matrix(relation) and integrating more matrices/tensors, we focus in this current work the model design for four specific prediction tasks. Table 7.4 lists the notations.

The joint multi-relational model is a further extension and generalization of the classical matrix or tensor factorization, in which each entity in the interactions can be represented by a latent feature vector in $\mathbb{R}^D$, where D is typically a small number. By doing this, each tensor or matrix can be factorized into lower rank approximations. Figure 7.5 shows the graphical model for the data factorization associated with our model. The lower-dimensional latent vectors are denoted by $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ $(K = 6)$, where for each $k \in K$ $\theta_k = (\boldsymbol{\theta}_{k1}, \ldots, \boldsymbol{\theta}_{kN_k}) \in \mathbb{R}^{N_k \times D}$.

7.4.2 Model Factorization Maximizing MAP

Computing MAP

We choose to maximize MAP as our objective function due to its top-heavy bias property. Two questions remain for incorporating MAP into matrix/tensor factorization: how to represent the 'rank' of the entities and therefore compute the MAP scores based upon the latent feature vectors. We follow the same idea proposed in paper [157] to smoothly approximate MAP, and make it appropriate to be used for both tensors and matrices. Since our model contains one tensor and five matrices, for better illustration, we choose to take the APV-tensor and AP-matrix as two examples to show how to compute the MAP scores. The same method can be applied to the other four matrices.

In a tensor like APV-tensor, the predicted value for each entry $< a_i, p_j, v_m >$ can be computed as: $\hat{f}_{a_i p_j v_m} = \langle \theta_{a_i}, \theta_{p_j}, \theta_{v_m} \rangle = \sum_{d=1}^{D} \theta_{a_i d} \theta_{p_j d} \theta_{v_m d}$, where D is the dimension for latent vector.

Similarly, In a matrix like AP-matrix, the predicted value for each entry $< a_i, p_j >$ can be computed as: $\hat{f}_{a_i p_j} = \langle \theta_{a_i}, \theta_{p_j} \rangle = \sum_{d=1}^{D} \theta_{a_i d} \theta_{p_j d}$.

Under these schemes, suppose v_m in triple $< a_i, p_j, v_m >$ is the entity that needs to be ranked, and p_j in tuple $< a_i, p_j >$ is the entity that needs to be ranked, then we can directly approximate $1/r_{a_i p_j v_m}$ for v_m and $1/r_{a_i p_j}$ for p_j by: $\frac{1}{r_{a_i p_j v_m}} \approx g(\hat{f}_{a_i p_j v_m}) = g(\langle \theta_{a_i}, \theta_{p_j}, \theta_{v_m} \rangle)$, $\frac{1}{r_{a_i p_j}} \approx g(\hat{f}_{a_i p_j}) = g(\langle \theta_{a_i}, \theta_{p_j} \rangle)$, where function $g(\cdot)$ is the sigmoid function satisfying $g(x) = \frac{1}{1+e^{-x}}$.

Correspondingly, the loss function in terms of the MAP values for APV-tensor and AP-matrix can be computed as below:

To compute the loss function for matrix AA, PP, PW and AF, we can follow the same way as we do for the AP matrix.

Optimization

We introduced the loss function for each individual matrix/tensor in the last section. The overall loss function for this multi-relational model is therefore a summation

over all individual loss functions plus the regularization terms to prevent overfitting, as shown in Equation 7.7. We use Ω to denote the regularization terms, where $\| \cdot \|$ indicates the Frobenius norms.

We choose to use gradient ascent to solve this optimization problem. For each relation (matrix or tensor) in the model, we alternatively perform gradient ascent on the latent feature vector for one entity at each step, while keep the other latent vectors unchanged. The gradients for the same entity across different relations will be merged. The same process will be repeated for a certain number of times, or until it finally converges with no further updates on all latent feature vectors. To better illustrate, we list below the gradients for the author, paper and venue entity in the APV-tensor, and author and paper entity in the AP-matrix. Similar process can be applied into other entities in other relations. We leave the generalized updating forms for a model with K $N \times M$ matrices for future's work.

$$\begin{aligned} L_{apv} &= MAP_{apv} = \frac{1}{N_a N_p} \sum_{i=1}^{N_a} \sum_{j=1}^{N_p} \frac{1}{\sum_{t=1}^{N_v} f_{APV_{a_i p_j v_m}}} \\ &\times \sum_{t1=1}^{N_v} f_{APV_{a_i p_j v_{t1}}} g(\langle \theta_{a_i}, \theta_{p_j}, \theta_{v_{t1}} \rangle) \\ &\times \sum_{t2=1}^{N_v} f_{APV_{a_i p_j v_{k2}}} g(\langle \theta_{a_i}, (\theta_{v_{t2}} - \theta_{v_{t1}}), \theta_{p_j} \rangle) \qquad (7.5) \\ L_{ap} &= MAP_{ap} = \frac{1}{N_a} \sum_{i=1}^{N_a} \frac{1}{\sum_{j=1}^{N_p} f_{AP_{a_i p_j}}} \\ &\times \sum_{t1=1}^{N_p} f_{AP_{a_i p_{t1}}} g(\langle \theta_{a_i}, \theta_{p_{t1}} \rangle) \\ &\times \sum_{t2=1}^{N_p} f_{AP_{a_i p_{t2}}} g(\langle \theta_{a_i}, (\theta_{P_{t2}} - \theta_{P_{t1}}) \rangle) \qquad (7.6) \end{aligned}$$

$$\begin{aligned} L &= L_{APV} + L_{AA} + L_{AP} + L_{PP} + L_{PW} + L_{AF} + \Omega \\ \Omega &= \sum_{k \in a,p,p_c,v,w,a_f} \frac{\lambda_{\boldsymbol{\theta_k}}}{2} \| \boldsymbol{\theta_k} \|^2 \qquad (7.7) \end{aligned}$$

For one particular author a_i, paper p_j and venue v_m in the APV-tensor, the gradients for updating their corresponding latent vector θ_{a_i}, θ_{p_j} and θ_{v_m} can be computed as follows. For notation convenience, we adopt the following substitutions:

$$\hat{f}_{APV_{a_i p_j v_m}} = \langle \theta_{a_i}, \theta_{p_j}, \theta_{v_m} \rangle$$
$$\hat{f}_{APV_{a_i p_j (v_{t_2} - v_{t_1})}} = \langle \theta_{a_i}, \theta_{p_j}, (\theta_{v_{t_1}} - \theta_{v_{t_1}}) \rangle$$

$$\begin{aligned}
\frac{\partial L_{APV}}{\partial \theta_{a_i}} &= \sum_{s=1}^{N_p} \frac{1}{\sum_{t=1}^{N_v} f_{APV_{a_i p_s v_t}}} \sum_{t_1=1}^{N_v} f_{APV_{a_i p_s v_{t_1}}} \\
&\times [\delta_1(\theta_{p_s} \odot \theta_{v_{t_1}}) + g(\hat{f}_{APV_{a_i p_s v_{t_1}}}) \\
&\times \sum_{t_2=1}^{N_v} f_{APV_{a_i p_s v_{t_2}}} g'(\hat{f}_{APV_{a_i p_s (v_{t_2} - v_{t_1})}}) \\
&\times (\theta_{p_s} \odot \theta_{v_{t_2}})] - \lambda \theta_{a_i} \\
\frac{\partial L_{APV}}{\partial \theta_{p_j}} &= \sum_{s=1}^{N_a} \frac{1}{\sum_{t=1}^{N_v} f_{APV_{a_s p_j v_t}}} \sum_{t_1=1}^{N_v} f_{APV_{a_s p_j v_{t_1}}} \\
&\times [\delta_1(\theta_{a_s} \odot \theta_{v_{t_1}}) + g(\hat{f}_{APV_{a_s p_j v_{t_1}}}) \\
&\times \sum_{t_2=1}^{N_v} f_{APV_{a_s p_j v_{t_2}}} g'(\hat{f}_{APV_{a_s p_j (v_{t_2} - v_{t_1})}}) \\
&\times (\theta_{a_s} \odot \theta_{v_{t_2}})] - \lambda \theta_{p_j} \\
\frac{\partial L_{APV}}{\partial \theta_{v_m}} &= \sum_{s=1}^{N_a} \sum_{d=1}^{N_p} \frac{f_{APV_{a_s p_d v_m}} (\theta_{a_s} \odot \theta_{p_d})}{\sum_{t_1=1}^{N_v} f_{APV_{a_s p_d v_{t_1}}}} \\
&\times \sum_{t_2=1}^{N_v} f_{APV_{a_s p_d v_{t_2}}} [g'(\hat{f}_{APV_{a_s p_d v_m}}) \\
&\times g(\hat{f}_{APV_{a_s p_d (v_{t_2} - v_m)}}) + (g(\hat{f}_{APV_{a_s p_d v_{t_2}}}) \\
&- g(\hat{f}_{APV_{a_s p_d v_m}})) g'(\hat{f}_{APV_{a_s p_d (v_{t_2} - v_m)}})] \\
&- \lambda \theta_{v_m}
\end{aligned} \tag{7.8}$$

where

$$\begin{aligned}\delta_1 &= g'(\hat{f}_{APV_{a_ip_jv_m}}) \sum_{t_1=1}^{N_v} f_{APV_{a_ip_jv_{t_1}}} g(\hat{f}_{APV_{a_ip_j(v_{t_1}-v_m)}}) \\ &- g(\hat{f}_{APV_{a_ip_jv_m}}) \sum_{t_1=1}^{N_v} f_{APV_{a_ip_jv_{t_1}}} g'(\hat{f}_{APV_{a_ip_j(v_{t_1}-v_m)}})\end{aligned} \tag{7.9}$$

For one author a_i and paper p_j in the AP-matrix:

$$\begin{aligned}\frac{\partial L_{AP}}{\partial \theta_{a_i}} &= \frac{1}{\sum_{t=1}^{N_p} f_{AP_{a_ip_t}}} \sum_{t_1=1}^{N_p} f_{AP_{a_ip_{t_1}}} [\delta_2(\theta_{p_{t_1}}) \\ &+ g(\hat{f}_{AP_{a_ip_{t_1}}}) \sum_{t_2=1}^{N_p} f_{AP_{a_ip_{t_2}}} g'(\hat{f}_{AP_{a_i(p_{t_2}-p_{t_1})}})(\theta_{p_{t_2}})] \\ &- \lambda\theta_{a_i} \\ \frac{\partial L_{AP}}{\partial \theta_{p_j}} &= \sum_{s=1}^{N_a} \frac{f_{AP_{a_sp_j}}(\theta_{a_s})}{\sum_{t_1=1}^{N_p} f_{AP_{a_sp_{t_1}}}} \times \sum_{t_2=1}^{N_p} f_{AP_{a_sp_{t_2}}} [g'(\hat{f}_{AP_{a_sp_j}}) \\ &\times g(\hat{f}_{AP_{a_s(p_{t_2}-p_j)}}) + (g(\hat{f}_{AP_{a_sp_{t_2}}}) \\ &- g(\hat{f}_{AP_{a_sp_j}}))g'(\hat{f}_{AP_{a_s(p_{t_2}-p_j)}})] \\ &- \lambda\theta_{p_j}\end{aligned} \tag{7.10}$$

where

$$\begin{aligned}\delta_2 &= g'(\hat{f}_{AP_{a_ip_j}}) \sum_{t_1=1}^{N_p} f_{AP_{a_ip_{t_1}}} g(\hat{f}_{AP_{a_i(p_{t_1}-p_j)}}) \\ &- g(\hat{f}_{AP_{a_ip_j}}) \sum_{t_1=1}^{N_p} f_{AP_{a_ip_{t_1}}} g'(\hat{f}_{AP_{a_i(p_{t_1}-p_j)}})\end{aligned} \tag{7.11}$$

where $g'(x)$ is the derivative of $g(x)$ and $\odot$ denotes element-wise product, and λ is the regularization parameter.

7.4.3 Recommendation by Factor Matrices

After retrieving the latent matrix for each entity type, it is straightforward to generate the ranking list based upon the recommendation task and the design of matrix/tensor. Take the prediction task for the author-paper citation as one example,

given one author a_i, we can achieve the relevance score of each paper p_j in the candidate set by computing $\frac{1}{r_{a_i p_j}} \approx g(\hat{f}_{a_i p_j}) = g(\langle \theta_{a_i}, \theta_{p_j} \rangle)$, and rank all papers in descending order. The same process can be applied to all other recommendation tasks considered in our model.

7.5 Experimental Evaluation

We report in this section the experimental evaluation results for our model, and compare it with several existing state-of-the-art algorithms.

7.5.1 Data Preprocessing

We conduct our experiments on a subset of the ACM and ArnetMiner data set introduced in section 7.2.1. For papers in each data set separately, we collect the papers with complete information (authors, abstract, publishing venue and publishing year) and have been cited at least 5 times in the ACM data set and 10 times in the ArnetMiner data set. Based on these papers, we further collect all their authors and publishing venues.

We construct the tensor and matrices as introduced in section 3 for each data set. The β parameter in AA, AP and PP matrix is set to be 0.5. The PW-relation and AF-relation are constructed for all valid authors and papers. Table 7.5 shows a brief data statistics for both data sets, and the total number of records for each relation. Five-fold cross validation is conducted over the APV-relation, AA-relation, AP-relation and PP-relation to get the averaged predicting results. In the APV-relation, since each paper can have multiple authors but just one publishing venue, in order to avoid to have overlapped records in the training and testing set, we split the APV-relation into five folds by guaranteeing that one particular paper with all its authors (and the associated records) would appear in either the training or the testing set.

We further compute the average number of coauthors and cited papers for authors and papers in the AA-relation, AP-relation, and PP-relation constructed from

Table 7.5: Data Set Statistic (1)

data set	authors	papers	venues	APV records	AA records	AP records	PP records
ACM	24,764	18,121	846	47,810	112,456	366,201	71,396
ArnetMiner	49,298	47,794	1,682	132,186	361,794	1,675,564	237,531

Table 7.6: Data Set Statistics (2)

	Avg. node degree			
data set	APV	AA	AP	PP
ACM	1	10.28	17.51	4.71
ArnetMiner	1	18.40	42.03	7.81

the ACM and ArnetMiner data set separately. For simplicity, we name the average number as the 'node degree' in each relation. For example, in APV-relation, each author-paper pair is associated with one venue, therefore the average node degree in APV-relation is 1. Table 7.6 shows the simple statistic results.

We adopted MAP as our evaluation metric, as the model is specially designed for maximizing MAP. Since the data in each relation is quite sparse (as shown in Table 7.6), we cannot treat all entries with no observed data as negative samples (consider the situation that paper a should also cite paper b, but unfortunately it did not.), in which case the recommendation results would be deteriorated. To avoid this, we randomly select 200 negative samples (much higher than the average node degree in each relation) for each entity in the testing set. The performance is therefore measured based on the recommendation list that contains the known positive samples and 200 randomly selected negative samples.

In all experiments, we set the latent dimensionality $D = 10$, the regularization parameter $\lambda = 0.001$ and the learning-rate as 0.001.

7.5.2 Co-effects Analysis of Multiple Relations

In this part of experiments, we work on totally eight different kinds of multi-relational combinations, and evaluate the performance over four tasks respectively.

Table 7.7: Performance comparison over different combinations of relations (1)

Combinations	ACM			
	APV	AA	AP	PP
C0	**0.0329**	0.0487*	0.0456*	0.0389
C1	0.0263*	**0.0560**	0.0455*	0.0325*
C2	0.0282*	0.0462*	0.0458*	0.0338*
C3	0.0307*	0.0460*	0.0455*	0.0329*
C4	0.0279*	NA	NA	NA
C5	NA	**0.0560**	NA	NA
C6	NA	NA	**0.0465**	NA
C7	NA	NA	NA	**0.0395**
C8	NA	0.0468*	0.0453*	0.0325*

Table 7.8: Performance comparison over different combinations of relations (2)

Combinations	ArnetMiner			
	APV	AA	AP	PP
C0	0.0277*	0.0534*	0.0782*	0.0342*
C1	0.0289*	**0.0566**	**0.0788**	**0.0357**
C2	**0.0317**	0.0541*	0.0786	0.0353
C3	0.0285*	0.0538*	0.0784	0.0353
C4	0.0316	NA	NA	NA
C5	NA	0.0565	NA	NA
C6	NA	NA	0.0786	NA
C7	NA	NA	NA	0.0348*
C8	NA	0.0543*	0.0787	0.0349*

The eight combinations we considered include:

- c_0 indicates each single relation;
- $c_1 = \{apv, aa, ap, pp, pw, af\}$, integrating APV-relation, AA-relation, AP-relation, PP-relation, PW-relation and AF-relation;
- $c_2 = \{apv, aa, ap, pp, pw\}$, integrating APV-relation, AA-relation, AP-relation, PP-relation and PW-relation;

- $c_3 = \{apv, aa, ap, pp\}$, integrating APV-relation, AA-relation, AP-relation and PP-relation;
- $c_4 = \{apv, pw, af\}$, integrating APV-relation, PW-relation and AF-relation;
- $c_5 = \{aa, af\}$, integrating AA-relation and AF-relation;
- $c_6 = \{ap, pw, af\}$, integrating AP-relation, PW-relation and AF-relation;
- $c_7 = \{pp, pw\}$, integrating PP-relation and PW-relation;
- $c_8 = \{aa, ap, pp\}$, integrating AA-relation, AP-relation and PP-relation.

Several observations can be drawn from the results. 1) Under almost all situations, jointly modeling multiple relations can indeed improve the prediction performance. For the four tasks over two data sets (just except the publishing venue prediction (APV) on ACM data set), the best performance is always achieved when some relations are jointly modeled. 2) There is no clear trend that the more relations we jointly modeled, the better performance we can achieve. For some prediction task, i.e., the paper-paper citation prediction on ACM data set, best performance is obtained when only paper-paper-citation and paper-word relation are incorporated. However, for the ArnetMiner data set, three out of four tasks have the best performance with all relations incorporated.

For each relation in both of the two data sets, we conducted the students' t test between the best performance result with others. Statistically significant improvements (paired-based $p \leq 0.05$) are labeled with a $*$ in Table 7.7 and 7.8.

7.5.3 Comparison with Existing Methods

We report the performance comparison with three state-of-the-art approaches: the Factorization Machines (short as FM) [139], the Collaborative Topic Regression (short as CTR) [178] and the Bayesian probabilistic relational-data Analysis [194] approach.

Table 7.9: Performance Comparison: ACM data set

Approaches	ACM			
	APV	AA	AP	PP
JMRM	0.0329*	**0.0560**	0.0465*	**0.0395**
FM	**0.2127**	0.0434*	0.0388*	0.0053*
CTR		0.0374*	**0.0513**	0.0341*
BPRA	0.0161*	0.0558	0.0360*	0.0216*

Table 7.10: Performance Comparison: ArnetMiner data set

Approaches	ArnetMiner			
	APV	AA	AP	PP
JMRM	0.0317*	**0.0566**	0.0788	0.0357*
FM	**0.1595**	0.0402*	0.0613*	0.0047*
CTR		0.0395*	0.0756*	**0.0375**
BPRA	0.0176*	0.0359*	**0.0794**	0.0286*

Factorization machines are a generic approach which can effectively combine the generality of feature engineering with the high-prediction accuracy superiority of factorization models. It therefore can mimic most factorization models by simple feature engineering.

CTR model combines traditional collaborative filtering with topic modeling. BPRA jointly models coupled matrices and tensors but optimizes the model by minimizing RMSE.

For FM, CTR and BPRA models, we feed the same training and testing set we used for JMRM, and evaluate the prediction performance on each individual relations separately. For JMRM, the reported results are the best results selected from different combinations of multiple relations (as shown in Tables 7.7 and 7.8). For using FM method, we regard the tasks as 'regression' tasks; The dimensionality of the factorization machine is set to be '1,1,8', indicating that the global bias, one-way interactions and pairwise interactions are all used, and that the number of factors used for pairwise interactions is set to be 8. Stochastic gradient descent (SGD) is chosen to used as the learning method. For CTR method, we construct

paper profiles by their abstracts, and author profiles by concatenating all their publications. The basic LDA is used to retrieve the topic proportion and distribution vectors. The dimension for latent factor is set to be 10, and the number of latent topics is set to 20. Since CTR is only proposed for factorizing two types of entities, we did not adopt it to the task of publishing venue prediction (the APV-relation). Note that both FM and CTR are implemented using publicly available software. We also set the dimension for latent factor in BPRA as 10.

Table 7.9 and 7.10 show the results. As indicated, we found that our JMRM mode can outperform FM and CTR in several cases which demonstrates the effectiveness of our model. FM can achieve significantly better results than JMRM in predicting publishing venue, but has a very poor performance in predicting paper-paper citation. Our model shows the best overall performance, since out of 8 cases (four recommendation tasks over two data sets), our model ranks first for three cases, and the second for the other five cases, demonstrating its superiority in providing recommendations for four tasks simultaneously.

7.6 Bibiliographic Notes

In this section, we first review three lines of recent development of the latent factor based collaborative filtering (CF) models that are relevant to our research in this work, and then introduce some related research on each specific recommendation task we considered in this work. The three lines of research are: latent factor models 1) with additional features or contents integration 2) for multi-relational higher-order matrices factorization and 3) for ranking-based optimizations.

Recently, researchers have explored to enhance the traditional latent factor models by incorporating additional features or content of participating entities. One group of work in this direction is the 'Regression Based Factor Models', proposed by Agarwal and Chen [4], whose basic idea is to replace the zero-mean Gaussian distributions with regression-based means. Another work is the CTR model [178], which combines matrix factorization with probabilistic topic modeling for scientific

papers recommendation. The third work is the 'feature-based matrix factorization' [32], which combines the traditional latent factor model with linear regression. However, all of these three models can only cope with the two-order data interactions, and cannot be model higher-order data structures. The forth work is the 'Factorization Machine' model proposed by Rendle [139], which combines latent factorization model with SVM. Compared with these work, we incorporate both features for papers and authors in our model. The model is designed for more than two-order data interactions, and is based on pair-wise learning mechanism.

The second direction of development for latent factor model emphasizes on joint modeling multi-relational relations. The 'collective matrix factorization' from Singh and Gordon [161] is one typical work in this direction. However, the 'multi-relation' shown in this work is only limited to be two or three relations. Most recently, Yin et al. [194] proposed a 'Bayesian probabilistic relational-data Analysis' (BPRA) model which extends the BPMF and BPTF model by making it applicable to arbitrary order of coupled multi-relational data structures. However, the model is also used for personalized tag recommendation, which is a different research domain with our paper, and is based upon point-wise RMSE optimization, different from our targeted ranking-based optimization.

Even though most of the traditional latent factor models target at optimizing point-wise measures, such as RMSE or MSE, several ranking-based optimization models have been proposed. One relevant work is the 'Bayesian Personalized Ranking' (BPR) model [141], which minimizes the AUC metric by using a smooth version of the hinge loss.The method that is most similar to our work is the TFMAP model [157], which proposes a method to approximate and optimize the MAP measure. However, their model is for user-item-context recommendation, and is only able to deal with one single tensor relation, which are both different from our work in this chapter.

We then summarize some relevant work with each specific recommendation task considered in this chapter. Future paper citation recommendation is the most widely explored problem. We categorized existing works into three groups. In the first

group, neighborhood based CF models along with graph-based link prediction approaches are widely used to tackle the citation recommendations for a given author or paper with a partial list of initial citations provided, typical works in this category include [117], [207], [167] and etc. In the second group of approach, probabilistic topic modeling is used for citation list generation.In the third group, citation context (the text around citation mentions) is utilized. Typical work includes the context-award citation recommendation work and its extensions proposed by He et al. [69, 68] Despite of these existing work, few work has be developed using CF latent factor models for recommendation, excluding the CTR model.

Coauthor-ship recommendation is mostly tackled by using graph-based link prediction approach. The most representative work is proposed by Liben-Nowell [101], which measures the performance on using several graph-based metrics. The work on predicting future conference(venue) submission is seldom explored. Lau and Cohen [99] develop a combined path-constraint random walk-based approach, not only for venue recommendation, but also for citation recommendation, gene recommendation and expert finding. Pham et al. [135, 136] define the task of venue recommendation as predicting the participating venues of users, and therefore their input is users rather than papers.

7.7 Summary

In this chapter, we proposed a joint multi-relational model to recommend author-author coauthorships, author-paper citations, paper-paper citations and paper publishing venues. The model is proposed based on the assumption that these activities are coupled, and that joint modeling can help us in achieving more coherent and accurate latent feature vectors. Moreover, we extend an existing work maximizing MAP over one single tensor into a more generalized form which is able to maximize MAP over several matrices and tensors. Experiments carried out over two real world data sets demonstrate the effectiveness of our model.

Chapter 8

Conclusions and Future Work

In this chapter, we conclude this dissertation. We fist summarize the main contributions we have made in this dissertation, and then analyze its potential impact and applications in other research directions; we further discuss the limitations and deficiencies of the current research, and finally discuss possible future work directions.

8.1 Recapitulation

In this dissertation, we focus on applying information retrieval, data mining and machine learning techniques into mining and analyzing academic network, which to our definition, is a certain kind of social network that concentrates in the academic domain. The nodes in an academic work are scientific-related entities, such as authors, papers, venues, and the links in the network model the relationships between these scientific entities, including the co-authorships, citations, and etc. Two specific research problems: the expertise retrieval problem and research action prediction and recommendation problem are particularly addressed.

Academic network has its own characteristics. It consists of heterogeneous data; it can be divided into multiple levels of communities; and it is often dynamic. These characters make the research on academic network interesting and challenging. In this dissertation, we mainly focus on the property of heterogeneity, where several algorithms and models have been proposed to integrate different sources of data and

relationships, and their effects have been demonstrated in both expertise retrieval and research action recommendation tasks. Temporal factor is another factor that we have considered and made endeavors to incorporate it into our proposed models. The specific contributions of our dissertation are introduced as follows.

For the task of expertise retrieval, we made several contributions:

Firstly, we generated a unified heterogeneous framework that consists of four types of academic entities: authors, papers, publishing venues and authors' affiliations to evaluate the expertise of authors. To our best knowledge, this is the first work that specifically combining both the 'venues' and 'affiliations' into an academic network, and therefore provides a more complete and general aspects of view of the academic environment, and can evaluate the expertise of a researcher more comprehensively. Based on this unified framework, we first test the performance for expert finding on different versions of this framework by either deleting a certain type of entities or relationships, and experiments demonstrate the effectiveness of integrating more complete data entities. We further proposed/applied three modified PageRank-like algorithms on this network to estimate and rank researchers' expertise. In the first algorithm, we introduced the topical PageRank into academic network analysis, and therefore we can identify experts on the topic level; we then proposed a heterogeneous PageRank algorithm, which investigates the different contributions of the participating entities in determining the expertise of a researcher; we finally distinguished some temporal-related features, and proposed a temporal-based PageRank into a particular expert finding work on SIG-community award predictions. We compared our proposed algorithms based on the ACM digital library data set with several state-of-the-art approaches, and demonstrated their superiority.

Secondly, we proposed an enhanced author-topic model (the ACTV model) by directly modeling two additional information: the conference venues and cited authors information into the topic modeling process. This extends the previous author-topic-model based approaches when fewer valuable information is incorporated. Experiments based on two real world data sets: the ACM digital library data set

and ArnetMinet data set with two sets of different queries and ground truth labels demonstrate the effectiveness of our proposed model as it can outperform the previous state-of-the-art approaches.

Thirdly, we proposed a model that formally incorporates the pair-wise based learning-to-rank algorithm into topic modeling process. This is a fundamental direction in expert finding where the probabilistic discriminative model (the learning-to-rank approach) can be effectively combined with the generative probabilistic model (the topic modeling approach). Even though previous works have been conducted using either the discriminative or generative models, the combination of them is seldom explored before. We took this step and demonstrated the model's effectiveness via experiments on both the ACM and ArnetMiner data sets.

For the task of research action prediction, we have made the several contributions:

We took the first step in investigating whether publishing venues can be classified and predicted by leveraging linguistic stylometric features. Since there are many available conferences, it is sometimes difficult to decide which to submit. One of the main contribution we made is that we identified several stylometric features, and we compared and showed the improved classification performance when combing both the content-based and stylometric features. We then proposed a modified collaborative filtering approach for venue recommendation, in which two extensions were made and verified: the extension on incorporating the stylometric features into computing the similarity between papers, and the extension on distinguishing the different weight of contributions of the neighboring papers via parameter tuning and optimizing.

We then tested and demonstrated the capability of our proposed ACTV model in both cited-author prediction and publishing venue prediction, and shown improved performance over other existing topic modeling based work in these two tasks.

We proposed an extended latent factor model that can jointly model several relations in an academic environment and evaluated its performance in four recommendation tasks: the recommendation on author-coauthorship, author-paper citation, paper-paper citation and paper-venue submission. The model is proposed based

upon the assumption that several academic activities are highly coupled, and that by joint modeling, we can not only solve the cold start problem but also help in achieving more coherent and accurate latent feature vectors. Moreover, to facilitate ranking, we extend an existing work which directly maximizes MAP over one single tensor into a more generalized form and is therefore able to maximize MAP over several matrices and tensors. Experiments carried out over two real world data sets demonstrate the effectiveness of our model.

8.2 Impact

We focus on mining and analyzing the academic network in this dissertation, which is a subset and special case of a much larger, complicated and varied social network in the social media domain, connecting millions of common users (not limited to researchers in the academic domain) and different kinds of social entities (such as movies, tags, videos, photos, social comments, products, etc). Even though the models and algorithms we developed in this dissertation are especially designed for the academic network, the ideas behind those models can be extended beyond academic network research and inspire the research in other domains, due to the following two reasons: 1) the academic network and other social networks share common properties; 2) there exist similar information needs and applications in other social medias. In this section, we will first discuss the similarity between the academic network and other social networks in terms of both network property and similar applications, and then discuss the possible application of each of our proposed models int other domains.

Similarity in network properties

Social media has provided us abundant services nowadays, including the social tagging or information sharing systems (e.g., Youtube, Flickr [53], Bibsonomy, and Delicious), microblogging systems (e.g., Twitter, Weibo), social communication networks (e.g., Facebook, Renren), professional networks (e.g. Linkedin), information filtering and recommender systems (e.g. Netflix, MovieLens, Amazon product

reviews), news search and online computational advertising (e.g., Bing sponsored search). Despite different kinds of services they provided, these applications can all be represented and modeled as social networks, in the same way as we model the academic network, where system participators like users, products, and comments can be represented as nodes and their mutual relationships like following/follower (in Twitter), being friends (in Facebook), rating a movie (in Netflix) can be represented as links. These constructed networks also show the same properties as we analyzed for the academic networks. First of all, they are often heterogeneous networks, consisting of different types of entities and relations. In Netflix system, we have users and movies being connected by ratings; in Twitter, we have users who can follow each other, and tweets which can be re-tweeted by users; in social tagging systems, we have even more entities and relationships, for example, in Flickr, we have users, photos(items), tags, and comments; users can be friends with each other; can tag an item as well as comment an item. Secondly, entities in those social networks can form communities. Linkedin offers a good example, as it allows users to select different groups or communities to join in. Thirdly, those networks are also dynamically changing. In Facebook, users often update their status, locations, and generate new friendship with other users; in Linkedin, users often update their status by changing affiliations, job titles or getting connections with new friends; in Twitter, the tweet which is most frequently re-tweeted varies over time, indicating the evolution of hot topics either globally or locally over time. Due to the similar network properties we considered when developing algorithms, the models we proposed are applicable to tackle those similar problems in other social networks.

Similarity in applications

We emphasize on mining and analyzing the academic network for two specific tasks in this dissertation: the ranking of research experts and recommendations for academic actions, both of which can find similar applications in web search and/or other social media applications. The task of ranking experts in terms of their estimated expertise to a query (in a domain) is essentially equivalent to ranking web pages according to their relevance to a query. In social media domain,

we have various such information needs and applications. In the question-answer (Q&A) systems, we often need to identify the best answers (ranking answers) to a given question, and/or to rank users who can provide the most best answers. In Blog or Twitter, people are also interested in finding blogs/tweets which are most popular or most instructive on certain topics, and those bloggers or Twitter users who are believed to be the most prestigious in raising or discussing about a certain topic. Developing recommender systems has an even wider range of applications in social media domain, for example, recommending tags or comments for user-item pairs (Flickr, Delicious), predicting ratings of a movie to users (Netflix system), or recommending ads which can attract the highest clicks from users (online advertising systems). The ideas behind our design for generating recommendations for co-authors, citations and publishing venues in the academic domain can also be adapted in other domains.

Besides the generalized analysis on the similarity between the academic network and the other social networks, we then discuss some specific impact of each of our models on other research domains.

Topic-driven multi-type citation network analysis for ranking authors: in this research work, we developed a multi-type heterogeneous citation network connecting four types of entities authors, papers, affiliations and venues to ranking authors. The following properties distinguish the model from other related: 1) a multi-type heterogeneous network; 2) a Page-Rank basic ranking function with modification; 3) combining both content-based expertise with graph-based ones. 4) borrowing the Topical PageRank algorithm into citation network analysis to differentiate the topic-based difference in expertise propagation; 5) differentiating the importance of different types of entities in propagation; and 6) incorporating temporal factors to differentiate the expertise importance in propagation. Such a model design with its specific features can also be applied to other systems ranking entities.

- In Blog search or Twitter search, where we intend to rank blogs/tweets or

bloggers/twitters, we can construct a bipartite graph connecting tweets(blogs)-tweets(blogs), twitters(bloggers)-twitters(bloggers), and twitters(bloggers)-tweets(blogs) rather than individual homogeneous graphs. In spite of its simple scheme, PageRank-like ranking functions are still verified to be effective and efficient ranking functions, and we can also modify it by combining with the blogs/tweets content information. Moreover, twitters/bloggers' interests over topics may evolve over time, and therefore changing their weight of importance over topics by either crediting or discrediting the importance of old tweets/blogs may lead to improvement of the ranking results.

- In Q&A systems, where the best answers and/or the most knowledgeable persons on certain topics are to be identified, we can construct a tripartite graph connecting questions, answers and answer providers (users). Temporal factors can also be incorporated to represent the expertise of an answer provider, such as how long the person has been an active user, and how frequently and how quickly he answers the problems.

- The basic idea of ranking authors by leveraging the information from integrated information sources can also be applied into link prediction or recommendation task. In the social tagging systems, for example, users, tags and items can be connected to form a multi-type network, based upon which, the most related tags for user-item pairs can be identified. users, tags, and items specific features, including temporal features can be incorporated to determine the propagation weight. Decayed importance can be applied for users not using a specific tag for a long time.

A joint topic modeling approach for academic network analysis: in this research work, we extended the previous author-topic models by incorporating citation and venue information for three tasks: ranking authors, cited author predictions and venue predictions. The fundamental idea of integrating additional factors is essentially the sames as the our topic-driven multi-type citation analysis work, however, we set it in the topic modeling framework, which has at least the following two

advantages: 1) topic models can better discover the latent meanings of words than bag-of-words approach, which is especially important when documents have fewer words. 2) we naturally combine content and link information in the generating process rather than linearly combine them after each ranking is achieved.

- In microblog search, such as Twitter search when influential twitter users are to be identified, topic modeling can help to achieve users' expertise distributions over topics as represented in his posted tweets. Since tweets are normally short with at most 140 words, topic modeling based approach can help to better understand the content of tweets than bag-of-words based approach.

Ranking authors by learning-to-rank topic modeling: in this work, we integrated pairwise learning-to-rank into topic modeling for ranking authors. The prominent advantage of introducing the learning-to-rank mechanism is that we can easily incorporate features of ranking entities in addition to textual features derived from topic modeling process. Other ranking-oriented research tasks can also get benefit from such an integration.

- In Twitter search for influential twitter users, topic models can help to achieve users' expertise distributions over topics as represented in their posted tweets; Other user-specific features, like user's status, geographical locations, number of followers/followees, number of tweets can be incorporated by the learning-to-rank scheme. It would also be helful in finding most popular tweets over topics as there are additional metadata on tweets, such as hashtags and thematic labels provided by users. All these metadata can be well incorporated by the learning-to-rank scheme.

- Similar mechanism can be applied into Q&A systems, where user-related and answer-related features can be explicitly represented and incorporated into the learning and ranking process.

Venue classification and prediction: Given a paper to determine its potential publishing venue is equivalent to the task of given a user in Facebook to determine

which group he/she can join in. To do that, making use of the friends information of that user (which is equivalent to making use of the neighboring papers of the target paper) will help to achieve satisfying results. Identifying and integrating stylometric features in both classification and prediction is one distinguished property of our model, which to our belief can shed light in microblog Twitter search on finding global or local influential topics, as geographically different people tend to have different speaking/writing styles.

Joint multi-relational model for recommendations: Two properties distinguished our joint multi-relational model from other recommender systems: 1) we integrate and jointly model several coupled relations represented as either tensors or matrices in order to achieve more coherent and accurate latent factors among entities; 2) choose to optimize the ranking-based metric MAP in order to favor the top N ranking results. Both these two properties can benefit other related recommendation tasks.

- In social tagging systems, such as Flickr and MovieLens, there exist multiple types of entities generating coupled relations, such as user-tag-item, user-comment-item, user-user-friendship. There has exist research work optimizing RMSE over such joint coupled relations, or optimizing MAP over one single tensor. However, combining these two mechanisms as our model proposed has not been applied in recommendations in social tagging systems.

- Similar mechanism can be applied into social communication systems (such as Facebook) where friendship is to be recommended, microblogging systems (such as Twitter) where new following/follower relation is to be predicted, or online advertising systems where users' browse and click behavior are to be predicted. In all these systems, we have multi-type of entities enriched with features generating coupled relations, and we normally appreciate the top ranked results, for example, in online advertising systems, the top four ads are considered.

8.3 Caveats

We recapitulated the main contribution of this dissertation; presented the impact of our work; we now analyze the limitations and deficiencies of the dissertation projects respectively.

Topic-driven multi-type citation network analysis for ranking authors Several limitations remain for this work.

- We made use of the hierarchically-organized ACM categories to retrieve topic distributions of entities (authors, papers, queries, etc). Some other more widely used topic modeling approach, such as pLSA and LDA can be adopted.
- Experiments are conducted on one data set, the ACM data set. Additional experiments on other data sets may better validate the effectiveness of our model.

A joint topic modeling approach for academic network analysis

- One of the limitations in our model design is that we assume that each word in the author profiles will be associated with a cited author. However, in real situations, only those words in the introduction section or related work section are likely to be related with cited authors. Therefore, a better model design would be firstly identify word portions that are cited-author related, and only model those words in the joint modeling process for topic, cited authors and venues, while other words only contribute to the topic and venue generation process.

Ranking authors by learning-to-rank with topic modeling

- One of the limitations in this model is that we create a virtual profile for each author by concatenating all his/her publications. This process may introduce much noise, as different papers of an author may cover different topics. A solution to this problem is to develop a two-layer topic models, in which the lower level models paper content, and the upper layer models authors' interests.

Venue classification and recommendation

- More stylometrics may need to be identified and used in addition of the currently used ones, for example, the POS tags.
- A larger size of testing set (more than 10000 randomly chosen papers) could be constructed to better validate the model performance.

A joint multi-relational model for recommendations in academia

- Computational efficiency is the biggest problem of this model. It normally takes over a week for 50 iterations, which would make the model inappropriate for online recommendation. A more efficient algorithm need to be developed for MAP computing and entities' latent vector updating.

8.4 Future Work

Even though a number of achievements in both expertise retrieval and research action prediction have been presented in this dissertation, there are several open issues that need to be explored in future work. We discuss them for the two tasks separately.

8.4.1 Expertise Retrieval

Temporal evolution

In this dissertation, we identified several temporal factors, and incorporated them into the temporal PageRank algorithm or took them as individual features to feed into the learning-to-rank topic modeling process. However, this seems far from enough. More well-formed machine learning techniques, such as time series analysis techniques may be utilized to better model the temporal evolution of experts' expertise and further improve the expert finding ranking performance. Prior research

on temporal analysis for traditional web search can be valuable in addressing the problem in expertise retrieval.

Expertise retrieval on Web and social media

Expertise retrieval has been traditionally studied on enterprise intranet or limited to a specific domain, for example, as emphasized in this dissertation, in the academic domain. However, it would be a more interesting and challenging work to find experts on Web where more plenty of information are available with varying quality, and in social media which provides a modern platform for more and more people indicating and sharing their expertise. Even though some work has been proposed for expert finding on question-answering sites or on Twitter, more research efforts can be made in this direction as finding experts in social media is very challenging. First of all, there would be a much wider variety of expertise areas compared to those identified areas in enterprise intranet or academic domain; Secondly, there are a huge number of users online, which would make the scalability and efficiency problem a key research problem. Thirdly, expertise identified via social media would be highly dependent on time and location which indicates that more research work will be emphasized on temporal or geographical analysis. Privacy and security issue will also play a role in finding experts in social media.

Besides conducting research on identifying experts on social media alone, it would be an interesting task to combine and integrate those expertise represented in enterprise intranet, academic domain, and social media domain to leverage the advantages from all of them.

Personalized expertise retrieval

The current expert finding task normally generates one global ranking results for all users with the same query. However, people may tend to have their understanding or interpretation on what is expertise and who can be regarded as experts, specially in social media domain where various query topics exist. Therefore, generating personalized expert ranking results would be a challenging task. Collecting user interactions with the expert ranking system via their query logs, click-through data,

explicit or implicit feed-backs will help to address this problem.

Community-based expertise retrieval and fuzzy query match

Community is one prominent property of academic network and other social networks. Finding community-based experts would be more accurate in some scenarios than query-based expert finding since sometimes it is difficult to use a query consisting of several terms to describe a community. For example, if we present each author by his publishing paper titles, and suppose we intend to find experts on 'information retrieval', if only using content-based algorithms, the ranking performance would not be good enough, since few authors will explicitly indicate 'information' and 'retrieval' in their paper titles or even abstracts. However, these researchers on information retrieval would form a community by other kind of interaction. The community identified in the work of H. Deng [42] is based upon publishing in the same conferences but not automatically generated. Moreover, enhanced methodologies can be provided which allow us to estimate the relevance of experts by their close meaning but not term exact match to the given query. Natural language processing and machine translation techniques may help in solving this problem.

Expertise retrieval: go beyond just relevancy

Expertise retrieval has been widely researched to retrieve and rank experts based on their estimated expertise in terms of their relevance to a given query. However, there are other interesting aspects of people's expertise, for example, their diversity (doing research covering different domains), their sociability (active academic activities organizers), or their potential capability (research rising stars). Identifying experts from multiple facets would provide a more comprehensive view of experts.

8.4.2 Research Action Prediction and Recommendation

Temporal-sensitive recommendation

In this dissertation, we aim at generating accurate recommendations or predictions while ignoring the temporal requirements. For example, to recommend the

most recently published paper citations, or to recommend future possible collaborators. In our work on joint multi-relational model, we constructed the matrices/tensors by considering the temporal factor, however, we did not evaluate the performance for recommending temporal-sensitive actions. This will constitute one part of our future work.

Personalized recommendation and prediction

Generating personalized recommendation is one of the key requirements in modern recommender systems, and there also exist such information needs for the recommendation tasks in the academic environments. For example, some authors prefer to cite papers with higher relevancy, but others prefer to cite those more recently published. In our current proposed models, we do not explicitly consider this factor. In the joint multi-relational model, for example, no user or paper specific bias has been incorporated. Future work can be conducted to overcome this deficiency.

Bibliography

[1] Friends and neighbors on the web. *Social Networks*, 25(3):211–230, July 2003.

[2] Finding and evaluating community structure in networks. *Physical Review E*, 69, 2004.

[3] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommendation systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans on Knowledge and Data Engineering*, 17(6):734–749, June 2005.

[4] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *KDD*, 2009.

[5] R. Albright, J. Cox, D. Duling, A. N. Langville, and C. D. Meyer. Algorithms, initializations, convergence for the nmf.

[6] S. Argamon, M. Saric, and S. Stein. Style Mining of Electronic Messages for Multi Authorship Distrimintation: First Results. In *KDD*, 2003.

[7] R. Arun, R. Saradha, V. Suresh, and M. Murty. Stopwords and Stylometry: A Latent Dirichlet Allocation Approach. In *NIPS workshop on Application for Topic Models: Text and Beyond*, 2009.

[8] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 43–50, New York, NY, USA, 2006. ACM.

[9] K. Balog, T. Bogers, L. Azzopardi, M. Rijke, and A. Bosch. Broad Expertise Retrieval in Sparse Data Environments. In *30th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, 2007.

[10] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and S. Luo. Expertise retrieval. *Found. Trends Inf. Retr.*, 6:127–256, 2012.

[11] K. Balog and M. Rijke. Finding Experts and their Details in E-mail Corpora. In *15th International World Wide Web Conference (WWW 2006)*, 2006.

[12] K. Balog and M. Rijke. Determining Expert Profiles (With an Application to Expert Finding). In *In: Proceedings IJCAI-2007*, 2007.

[13] P. Batista, M. Campiteli, and O. Kinouchi. Is it possible to compare researchers with different scientific interests? *Scientometrics*, 68:179–189, 2006.

[14] W. Berger, Y. Tanya, and J. Saia. A framework for analysis of dynamic social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 523–528, 2006.

[15] H. Biswas and M. Hason. Using publications and domain knowledge to build research profiles: an application in automatic reviewer assignment. In *International conference on Information and Communication Theory*, 2007.

[16] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, New York, NY, USA, 2006. ACM.

[17] D. M. Blei and J. D. McAuliffe. Supervised topic models. In *NIPS*, 2007.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.

[19] Bogers and Toine. Using citation analysis for finding experts in workgroups.

[20] L. Bottou. Large-Scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics*, pages 177–187, Aug. 2010.

[21] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of Predictive Algorithms for Collaborative Filtering. In *UAI*, pages 43–52, 1998.

[22] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[23] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 89–96, 2005.

[24] J. Burrows. 'An Ocean where each kind...': Statistical analysis and some major determinants of literay style. *Computers and the Humanities*, 23(4-5):309–321, 1989.

[25] C. Campbell, P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, pages 528–531, 2003.

[26] Y. Cao, J. Liu, S. Bao, and H. Li. Research on expert search at enterprise track of TREC 2005. In *In TREC-2005: Proceedings of the 14th Text Retrieval Conference*, 2005.

[27] S. Chan, P. Hui, and K. Xu. Community detection of Time-Varying mobile social networks. In *Complex Sciences*, volume 4 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, chapter 115, pages 1154–1159. Springer Berlin Heidelberg, 2009.

[28] J. Chang and D. M. Blei. Hierarchical relational models for document networks. *Annals of Applied Statistics*, Oct. 2010.

[29] C. Chaski. Empirical evaluations of language-based author identification techniques. In *Forensic Linguistics*, 2001.

[30] H. Chen, H. Shen, J. Xiong, and S. T. X. Cheng. Social Network Structure behind the Mailing Lists: ICT-IIIS at TREC 2006 Expert Finding Track. In *The Fourteenth Text REtrieval Conference Proceedings (TREC 2006)*, 2006.

[31] P. Chen, H. Xie, S. Maslov, and S. Redner. Finding scientific gems with google's PageRank algorithm. *Journal of Informetrics*, 1(1):8–15, Jan. 2007.

[32] T. Chen, Z. Zheng, Q. Lu, W. Zhang, and Y. Yu. Feature-based matrix factorization. *CoRR*, abs/1109.2271, 2011.

[33] X. Chen, X. Hu, Z. Zhou, C. Lu, G. Rosen, T. He, and E. K. Park. A probabilistic topic-connection model for automatic image annotation. In *CIKM*, pages 899–908, 2010.

[34] N. Craswell, A. de Vries, and I. Soboroff. Overview of the trec 2005 enterprise track. In *TREC*, 2005.

[35] N. Craswell, D. Hawking, A. Vercoustre, and P. Wilkins. Broad Expertise Retrieval in Sparse Data Environments. In *30th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, 2007.

[36] B. D. Davison. Toward a unification of text and link analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 367–368, New York, NY, USA, 2003. ACM.

[37] O. de Vel. Mining Email authorship. In *Text Mining Workshop. KDD*, 2000.

[38] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[39] H. Deng, J. Han, M. R. Lyu, and I. King. Modeling and exploiting heterogeneous bibliographic networks for expertise ranking. In *JCDL*, pages 71–80, 2012.

[40] H. Deng, I. King, and M. R. Lyu. Formal models for expert finding on dblp bibliography data. In *ICDM*, pages 163–172, 2008.

[41] H. Deng, I. King, and M. R. Lyu. Enhancing expertise retrieval using community-aware strategies. In *CIKM*, pages 1733–1736, 2009.

[42] H. Deng, M. R. Lyu, and I. King. Effective latent space graph-based re-ranking model with global consistency. In *WSDM*, pages 212–221, 2009.

[43] B. Dom, A. Eiron, A. Cozzi, and Y. Zhang. Graph-based ranking algorithms for e-mail expertise analysis. In *In 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003.

[44] D. Duan, Y. Li, R. Li, R. Zhang, and A. Wen. Ranktopic: Ranking based topic modeling. In *ICDM*, pages 211 –220, dec. 2012.

[45] S. Dumais and S. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *In Proceedings of ACM SIGIR 1992*, pages 233–244, 1992.

[46] E. EErosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. In *Proc. of the National Academy Sciences*, pages 5220–5227, 2004.

[47] Egghe. Theory and practice of the g-index. *Scientometrics*, 69:131–152, 2006.

[48] Fackbook. Facebook homepage. `https://www.facebook.com`.

[49] H. Fang and C. Zhai. Probabilistic models for expert finding. In *Proceedings of the 29th European conference on IR research*, ECIR'07, pages 418–430, Berlin, Heidelberg, 2007. Springer-Verlag.

[50] Y. Fang, L. Si, and A. Mathur. Discriminative Models of Integrating Document Evidence and Document-Candidate Associations for Expert Search. In *SIGIR*, 2010.

[51] J. Farringdon, A. Morton, M. Farringdon, and M. D. Baker. *Analysis for Authorship: A Guide to the Cusum Technique*. University of Wales Press, 1996.

[52] O. Feiguina and G. Hirst. Authorship attribution for small texts: Literary and forensic experiments. In *Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*, 2007.

[53] Flickr. Flickr homepage. `https://www.flickr.com`.

[54] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, Jan. 2010.

[55] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, Dec. 2003.

[56] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[57] Y. Fu, R. Xiang, Y. Liu, M. Zhang, and S. Ma. Finding experts using social network analysis. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 77–80, 2007.

[58] E. Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178(60):471–479, Nov. 1972.

[59] E. Garfield and R. Merton. Citation indexing-its theory and application in science, technology and humanities. *garfield.library.upenn.edu*, Jan. 1979.

[60] P. Glenisson, W. Glanzel, F. Janssens, and B. D. Moor. Combining full text and bibliometric information in mapping scientific disciplines. *Inf. Process. Manage.*, 41(6):1548–1572, Dec. 2005.

[61] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, Dec. 1992.

[62] S. D. Gollapalli, P. Mitra, and C. L. Giles. Ranking authors in digital libraries. In *JCDL*, pages 251–254. ACM, 2011.

[63] T. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, pages 5228–5235, 2004.

[64] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 540–547, New York, NY, USA, 2009. ACM.

[65] A. Harzing. *The publish or perish book.* 2010.

[66] M. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[67] T. Haveliwala. Topic-sensitive pagerank. In *WWW*, pages 517–526, 2002.

[68] Q. He, D. Kifer, J. Pei, P. Mitra, and C. L. Giles. Citation recommendation without author supervision. In *WSDM*, pages 755–764, 2011.

[69] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *WWW*, pages 421–430, 2010.

[70] J. Hirsch. An index to quantify an individual's scientific research output. *Proc.Nat.Acad.Sci.*, 46:16569, 2005.

[71] J. E. Hirsch. Citation indexing: Its theory and application in science, technology, and humanities. In *Proceedings of National Academy of Sciences*. John Wiley and Sons, Inc., 2005.

[72] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[73] T. Hofmann and J. Puzicha. Latent Class Models for Collaborative Filtering. In *AI*, pages 688–693, 1999.

[74] A. Hogan and A. Harth. The expertfinder corpus 2007 for the benchmarking and development of expert-finding systems. In *In proceedings of the International ExpertFinder Workshop (EFW)*, 2007.

[75] D. Holmes and R. Forsyth. The Federalist revisited: New Directions in author attribution. *Literary and Linguistic Computing*, 10(2):111–127, 1995.

[76] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. FolkRank: A Ranking Algorithm for Folksonomies. In *In Proc. of LWA*, pages 111–114, 2006.

[77] M. Huhns, U. Mukhopadhyay, L. Stephens, and R. Bonnel. DAI for document retrieval: The MINDS project. In *In Huhns, M.N.,ed., Distributed Artificial Intelligence*, pages 249–283, 1987.

[78] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 538–543, 2002.

[79] X. Jiang, X. Sun, and H. Zhuge. Towards an effective and unbiased ranking of scientific literature through mutual reinforcement. In *CIKM*, pages 714–723, 2012.

[80] B. Jin. The ar-index: Complementing the h-index. *Intl. Socienty for Scientometrics and Informetrics Newsletter*, 2007.

[81] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM.

[82] T. Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 217–226, 2006.

[83] N. Johri, D. Roth, and Y. Tu. Experts' retrieval with multiword-enhanced author topic model. In *NAACL*, 2010.

[84] A. Kanfer, J. Sweet, and A. Schlosser. Humanizing the net: Social navigation with a "know-who" email agent. In *In proceedings of the 3rd Conference on Human Factors and the Web*, 1997.

[85] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*, RecSys '10, 2010.

[86] J. Karlgren. The wheres and whyfores for studying text genre computationally. In *Workshop on Style and Meaning in Larguange, Art, Music and Design. National Conference on Artificial Intelligence*, 2004.

[87] H. Kashima and N. Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 340–349, 2006.

[88] S. Kataria, P. Mitra, C. Caragea, and C. Giles. Context Sensitive Topic Models for Author Influence in Document Networks. In *IJCAI*, 2011.

[89] L. Katz. A new status index derived from sociometric analysis. *PSYCHOMETRIKA*, 18(1):39–43, 1953.

[90] H. Kautz, B. Selman, and M. Shah. Referral Web: Combining Social Networks and Collaborative Filtering. . In *Comm. ACM, vol. 30 no. 3*, 1997.

[91] H. Kautz, H. Selman, and M. Shah. The Hidden Web. In *The AI Magazine, vol. 18, no. 2*, pages 27–36, 1997.

[92] V. Keselj, F. Peng, N. Cercone, and C. Thomas. N-gram-based author profiles for authorship attribution. In *In Proc. of the conference pacific association for computational linguistics*, pages 255–264, 2003.

[93] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Journal of the ACM, 46(5)*, pages 604–632, 1999.

[94] A. Kohrs and B. Merialdo. Clustering for Collaborative Filtering Applications. In *Computational Intelligence for Modelling, Control and Automation*. IOS, 1999.

[95] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM REVIEW*, 51(3):455–500, 2009.

[96] M. Kolla and O. Vechtomova. Broad Expertise Retrieval in Sparse Data Environments. In *SIGIR*, 2007.

[97] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*.

[98] B. Krulwich and C. Burkey. The ContactFinder: Answering bulletin board questions with referrals. In *In proceedings of the National Conference on Artitificial Intelligence*, pages 10–15, 1996.

[99] N. Lao and W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learn*, 81:53–67, 2010.

[100] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, pages 556–559, 2003.

[101] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7), 2007.

[102] D. Lichtnow, S. Loh, L. Riberio, and G. Piltcher. Using Text Mining on Curricula Vitae for Building Yellow Pages. 2006.

[103] Linkedin. Linkedin homepage. `https://www.linkedin.com`.

[104] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331.

[105] X. Liu, J. Bollen, M. Nelson, L. Michael, and H. V. de Sompel. Co-authorship networks in the digital library research community. *Inf. Process. Manage.*, 41(6):1462–1480, Dec. 2005.

[106] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-Link LDA: Joint Models of Topic and Author Community. In *ICML*, 2009.

[107] D. Lowe and R. Matthews. Shakepeare vs. Fletcher: A stylometric analysis by radial basis functions. *Computers and the Humanities*, 29(6):449–461, 1995.

[108] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 387–396, 2006.

[109] C. Macdonald and I. Ounis. Expertise drift and query expansion in expert search. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 341–350, 2007.

[110] C. Macdonald and I. Ounis. Learning models for ranking aggregates. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 517–529, 2011.

[111] S. Mann, D. Mimno, and A. McCallum. Bibliometric impact measures leveraging topic analysis. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '06, pages 65–74, 2006.

[112] M. Maron, S. Curry, and P. Thompson. An Inductive Search System: Theory, Design and Implementation. In *IEEE Transaction on Systems, Man and Cybernetics, vol. SMC-16, No.1*, pages 21–28, 1986.

[113] D. Mattox, M. Maybury, and D. Morey. Enterprise expert and knowledge discovery. . In *In proceedings of the 8th International Conference on Human-Computer Interaction (HCI International'99)*, pages 303–307, 1999.

[114] D. Mattox, K. Smith, and L. Seligman. Software Agents for Data Management. In C. Press, editor, *In Thuraisingham, B. Handbook of Data Management*, pages 703–722, 1998.

[115] A. McCallum. MALLET: A Machine Learning for Language Toolkit. In *http://mallet.cs.umass.edu*, 2002.

[116] A. McLean, A. Vercoustre, and M. Wu. Enterprise PeopleFinder: Combining Evidence from Web Pages and Corporate Data. In *In The 8th Australasian Document Computing Conference (ADCS'03)*, 2003.

[117] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, CSCW '02, pages 116–125, 2002.

[118] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic Modeling with Network Regularization. In *WWW*, 2008.

[119] Q. Mei and C. Zhai. A mixture model for contextual text mining. In *KDD*, pages 649–655, 2006.

[120] T. Mendenhall. The characteristics curves of composition. *Science*, 11(11):237–249, 1887.

[121] D. Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Inf. Retr.*, 10(3):257–274, June 2007.

[122] D. A. Metzler. Automatic feature selection in the markov random field model for information retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, 2007.

[123] D. Mimno and A. McCallum. Mining a digital library for influential authors. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '07, pages 105–106, 2007.

[124] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *UAI*, 2002.

[125] A. Mockus and J. Herbsleb. Expertise browser: a quantitative approach to identifying expertise. In *Proceedings of the 24th International Conference on Software Engineering*, ICSE '02, pages 503–512, 2002.

[126] C. Moreira. Learning to rank academic experts, 2011.

[127] F. Mosteller and D. Wallace. Inference and Disputed Authorship: The Federalist. Addison-Wesleym Reading, Mass., 164.

[128] R. Nallapati, A. Ahmed, E. Xing, and W. Cohen. Joint Latent Topic Models for Text and Citations. In *KDD*, 2008.

[129] Netflix. Netflix homepage. `https://www.netflix.com`.

[130] L. Nie, B. D. Davison, and X. Qi. Topical link analysis for web search. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 91–98, 2006.

[131] Z. Nie, Y. Zhang, J. Wen, and W. Ma. Object-level ranking: bringing order to web objects. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 567–574, 2005.

[132] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. In *Stanford InfoLab, Technical Report 1999-66*, 1998.

[133] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDD Cup and Workshop*, 2007.

[134] D. Petkova and W. Croft. Hierarchical language models for expert finding in enterprise corpora. In *ICTAI 2006*, pages 599–608, 2006.

[135] M. Pham, Y. Cao, and R. Klamma. Clustering Technique for Collaborative Filtering and the Application to Venue Recommendation. In *Proc. of I-KNOW*, 2010.

[136] M. Pham, Y. Cao, R. Klamma, and M. Jarke. A clustering approach for collaborative filtering recommendation using social network analysis. *Journal of Universal Computer Science*, 17(4):583–604, 2011.

[137] G. Pinski and F. Narin. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. *Information Processing and Management*, 12(5):297–312, 1976.

[138] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256. Association for Computational Linguistics, 2009.

[139] S. Rendle. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22.

[140] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 727–736, 2009.

[141] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, UAI '09, pages 452–461, 2009.

[142] Renren. Renren homepage. `https://www.renren.com`.

[143] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of*

the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, pages 175–186, 1994.

[144] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An Open Architecture for Collaborative Filtering of Netnews. In *ACM CSCW*, pages 175–186, 1994.

[145] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58.

[146] S. Robertson. Overview of okapi projects. *Journal of Documentation*, 53:3–7, 1997.

[147] M. Rosen-Zvi, T. G. ad M. Steyvers, and P. Smyth. The Author-Topic Model for Authors and Documents. In *UAI*, 2004.

[148] F. Ruane and R. Tol. Rational (successive) h-indices: An application to economics in the republic of iremand. *Scienctometrics*, 2008.

[149] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 791–798, 2007.

[150] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. 1986.

[151] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[152] M. Schwarts and D. Wood. Discovering shared interests using graph analysis. In *Communications of the ACM, 36(8)*, pages 78–89, 1993.

[153] P. Serdyukov, H. Rode, and D. Hiemstra. Modeling multi-step relevance propagation for expert finding. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 1133–1142, 2008.

[154] Y. Seroussi, I. Zukerman, and F. Bohnert. Authorship Attribution with Latent Dirichlet Allocation. In *CoNLL*, pages 181–189, 2011.

[155] B. Shaparenko and T. Joachims. Identifying the original contribution of a document via language modeling. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 350–365, 2009.

[156] B. Shaparenko and T. Joachims. Identifying the original contribution of a document via language modeling. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 696–697, 2009.

[157] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. Tfmap: optimizing map for top-n context-aware recommendation. In *SIGIR*, pages 155–164, 2012.

[158] A. Sidiropoulos, D. Katsaros, and Y. Manolopoulos. Generalized hirsch h-index for disclosing latent facts in citation networks. *Scientometrics*, 72:253–280, 2007.

[159] B. Sigurbjornsson and R. Zwol. Flickr tag recommendation based on collective knowledge. In *WWW'08*, pages 327–336, 2008.

[160] Sina. Weibo homepage. `https://www.weibo.com`.

[161] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *KDD*, 2008.

[162] Y. Song, J. Huang, and I. Councill. Efficient topic-based unsupervised name disambiguation. In *JCDL*, pages 342–351, 2007.

[163] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W. Lee, and C. Giles. Real-time automatic tag recommendation. In *SIGIR'08*, pages 515–522, 2008.

[164] D. Sorokina, R. Caruana, and M. Riedewald. Additive groves of regression trees. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, pages 323–334, 2007.

[165] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic text categorization in terms of genres and author. *Comp. Ling*, 26(4):471–495, 2001.

[166] L. Steeter, , and K. Lochbaum. An Expert/Expert Locating System based on Automatic Representation of Semantic Structure. In C. S. of the IEEE, editor, *in Proceedings of the Fourth IEEE Conference on Artificial Intelligence Applications*, pages 345–349, 1988.

[167] T. Strohman, W. B. Croft, and D. Jensen. Recommending citations for academic papers. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 705–706, 2007.

[168] J. Tang, R. Jin, and J. Zhang. A Topic Modeling Approach and its Integration into the Random Walk Framework for Academic Search. In *ICDM*, 2008.

[169] J. Tang and J. Zhang. A discriminative approach to topic-based citation recommendation. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pages 572–579, 2009.

[170] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. ArnetMiner: extraction and mining of academic social network. In *KDD*, 2008.

[171] C. Tantipathananandh, B. Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 717–726, 2007.

[172] R. Torres, S. McNee, M. Abel, J. Konstan, and J. Riedl. Enhancing digital libraries with techlens. In *Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on*, pages 228 – 236, 2004.

[173] Y. Tu, N. Johri, D. Roth, and J. Hockenmaier. Citation Author Topic Model in Expert Search. In *COLING*, 2010.

[174] F. Tweedie and R. Baayen. How variable may a constant be? Measures of Lexical Richness in Perspective. *Computers and the Humanities*, 32:323–352, 1998.

[175] Twitter. Twitter homepage. `https://www.twitter.com`.

[176] T. Tylenda, R. Angelova, and S. Bedathur. Towards time-aware link prediction in evolving social networks. In *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, SNA-KDD '09, pages 9:1–9:10, 2009.

[177] W. Hsu and A. King and M. Paradesi and T. Pydimarri and T. Weninger. Collaborative and structual recommendation of friends using weblog-based social network analysis. In *AAAI Spring Symposium '06*, 2006.

[178] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, 2011.

[179] J. Wang, Z. Chen, L. Tao, W. Ma, and L. L. Wenyin. Ranking user's relevance to a topic through link analysis on web logs. In *WIDM 2002*, pages 49–54, 2002.

[180] J. Wang, X. Hu, X. Tu, and T. He. Author-conference topic-connection model for academic network search. In *CIKM*, pages 2179–2183, 2012.

[181] X. Wang, J. Sun, Z. Chen, and C. Zhai. Latent semantic analysis for multiple-type interrelated data objects. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 236–243, 2006.

[182] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)*. Structural analysis in the social sciences, 8. Cambridge University Press, 1 edition, Nov. 1994.

[183] M. Weimer, A. Karatzoglou, Q. Le, A. Smola, and Others. COFIrank-maximum margin matrix factorization for collaborative ranking. In *NIPS*, 2007.

[184] J. Weng, E. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 261–270, 2010.

[185] W. Xi, B. Zhang, Z. Chen, Y. Lu, S. Yan, W. Ma, and E. Fox. Link fusion: a unified link analysis framework for multi-type interrelated data objects. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 319–327, 2004.

[186] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR*, SIGIR '07, pages 391–398, New York, NY, USA, 2007. ACM.

[187] Yahoo! Delicious homepage. `https://delicious.com`.

[188] Z. Yang and B. Davison. Distinguishing Venues by Writing Styles. In *JCDL*, 2012.

[189] Z. Yang and B. Davison. Venue recommendation: Submitting your paper with style. In *ICMLA (1)'12*, pages 681–686, 2012.

[190] Z. Yang, L. Hong, and B. Davison. Topic-driven multi-type citation network analysis. In *RIAO*, 2010.

[191] Z. Yang, D. Yin, and B. Davison. Award prediction with temporal citation network analysis. In *SIGIR*, SIGIR '11, pages 1203–1204, 2011.

[192] D. Yimam and A. Kobsa. Demoir: A hybrid architecture for expertise modeling and recommender systems. In *Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, WETICE '00, pages 67–74, 2000.

[193] D. YimamSeid and A. Kobsa. Expert finding systems for organizations: Problem and domain analysis and the demoir approach. In *Organizational Computing and Electronic Commerce, 13(1)*, pages 1–24, 2003.

[194] D. Yin, S. Guo, B. Chidlovskii, B. D. Davison, C. Archambeau, and G. Bouchard. Connecting comments and tags: improved modeling of social tagging systems. In *WSDM*, pages 547–556, 2013.

[195] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang. Geographical topic discovery and comparison. In *WWW*, pages 247–256, New York, NY, USA, 2011. ACM.

[196] Z. Yin, M. Gupta, T. Weninger, and J. Han. Linkrec: a unified framework for link recommendation with user attributes and graph structure. In *WWW*, pages 1211–1212, 2010.

[197] Z. Yin, M. Gupta, T. Weninger, and J. Han. A unified framework for link recommendation using random walks. In *ASONAM*, pages 152–159, 2010.

[198] Youtube. Youtube homepage. `http://www.youtube.com`.

[199] B. Yu and M. Singh. Searching Social Networks. In *AAMAS'03*, 2003.

[200] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 271–278, 2007.

[201] G. Yule. The Statistical Study of Literary Vocabulary. Cambridge University Press, 1944.

[202] C. Zhang. The e-index, complementing the h-index for excess citations. *PLos One*, 4(5):1–4, 2009.

[203] J. Zhang, M. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *In Proceedings of WWW'2007*, pages 221–230, 2007.

[204] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. In *DASFAA*, pages 1066–1069, 2007.

[205] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3):378–33, 2006.

[206] D. Zhou, A. Orshanskiy, H. Zha, and C. Giles. Co-ranking authors and documents in a heterogeneous network. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 739–744, 2007.

[207] D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. L. Giles. Learning multiple graphs for document recommendations. In *Proceedings of the 17th international conference on World Wide Web*, WWW, pages 141–150, 2008.

[208] J. Zhu, A. Ahmed, and E. P. Xing. Medlda: maximum margin supervised topic models for regression and classification. In *ICML*, pages 1257–1264, New York, NY, USA, 2009. ACM.

[209] G. Zipf. Human Behaviour and the principle of least effort. An introduction to human ecology. Houghton-Mifflin, 1932.

Vita

Zaihan Yang was born in Changsha, Hunan, P.R. China. She completed her Ph.D. in Computer Science from Lehigh University, PA, USA in May, 2014. She holds a Bachelor of Engineering in Computer Science and Engineering from Central South University, Changsha, Hunan, China, a Master of Engineering in Computer Science and Engineering from Central South University, Changsha, Hunan, China, and a Master of Science in Computer Science from Lehigh University, USA. Her primary research interests include information retrieval, data mining, machine learning, social network/social media analysis, citation network analysis, web search and web mining.

LIST OF PUBLICATION

2013 **Z. Yang** and L. Hong and B. D. Davison. Academic Network Analysis: A Joint Topic Modeling approach. In Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 324-333, 2013

2012 **Z. Yang** and B. D. Davison. Writing with Style: Venue Classification. In Proceedings of the International Conference on Machine Learning and Applications (ICMLA), 250-255, 2012

2012 **Z. Yang** and B. D. Davison. Venue Recommendation: Submitting your Paper with Style. In Proceedings of the International Conference on Machine Learning and Applications (ICMLA), 681-686, 2012

2012 **Z. Yang** and B. D. Davison. Distinguish Venues by Writing Styles. In Proceedings of ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL), 371-372, 2012

2011 **Z. Yang**, D. Yin and B. D. Davison. Award prediction with temporal citation network analysis. In Proceedings of Special Interest Group on Information Retrieval (SIGIR), 1203-1204, 2011

2010 **Z. Yang**, L. Hong and B. D. Davison. Topic-driven Multi-type Citation Network Analysis. In Proceedings of the International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information (RIAO), 24-31, 2010

2009 L. Hong, **Z. Yang** and B. D. Davison. Incorporating Participant Reputation in Community-driven Question Answering Systems. In Proceedings of the Symposium on Social Intelligence and Networking (SIN), held in conjunction with IEEE SocialCom-09, 475-480, 2009