A Genetic Programming Based Algorithm for Web Mining

J. Aguilar and J. Altamiranda
Universidad de los Andes, Facultad de Ingeniería, Departamento de Computación, CEMISID,
Mérida, Venezuela, 5101

Abstract

Data Mining is composed by a set of methods to extract knowledgement from large database. One of these methods is Genetic Programming. In this work we use this method to build a Data Mining System that define a set of patterns in order to classify the data. We define a grammar, which is used by the Genetic Programming in order to define the rules that represent the patterns. In this way, we can group the data in class and simplify the information in the database according to the set of patterns.

Keywords: Data Mining, Genetic Programming, Clustering.

1. Introduction

The large utilization of distributed systems has generated an explosion in the quantity of available information, which should be processed by sophisticated mechanisms to be able to be used efficiently. A technique for that is Data Mining, whose goal is to discover knowledge from a group of data [7, 8, 9]. Data Mining and Knowledge Discovery are topics of interest for researches in areas such as: database, automatic learning, inductive logical programming, artificial intelligence, among others. This work bellongs to areas of Data Mining and Knowledge Discovery in the web. In particular, we will be interested in the knowledge discovered described as classification rules "IF <CAUSES> - THEN <CONSEQUENCES> ". CAUSES contains a logical combination of conditions, and CONSEQUENCES contains attributes that are originates by the CAUSES. There are several paradigms for Data Mining. In this work we are going to work with the Genetic Programming. The utilization of Genetic Programming for the definition of classification rules, in the domain of Data Mining is a relatively unexplored area [8]. This is a promising approach, due to the efficiency of the Genetic Programming in the search and construction of patterns adapted to an environment according to a given structure (in our case, a grammatical structure). Specifically, in this work will be built a Data Mining System based on Genetic Programming to discover patterns, which are defined like classification rules. The Genetic Programming uses a grammatical structure that defines the form of the rules.

2. Theoretical Aspects

2.1 Data Mining

Data Mining groups the techniques and tools used to extract useful information from large databases. In general, the techniques obtain patterns or models using the gathered data. This process involves a data analysis, the patterns recognition from a group of data and a classification of them. The Data Mining is one phase of the process of extraction of information known as KDD (Knowledge Discovery in Databases). This process include, also, the data preparation and the results interpretation. There are differents techniques that can be used in Data Mining, for example: Decision Trees, Artificial Neuronal Network, Genetic Programming, Fuzzy Logic, others. Data Mining techniques transform the data in useful information [7, 8, 9].

2.2 Grammatical Strutures

The grammatical structures define the format of a given language [2]. that is, it is a formal notation that describes like must be integrated the elements that compose a language. It consists of a sequence of declarations of the allowed operations. Grammatical structures grants flexibility to the format of the rules and allows to form interesting and regular patterns. To be able to use a grammatical structure in a problem, the next phases must be follows:

- **Lexical analysis:** in this phase, the input is a sequence of characters and the output is a sequence of atoms. The atoms are entities that identify a logical sequence of characters [2]. Some types of atoms are: names (identifiers), spaces, operators (+, -, =, <=), numbers (integer, real), others.
- **Syntactic analysis:** in this phase, the input is a sequence of atoms and is verify if they can be generated by the

grammar [2]. The grammar is composed by a group of rules. Examples of these rules can be, for the case of a identifier:

Letter = a | A | b | B | ... | z | Z
digit = 0 | 1 | 2 | 3 | ... | 9
Letter_or_Digit = letter | digit
Identifier = Letter(Letter_or_Digit)*
Where : | means or
( )* means repetitions

– **Semantic analysis:** in this phase is guaranted that the grammatical components built in the previas phase are associated to a programming logic.

2.3 Genetic Programming

Evolutionary Computation (EC) is based on the theories of the natural evolution and of the genetic. This is one alternative to solve complex problems in diverse areas througth the adaptation of computational structures. It uses a population of possible solutions to a given problem, which evolves in each generation. EC combines the best individuals of the populations to transmit theirs characteristics to theirs descendants [1]. Genetic Programming is one of the techniques of the EC. Genetic Programming was created by John Koza at the end of the 80 [1, 3, 4, 5]. In this technique, the individuals represent a potential procedures to solve a given problem. Each individual has a value (fitness funtion), that indicates its quality. New individuals are generated by procedures similar to the biological reproduction, where the parents are selected according to theirs qualities. The evolutionary operators are crossover, mutation, among others. The new individuals replace the worst individuals of the current population, in this way, the population average quality will improve in each generation. In adition to the classical parameters of the EC, to use the Genetic Programming we need to define:

❑ **The set of functions and terminals:** The group of terminals are the atoms of the grammar: numbers, mathematical operators, identifiers, among others. The group of functions can be arithmetic operations, logical operations, others example: (if, then, else, while, for).

❑ **The Individuals:** The individuals are define for structures trees. In general, these structures are formed by nodes that represent functions and terminal, which are specific for each problem.

3. Our Data Mining System

3.1 Main characteristics

The Data Mining System is composed by three elements: a) A general grammatical structure, b) The database that contains the information to be classified, c) The algorithm that carries out the patterns construction and recognition, based on the Genetic Programming.

Our Data Mining System will work according to the following procedure:

1. Definition of a general grammar to be used by the Genetic Programming. The general grammar describes the way to build the classificates rules. It guarantees the construction of valid individuals.
2. Extraction of the atoms for the general grammar from the database studied. The system, must select atoms from the database in order to build the clasificates rules.
3. Utilization of the Genetic Programming to build and to evaluate the classification rules. In this case, each classfication rule will be an individual, which will be evaluated according to its aptitude to classify a large number of information. Also, the Genetic Programming provides new individuals using its evolutionary mechanisms.

This procedure is repeat from the step 2 until the group of obtained rules is able to classify the biggest quantity of information of the database. The system can incorporate new information from the database, in form of atoms, of will in this it enrichs the group of rules to be generated. This is necessary because the generated rules can be incomplete, that is, they can cover alone a small fraction of the database. On the other hand, although invalid rules can also be generated, the Genetic Programming eliminates them. The classification rules generated by our of Data Mining System have the next properties:

1. They have different grade of accuracy. If the accuracy is 1, it means that the rule represents all the informations of the database. If the accuracy is near to 1, it is a strong rule. If the accuracy of the rule is not very high, then it is a weak rule. Our Data Mining System should not discover alone strong rules, because the weak rules can also give useful knowledge [9].
2. They are comprehensible: in our case this concepts is associated with the simplicity of the rules. It consists on the number of elements that compose each individual [8].
3. They identify relationships among the atoms of the database which are not known previously.

3.2 The General Gramatical Structures

To build the classification rules is necessary to define the general grammar. This is one of the main point of our System, because the rules generated by the Genetic Programming will depend on it. The initial population of the Genetic Programming is created using the general grammar. The grammar is a powerful representation of the knowledge and grants flexibility for the construction of the rules. The grammar allows a natural combination of the atoms for the construction of patterns, which follow the generic structure defined by the group of productions of the grammar [10]. The general grammar model to build a group of classification rules is:

Operator → AND | OR | NOT

Attribute_Name → Name of one of the fields of the studied database (atom).

Attribute_Name: Attribute_Value → Random value generated from the domain of that atom | Value taken from the database of that atom.

Attribute → Attribute_Name:Attribute_Value

Cause → Attribute(Operator Attribute)*

Consequence → Attribute.

Rule → IF **Cause** THEN **Consequence**

The general grammar presented previously is the ideal for any type of problems. Particularly, it is important in a Data Mining System this must the relationships among the values of the fields of a database and as they generate the occurrence of a value in another field. In our case, in which patterns are building to data clustering, the consequence is not important. In this way, we have modified the general grammar to adapt it to our problem:

Operator → AND | OR | NOT

Attribute_Name → Name of one of the fields of the studied database (atom).

Attribute_Name: Attribute_Value → Random value generated from the domain of that atom | Value taken from the database of that atom.

Attribute → Attribute_Name:Attribute_Value

Cause → Attribute(Operator Attribute)*

Rule → **Cause**

3.3 Components of the Genetic Programming

- **Fitness Function:** The fitness function evaluates each individual (the rules) of the population in the classification problem. In our system, the fitnees function combines two indicators, the Predictive Accuracy (PA) and the Simplicity (Sim). In the case of the Predictive Accuracy (PA), using the attributes of the cause of the rule the system searches on the database the register that contain these terms. The system sums these registers, and this value is divided among the number of record on the database:

$$PA = \frac{\text{Number of activated registers of the database}}{\text{Number of registers of the database}}$$

Maybe, the Genetic Programming will produce simplex rules. Considering that a rule is comprehensible if its size is small, we prefer rules the shortest possible. Therefore, we define a second function that measures the Simplicity of a rule (Sim), given by [8]:

$$Sim = \frac{max\_ elt\_ rule - 0.5 * num\_ elt\_ rule - 0.5}{max\_ elt\_ rule - 1}$$

Where num_elt_rule is the number of elements in the rule, and max_elt_rule is the maximum number of elements allowed in a rule. Maximum value of the equation is 1.0 when a rule is so simple that it contents only a term. The value of the equation falls until a minimum value of 0.5, that is produced when the number of nodes is similar to the maximum allowed. The reason for which the minimum value of Sim is 0.5 is to penalize individuals of great size without force to disappear them. This is important for the individuals of the first generations, when many of them have very low Predictive Accuracy (PA), but they can have good genetic material [8]. In this way, the objective of the Genetic Programming is to maximize the Predictive Accuracy and to minimize the size of the rule simultaneously. In this way, the fitnees function used by our System is:

$$\text{Fitness Funtion} = (\text{PA x Sim})$$

- **Structures of the individuals:** An individual is a classification rule. Consequently, the structure of tree describes a possible rule, which is formed by a group of extracted atoms from the database that constitutes the terminals, and the functions AND|OR that are used to define the relationships among the terminals.

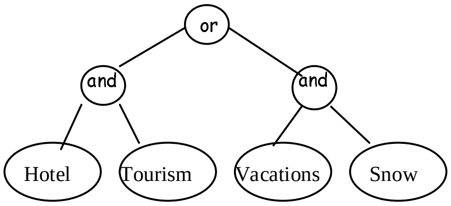

Figure 1. Individual Structure. (Classification Rule: ((HOTEL AND TOURISM) OR (VACATIONS AND SNOW)))

- **Set of terminals and functions:** The set of terminals and of functions should be able to express the solution of a problem. On the other hand, the functions should accept like argument any value of the database that can be used as terminal, just as it was defined in the general grammar. The set of terminals(atoms) consists of attributes from the database. The set of functions consists of the functions AND and OR. That is, an individual (rule) consists of a combination of the functions applied to the atoms.
- **The population size:** In our system, that can be between 0 and 99.999 [6]. A high value for the population size would be ideal because it would guarantee us an enormous diversity of rules, but with a large computational cost.
- **Number of generations:** In our system, that can be between 1 and 32.000 [6]. Normally, a large number of generations, could find better rules, but it can take place that after certain number of generations the group of rules doesn't improve.
- **Genetic Operators:** Our system uses three genetic operators: crossover, mutation and copy.

4. Experiment

4.1 Case of study

Internet is a large space of information that constantly grows. There is a large number of places that need to be visited and classified when must be made a search. There are powerful search tools that find information, such as, Altavista, Yahoo, Google, etc. Keywords are introduced and they determine web site that contain these words, trying to satisfy the user's requirements. Sometimes these tools bring inconsistent documents that fulfill the search requeriment, but not the user's interest. For this reason, there is a new approach called Web Mining, that is an extension of Data Mining to discover web pages in Internet, to extract patterns of them to classify the information. Web Mining offers advantages for the users that search a topic or for the people that design the pages. For the users because it simplifies and classifies the information that is presented. For the designers because with this information they can understand the behavior and the requirements of the users, and in this way, to improve the design of the links of the pages. In this way, our system obtains classification rules from the web pages gathered by Yahoo, Google, etc. The classification rules allow to classify and categorize the information.

4.2 Simulations

Our system has been made using the software "Estudio de Programación Genética" [6]. We have used Google to search the web pages with the words "Merida" and "Venezuela." We have obtained 8.640 pages. With this pages we have built a text file that contains: Title of the page, Address, and Description. We have cleaned the data eliminating repeated information and addresses of inconsistents web pages. With this file we have built the classification rule using our system. The atoms were chosen from the words more frecuent of the text file. The parameters of the Genetic Programming used for the construction of the classification rules are: a) Number of atoms {4, 6, 8, 10, 12, 14}, b) Individuals' number {10, 20, 30, 40, 50, 80, 100, 200, 300}, c) Number of generations {200, 400, 500, 600, 800, 1000}, d) Crossover probability {0.9}, e) Copy probability {0.1}, f) Mutation probability {0.01}, g) Maximum number of elements in an individual = {20, 50}. The values of crossover, copies and mutation probabilites were not varied, because the different values of these parameters don't produce differents results. In this section we are going to show the main results , the rest of the results for different cases of study can be found in [10].

4.1.1 Test with 10 atoms

For these initial test, the atoms are: CONGRESSES, CABLE CAR, TOURISM, HOTEL, VACATIONS, STUDENTS, MOUNTAINS, SNOW, UNIVERSITY, ANDES.

The figure 2 shows that when is increment the number of individuals the diversity of the rules is bigger, from 0.4 different rules for 10 individuals to 0.8 different rules for 40 individuals. This is due to that with a large number of individuals the Genetic Programming has a bigger search space to build different individuals.

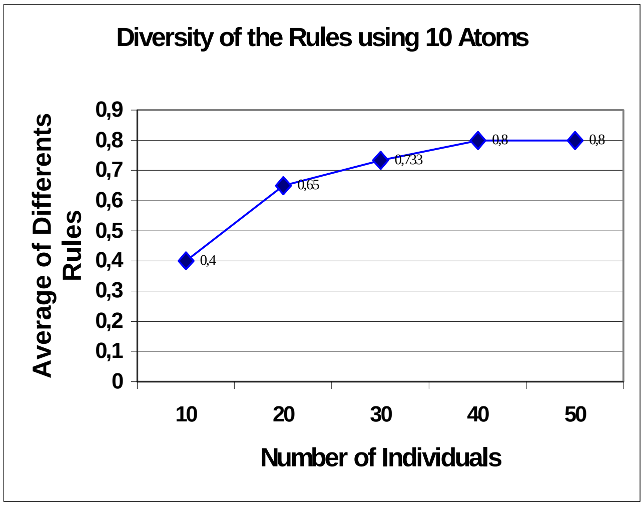

Figure 2. Diversity of the Rules using 10 Atoms.

The figure 3 shows that when is increased the number of individuals the Predictive Accuracy Average grows. The reason is because with a large number of individuals the system can explore a large number of possible combinations to built better classification rules.

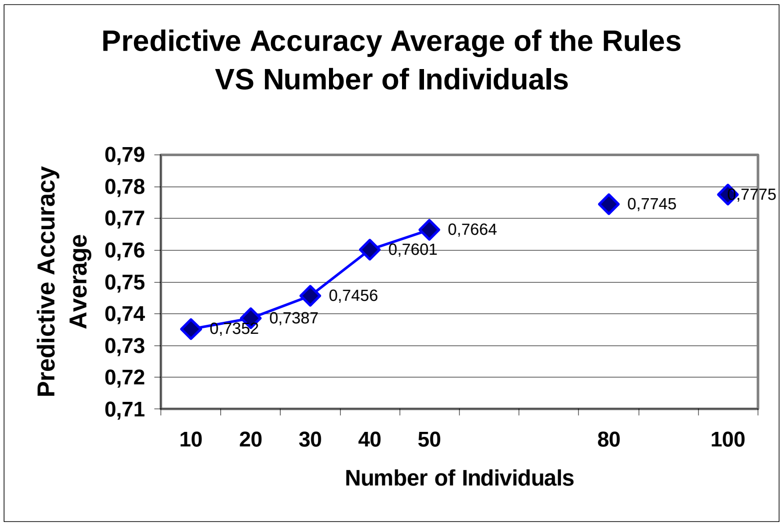

Figure 3. Predictive Accuracy Average of the Rules VS Number of Individuals.

The figure 4 shows that the Predictive Accuracy Average of the Rules goes from 0,7434 in 200 generations to 0,7845 in 1000 generations. This way, the classification rules improve the Predictive Accuracy.

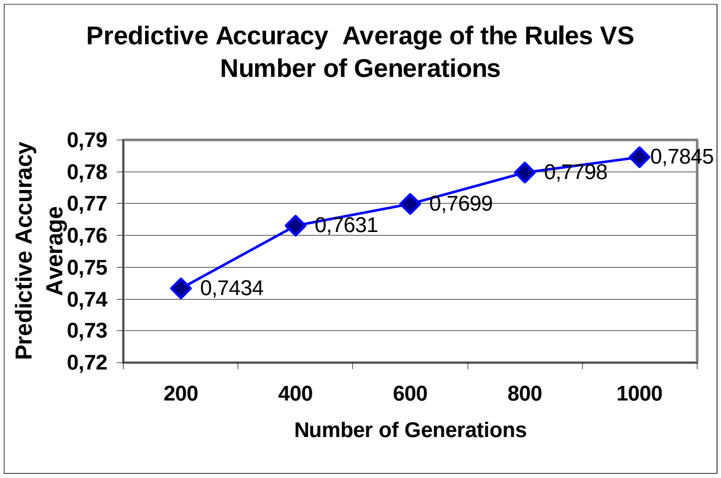

Figure 4. Predictive Accuracy Average of the Rules VS Number of Generations

RULE	PE	Sim	PE x Sim
(((UNIVERSITY AND ANDES) OR MOUNTAINS) OR CONGRESS)))	0,7607	0,84	0,6389
((UNIVERSITY AND STUDENTS) OR (CONGRESS AND ANDES))	0,7514	0,84	0,6311
((CABLE CAR AND CONGRESS) OR (UNIVERSITY AND ANDES))	0,7422	0,84	0,6234
((ANDES AND STUDENTS) OR (VACATIONS AND TOURISM))	0,7393	0,84	0,6210
(((UNIVERSITY AND ANDES) OR (STUDENTS AND CONGRESS)) OR (TOURISM AND MOUNTAINS))	0,7961	0,78	0,6209
(((UNIVERSITY AND ANDES) OR (STUDENTS AND UNIVERSITY)) OR (TOURISM AND SNOW))	0,7885	0,78	0,6150
((UNIVERSITY AND SNOW) OR (HOTEL AND VACATIONS))	0,7223	0,84	0,6067
((((MOUNTAINS OR (TURISMO AND ANDES)) OR (UNIVERSITY AND ANDES)) OR (CONGRESS AND STUDENTS))	0,8006	0,68	0,5444
(((TURISMO OR (SNOW AND MOUNTAINS)) OR (UNIVERSITY AND CONGRESS)) OR (STUDENTS AND VACATIONS))	0,7719	0,68	0,5248

Table 1. Predictive Accuracy and Simplicity of the best rules using 10 Atoms.

The Table 1 shows that the rules defined by our system can classify more than 70% of the registers of the file, where some of them can classify 80%, but with 20% of the registers with out not able be classifed. On the other hand, most of the rules have few elements. Particularly, the rules with Predictive Accuracy higher have a large number of words. That is, these rules contain the most representative words that appear in the web page with the information about the Merida State in Venezuela, for 80% of the cases. Finally, the rules more efficient can be divided in several sub-rules where each one contributes with a percentage to the value of the Predictive Accuracy.

For example, in the Table 1, the best classification rule:

(((UNIVERSITY AND STUDENTS) OR MOUNTAINS) OR CONGRESS)).

Can be divided as:

UNIVERSITY AND STUDENTS
MOUNTAINS
CONGRESS

Each one of the sub-rules describes a data clustering.

4.1.2 Comparison with other works

In this part we present a set of results for a group of tests that were made with the purpose of compare the Predictive Accuracy (PA) that can be obtained with our System with other works [8].

Nº of Atoms	N° of Generations	Nº of Individuals	Rule	PÂ	Execute Time(Hrs)
12	500	200	(STUDENTS AND UNIVERSITY) OR (HOTEL AND TRAVEL) OR (CONGRESS AND STUDENTS) OR (CONGRESS AND ACCOMMODATIONS) OR (TOURISM AND HOTEL) OR (STUDENTS AND ACCOMMODATIONS) OR (CABLE CAR AND MOUNTAINS) OR (UNIVERSITY AND ANDES)	0.8262	1.5
14	500	300	(HOTEL AND ACCOMMODATIONS) OR (CONFERENCES AND CONGRESS) OR (CONFERENCES AND STUDENTS) O (RESTAURANT AND VACATIONS) OR (VACATIONS AND TOURISM) O (SNOW AND CABLE CAR) OR (UNIVERSITY AND ANDES) OR (TRAVEL AND TOURISM) OR (STUDENTS AND CONGRESS) OR (TOURISM AND MOUNTAINS) OR (SNOW AND TOURISM)	0.8806	2.5

Table 2. Predictive Accuracy for some special cases.

The Table 2 shows that a rule was obtained with a Predictive Accuracy (PA) value equal to 0.8860, that is, the rule classifies 311 records of the 351 in the file. If we compares our System with [8], the best value of Predictive Accuracy (EP) obtained was of 0.875 for a database of 48 records. We see that our system improves this result. For these cases, the execution time of our system is large, with a large of elements on the rules. The reason of the large execute time is due to the procedures of search of the key words in the text file (This procedure is made sequentially).

5. Conclusions

The Data Mining is a tool to explorate and not to explain. Our system follow this idea. Particularly, our system extracts patterns from a database and define classification rules. The system uses an explicit representation of the knowledge (general grammar), and able to discover relationships that could not be found by a human.

The results are very promising, since the system discovers comprehensible rules. But, the efficiency of our system depends of the atoms. On the other hand, the rules can be divided in several sub-rules. Each one of the sub-rules represents a data clustering. Among the possible extensions of the System we can mention:

1. The System acts over on data previously gathered in a file, that is, the data don't change while they are being analyzed. Corrently, the information in environments like Internet constantly changes. Our System should adapt to these changes of the information through of rules that can adapt dinamically. The procedure of the section 3.1 can be adjusted for this situation.
2. The system could be used to find a group of rules that describe properties of the data kept in a database. The general grammar propose in the section 3 were made to carry out this task.

This exploration of data, would allow to find relationships of the type:

If Buy: Bread and Buy: Sausages

Then Purchase: Mustard

If Person: Young and Carry out: Deport

Then Practice: Soccer.

That is, we can defines rules that when the CAUSE of the rule is satisfied, for a certain group of attributes with specific values, they generate as CONSEQUENCE a given situation. This rules should be analyzed to make more precise decisions. For example, a possible conclusion from the previous rule is solding the sausages and the breads, and to going up the price of the mustard.

References

[1] Aguilar, J. and Rivas, F. (Ed.), *Introducción a la Computación Inteligente*, Meritec, Mérida–Venezuela, 2001.

[2] Aguilar, J. *Compiler's Course*, Houston University, USA, Summer 2000.

[3] Kinnear, K. (Ed.), *Advances in Genetic Programming*" The MIT Press, 1994.

[4] Koza, J. *Genetic Programming: On the programming of Computers by Means of Natural Selection*, The MIT Press, 1992.

[5] Koza, J. *Genetic Programming II: Automatic Discovery of Reusable Programs*, The MIT Press, 1994.

[6] *Manual de Usuario. Estudio de Programación Genética*, Universidad de Córdoba, España, 1998.

[7] Kovalerchuk, B. Vityaev, E. and Ruiz, J. Consistent Knowledge Discovery in Medical Diagnosis, *IEEE Engineering in Medicine and Biology*, Vol. 4, (July 2000), pp. 26 – 37.

[8] Bojarczuk, C. Lopes, H. Freitas, A. Genetic Programming for Knowledge Discovery in Chest – Pain Diagnosis, *IEEE Engineering in Medicine and Biology*. Vol. 4, (July 2000), pp. 38 – 44.

[9] Wong, M. Lam ,W. Kwong, L. Ngan, P. Cheng, J. Discovery Knowledge from Medical Databases Using Evolutionary Algorithms, *IEEE Engineering in Medicine and Biology*, Vol. 4, (July 2000), pp. 45 – 55.

[10] Altamiranda, J. Aguilar, J. *Construccion y Optimizacion de Estructuras Gramaticales utilizando Programación Genética*, Technical Report 20-02, CEMISID, Universidad de los Andes, Mérida – Venezuela, Julio 2002.