Using Genetic Algorithms for Data Mining in Web-based Educational Hypermedia Systems

Cristóbal Romero[1], Sebastián Ventura[1], Carlos de Castro[1], Wendy Hall[2], and Muan Hong Ng[2]

[1]Universidad de Córdoba, Campus Universitario de Rabanales, 14071, Córdoba, España
{cromero,sventura,cdecastro}@uco.es
[2]University of SouthSampton, Highfield, SO17 1BJ, SouthSampton, UK
{wh,mhn99r}@ecs.soton.ac.uk

Abstract. In this paper we show how to apply genetic algorithms for data mining of student information obtained in a Web-based Educational Adaptive Hypermedia System. The objective is to obtain interesting association rules so that the teacher can improve the performance of the system. In order to check the proposed algorithm we have used a Web-based Course developed for use by medical students. First, we will describe the proposed methodology, later the specific characteristics of the course and we will explain the information obtained about the students. We will continue on with the implemented genetic algorithm and finally with the rules discovered and the conclusions.

1 Introduction

Nowadays there is a growing trend of web-based technology applied for distance education. Particularly, Web-based Adaptive Educational Hypermedia Systems have many advantages because they can adapt the course for each specific student. But usually, the methodology used to elaborate them is static, that is, when the course elaboration is finished and published on the Internet it is never modified again. The teacher only accesses the student evaluation information obtained from the course to analyze the student's progress. We propose, a dynamic elaboration methodology, where the evaluation information is used to modify the course and to improve its performance for better student's learning. Our approach is to use a knowledge acquisition method (machine learning and data mining) to discover useful information that might help the teacher to improve the course. Nowadays data mining researches are beginning to use techniques such as Web Data Mining to evaluate web-learning activities [7]. We propose a genetic algorithm for data mining to evaluate the student information obtained from a Web-based Adaptive Hypermedia System. We have used a Web-based Hypermedia Course that was designed to be used by medical student as an example to evaluate our algorithm and to obtain association rules [4]. These rules could then be shown to the teacher in order to help him decide how the course could be modified to obtain best performance.

2 Methodology

The dynamic construction methodology of Web-based Hypermedia Courses that we propose is recurrent and evolutionary (Fig 1) and while the number of students who use the system increases, more information is available to the teacher to improve it.

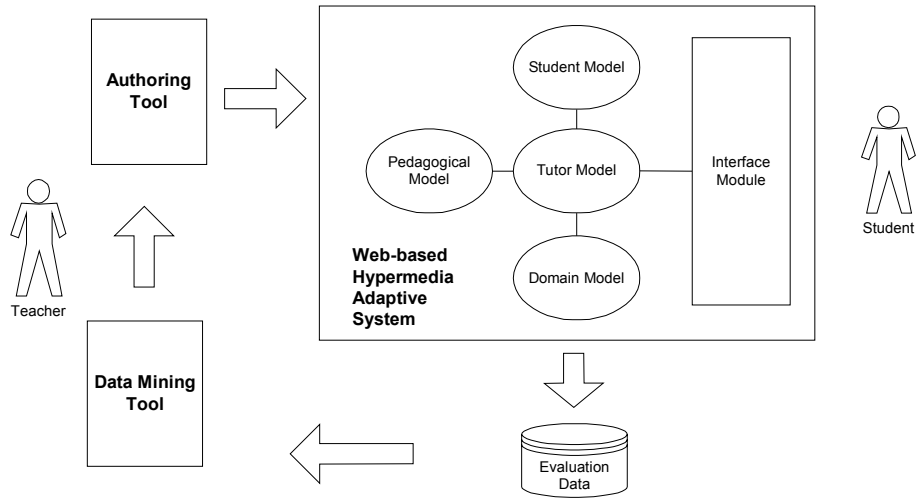

Fig. 1. Dynamic development methodology of Web-based Hypermedia Adaptive Systems.

In our methodology we can distinguish four main steps:

- Construction of the course. The teacher builds the Hypermedia Adaptive Course providing information of the domain model, the pedagogic model and the interface module. An authoring tool is usually used to facilitate this task. The remaining information, tutor model and the student model usually is given or acquired by the system itself. Once the teacher and the authoring tool finish the elaboration of the course, then, the full course's content may be published on a web server.
- Execution of the course. The students execute the course using a web navigator and in a transparent way the usage information is picked up and stored in the server in a huge database of all the students.
- Application of Data Mining. The teacher applies data mining algorithms [8] to the database to obtain important relationships among the data picked up. For this, he uses a graphical data mining tool.
- Improving the course. The teacher using the discovered relationships carries out the modifications that he believes more appropriate to improve the performance of the course. To do it, he again uses the authoring tool.

The process of execution, application and improvement can be repeated as many times as the teacher wants to do so. Although it is recommendable to have a significant amount of new students usage information before repeating it.

3 Web-based Hypermedia Medicine Course

We have used the data obtained from the evaluation of a Web-based Hypermedia Course in the study of Rheumatology. The system used to develop the course is an adaptive hypermedia system [6], but the evaluation data used for this paper was obtained from a usability study and there was no attempt to use the result to redesign the Course. As part of this evaluation the system was used by 30 users, of which 20 were medical and 10 were non-medical users. All the information was stored in a single database in the following tables: USER: String value that represents a system user, in our case they are 30. PERFORMANCE: Real value that represents the users performance in the 7 case studies in this application. AVEP_AH: Real value that represents the average performance of the users in the 7 case studies, adaptive application version. AVEP_NOAH: Real value that represents the average performance of the users in the 7 case studies, but in the version of the application without adaptation. CASETIME: Integer value that represents the time that a user takes in visualizing a complete case study. CASESCORE: Integer value that represents the score that an user has obtained when undertaking a case study. ACCESSTIME: Real value that represents the number of times a user has accessed the application. CONCEPT: Real value that represents the user's effort spent in the different concepts. QUESTIONSCORES: Integer value that represents the score obtained by the users in the relating questions to the case studies.

The data was preprocessed so that it will be easier to obtain relationship rules from them. This transformation consisted of a discretization, which mapped from continuous values (usually real values) to discrete values (strings that represent values groups) and integer values only needed to be labeled. In this way the modifications made to the tables are as follows: PERFORMANCE, AVEP_AH, AVEP_NOAH, CASETIME, ACCESSTIME and CONCEPT have been discretized to the labels VERYHIGH, HIGH, MEDIUM, LOW and VERYLOW. The values of CASESCORE and QUESTIONSSCORES have been named with the labels SUCCESSFIRST, SUCCESSSECOND, SUCCESSTHIRD and SUCCESSFOUR, which means getting the answer correct at the first attempt, second attempt and so forth. USER does not need modification.

4 Genetic Algorithm for Data Mining

Some of the main data mining tasks are [8]: classification, clustering, discovery of association rules, etc. We have used a genetic algorithm to obtain association rules from the user evaluation data. The association rules relate variable values. They are more general than classification rules due to the fact that in association rules any variable may be in the consequent or antecedent part of the rule. The classical problem of discovering association rules is defined as the acquisition of all the association rules between the variables [4]. Genetic algorithms are a paradigm based on the Darwin evolution process, where each individual codifies a solution and evolves to a better individual by means of genetic operators (mutation and crossover). In general the

main motivation for using genetic algorithms for rule discovery is that they perform a global search and cope better with attribute interaction than greedy rule algorithms often used in data mining [3]. Most data mining methods are based on the rule induction paradigm, where the algorithm usually performs a kinds of local search (hill climbing). Also the fitness function in genetic algorithm evaluates the individual as a whole, i.e. all the interactions among attributes are take into account. In contrast, most rule induction methods select one attribute at a time and evaluate partially constructed candidates rule, rather than full candidate rule. The Genetic Process we have used consists of the following steps [5]: The first step is Initialization, next Evaluation, Selection and Reproduction steps are repeated until the Finalization condition is fulfilled.

4.1 Initialization

Initialization consists of generating a group of initial rules specified by the user (50 - 500 rules). Half of them are generated randomly and the other half starting from the most frequent values in the database. We use a Michigan approach in which each individual (chromosomes) encodes a single rule. The format of the rules we are going to discover is: IF Variable1 = Value1 (AND Variable2 = Value2 …) THEN VariableX =ValueX

Where:

- Variable1, Variable2, VariableX: Are the database's field names. (P0..P6,AVEPH,AVEPNOH,CASETIME0..CASETIME6,CASESCORE0.. CASESCORE6,ACCESSTIME0.. ACCESSTIME6,C0..C37,MCQ0.. MCQ6).
- Value1, Value2, ValueX: Are the possible values of the previous database fields (VERYLOW,LOW,MEDIUM,HIGH,VERYHIGH,SUCCESSFIRST, SUCCESSSECOND, SUCCESSTHIRD, SUCCESSFOUR).

We use value encoding in which a rule is a linear string of conditions, where each condition is a variable-value pair. The size of the rules is dynamic depend of the number of elements in antecedent and the last element always represents the consequent.

4.2 Evaluation

Evaluation consists of calculating the fitness of the current rules and keeping with the best ones. To calculate the fitness we count the precision of the rule, the number of patterns in the database that fulfill both antecedent and consequent and do not fulfill both antecedent and consequent. That is, we obtain very strong association rules [2] that fulfill $[A=a] \rightarrow [C=c]$ and $[C\neq c] \rightarrow [A\neq a]$. So a rule is very strong if the previous two rules are strong, that is, both rules have greater support and confidence than a minimum values set by the user (0.5-1). Our formula detects both statistical negative dependence and independence between antecedent and consequent.

4.3 Selection

The selection chooses rules from the population to be parents to crossover or mutate. We use rank-based selection that first ranks the population and then every rule receives fitness from its ranking. The worst will have fitness 1, second worst 2, etc. and the best will have fitness N (number of rules in population). Parents are selected according to their fitness. With this method all the rules have a chance to be selected. We also use an elitism method, which first copies a few best rules to new population. Elitism increases performance of the genetic algorithm, as it prevents losing the best found solution.

4.4 Reproduction

Reproduction consists of creating new rules, mutating and crossing current rules (rules obtained in the previous evolution step). The crossover and mutation probability is set by the user. A higher crossover rate (50-95%) and a lower mutation rate (0.5-2%) are recommended. Additionally it is good to leave some part of population survive up to next generation. Mutation consists of the creation of a new rule, starting from an older rule where we change a variable or value. We randomly mutate a variable or values in the consequent or antecedent. Crossover consists of making two new rules, starting from the crossing of two existent rules. In crossing the antecedent of a rule is joined to the consequent of the other rule in order to form a new rule and vice versa (the consequent of the first rule is joined to an antecedent of the second). So it is necessary to have two rules to do the crossover.

4.5 Finalization

Finalization is the number of steps or generations that will be applied to the genetic process. The user chooses this value (10-500 steps). We could also have chosen to stop when a certain number of rules are obtained.

5 Rules discovered and Conclusions

We have carried out different execution proofs with the described genetic algorithm and the data obtained from the medical Web-based Hypermedia Course, to discover association rules. We have applied the algorithm to the whole data, only to the medical students and only to the other users. For each case, we have obtained different rules both in the content and in number and fit. For example, one of the rules obtained when using 100 initial rules and 100 steps is: IF CASESCORE2=SUCCESSFIRT AND CASESCORE4=SUCCESSFIRT THEN CASESCORE1=SUCCESSFIRST (with support = 0.73 and confidence=1). This can be interpreted as if a user gets the answer for case 2 and case 4 correct, he/she is likely to do well in case 1. The support of a rule gives the importance of a rule and the

confidence of a rule gives its predictability power. All the rules discovered are showed to the teacher in order he can obtain conclusion about the course functionality. The teacher has to analyze them and he has to decide what are the best modifications that can improve the performance of the course. Summarizing the main conclusions that we obtained starting from the discovered rules are: We obtained expected relations, for example: between CASESCORE and CONCEPT, and between MCQ and CONCEPT, due to the fact that the questions are about the concepts. We obtained useful relations, for example: between CASESCORE and MCQ. This could be because the questions are the same ones, or they refer to the same concepts, or they have equal difficulty. We obtained strange relations, for example between AVEP_AH and P, it is probable that this relation takes place by chance. And we didn't find any other relation, for example with ACCESSTIME or CASETIME. This could be because user access times were completely random and they did not determine any other variable as it might be expected.

Having tested our genetic algorithms on the Web-based Hpermedia Course as described above, and shown that they can produce potentially useful results, we plan to apply them to this and other adaptive hypermedia courses to test how they can be used to improve the adaptive features. The preliminary results reported in this paper are promising and they show that our genetic algorithm is a good alternative for extracting a small set of comprehensible rules, which is important in the context of data mining. We are now developing several courses with AHA system [1] and we expect to obtain interesting rules to improve the course adaptation. We have chosen AHA system because it is a generic model of hypermedia adaptive system and it has a high degree of adaptation. We are also developing a more sophisticated evolutionary algorithm with genetic programming [3], to obtain more complex and interesting rules.

References

1. De Bra P., Wu H., Aerst A., Houben G.: Adaptation Control in Adaptive Hypermedia Systems. International Conference on Adaptive Hypermedia. (2000).
2. Delgado M., Sánchez D., Martín-Bautista M.J., Vila A.: Mining association rules with improved semantics in medical databases. Artificial Intelligence in Medicine. 21. (2001).
3. Freitas, A. A.: A survey of evolutionary algorithms for data minig and knowledge discovery. To appear in: Advances in Evolutionary Computation. A Springer-Verlag, (2002).
4. Hipp J., Güntzer U., Nakhaeizaeh G.: Algorithms for Association Rule Mining – A General Survey and Comparison. SIGKDD Explorations. (2.000).
5. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
6. Ng. M, Hall W., Maier P., Armstrong R.: History-based Link Annotation for Self-Exploratory Learning in Web-based Hypermedia. 3rd workshop on Adaptive Hypertext and Hypermedia. Arhus, Denmark. (2001).
7. Osmar R. Zaïene.: Web Usage Mining for a Better Web-Based Learning Environment. Technical Report. (2001).
8. Witten I., Frank E.: Data Mining. Practical Machina Learning Tools an Techniques with Java implementations. Morgan Kaufmann Publishers (1999).