Global Journal of Computational Intelligence Research.
ISSN 2249-0000 Volume 1, Number 2 (2011), pp. 93-101

http://www.ripublication.com/gjcir.htm

Applying Association Mining based Evolutionary Algorithm to Augment Adaptive Educational Systems in E-Learning

R. Kavitha

Department of MCA,
Aloysius Institute of Management and Information Technology (AIMIT),
Mangalore, Karnataka, India.
E-mail: kavitha.rajamanii@gmail.com

Abstract

Due to the rapid advancements in this technical era, student depends on on-line contents for their learning activity. In this paper, I describe how to discover interesting associations from student's usage information to progress adaptive e-learning courses. The course should adapt both the presentation and navigation ways that depend on the knowledge skill of each student. Data mining methods are used to provide feedback to courseware authors. The discovered information is presented in the form of prediction rules because they are comprehensible and show only critical associations involved. They can be used to improve the courseware. I propose to use an objective technique for rule discovery. It will facilitate the knowledge discovery process to novice users in data mining.

Keywords: *Evolutionary Algorithm, Adaptive e-learning, Association Mining.*

Introduction

Due to the advancement in the technical era, there is a rapid growth of applying web-based technology in e-learning systems. Different techniques have been applied to the systems to improve students' learning. Web-based Educational Hypermedia Adaptive Systems [1] allow adapting the teaching to each individual student. The system logs information about the users' interaction with the course. Evaluators use this for student assessment. The proposed methodology uses the system usage information that will allow the teacher to improve the course. DBMiner and Weka are popular in discovering rules. These tools may be difficult for a novice user who does not familiar

with data mining. They do not contain features that are specific to Adaptive Systems for Web-based Education (ASWEs) [1]. To solve the problems a specific tool has been developed with EPRules (Education Prediction Rules), to simplify the process of discovering prediction rules [3]. AHA [4] is used to make courses that adapt both the presentation and the navigation depending on the knowledge skill that each particular student has. The objective is to carry out the suitable modifications in the course to improve it. Next section describes the application of rule mining techniques in e-learning systems, then the process of discovering information implemented in EPRules, and the proposed grammar-based programming for rule discovery. Then, EPRules tool and the performed tests are described. Final section has the main conclusions and future work.

Data Mining in E-Learning Systems

Data mining is a knowledge discovery process to find potentially useful and non-trivial patterns from large repositories of data. The methods in mining are: Association, Sequence analysis, Clustering and classification. In their pioneering article, Za¨iane [2] proposes to build an agent that could recommend on-line learning activities based on learner's access history. In Wang [5], describes a set of tools for analyzing browsing log files based on association mining and collaborative filtering. Yu et. al.[6], use data mining technology to find incorrect student behavior. Minaei-Bidgoli [7] introduce an approach for predicting student performance. All the approaches use the visited pages as input to the search, and so the discovered information describes associations between pages. The proposed method searches for associations between concepts and chapter units of courses. It uses evolutionary algorithms as association rule mining method for discovering associations in student's usage data [8]. Four main steps in the methodology are:

- **Structuring of the course:** The teacher organize the course providing information of the domain model, and the interface module. On completion, contents may be published.
- **Implementation of the course:** Students execute the course using a web navigator in a transparent way; the usage information is stored into the log file.
- **Rule discovery:** The evaluator can apply the evolutionary algorithms to discover associations among the data by using EPRules tool to facilitate this task.
- **Enhancing the course:** The evaluator can use the discovered associations to improve the performance of the course by using an authoring tool again.

Students' Usage Data

The course is developed using AHA [4] since, apart from being a generic adaptive hypermedia system, it captures all the user's usage information, and its source code is available in order to increase its adaptation power in education. To adapt or personalize the system to each particular student depending on his knowledge skill, change the following models:

***Domain model*:** A course consists of several chapters with several concepts, and the questions are divided in three levels of difficulty (high, medium or low).

***Learner model*:** The student's knowledge skill for each concept can be only one of these values: 0 (not yet read), 10 (low), 50 (medium) and 100 (high).

***Customizing engine*:** The system presents the students only the concepts with their initial knowledge level. When they completed all the concepts they have to do a final test to evaluate their knowledge about the chapter. If it is medium or high level, they can go to a higher level. If they are in the highest level already they can go to the next chapter. For each chapter everything starts again.

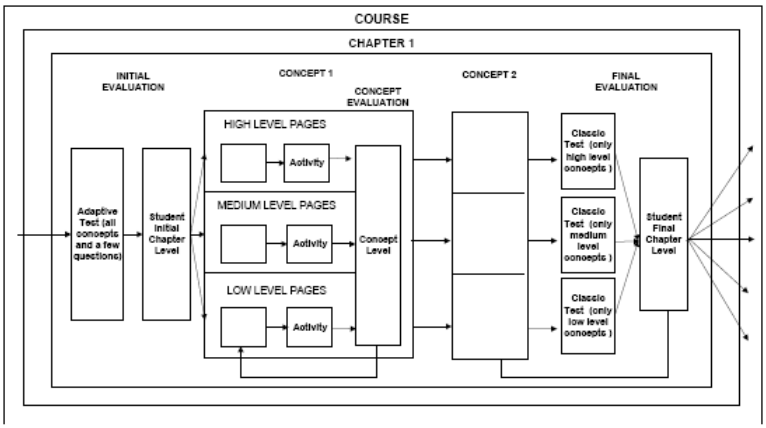

Figure 1: Modified AHA.

The chapter of the course is with two different difficulty levels (beginner and skilled). Each version has a different concept explanation that is suited to the respective knowledge skills. The AHA System stores the usage information in two web log files about user navigation and user knowledge. In addition, another log file (test file) is to store the scores of the questions. The specific usage is:

- *Times*: Created from the log files and has Web pages and the time in which the student has accessed to them.
- *Levels*: Created from the model file and has the knowledge level (high, medium, low) that the student has.
- *Success:* Created from the test file and it contains information about the success or failure of the students.

The data preprocessing tasks involved are: attribute selection, data cleaning, discretization of continuous attributes and data integration are to be carried out.

Knowledge Discovery Process

Knowledge discovery that consists of following steps:

- *Preprocessing*: It consists of the data gathering, data cleaning, discretization of continuous data, attribute selection, data integration, etc.
- *Data mining*: It consists of the classification, regression, clustering, rule discovery, etc.
- *Post processing*: It consists of the interpretation, evaluation of the obtained results and the utilization of the discovered knowledge.

The specific process of knowledge discovery is as follows.

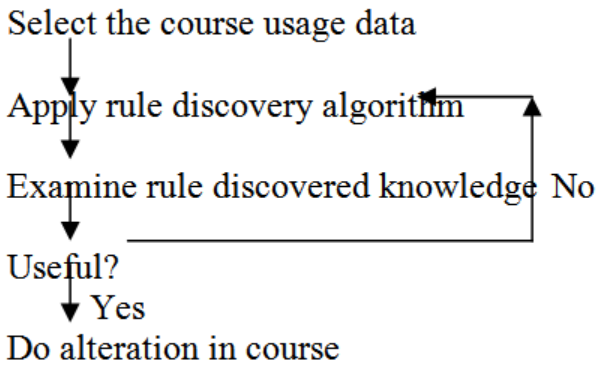

Figure 2: Specific Knowledge Discovery Process.

All this process can be carried out by the evaluator or author of the course, using the EPRules tool [8]. The rule discovery process begins with the selection of the database where the pre-processed usage data of the course to be used are stored. Then the knowledge discovery algorithms to be applied must be selected with their specific parameters and both the objective and subjective restrictions that we want the discovered rules to fulfill. After concluding the algorithm execution, the group of discovered prediction rules is displayed: the elements of the rule antecedent and consequent as well as the evaluation measures of each rule are shown. Then it is decided which of the discovered rules are interesting to use to take decisions on possible modifications. If the rules are not considered sufficiently interesting the algorithm is applied again, with different parameters and restrictions, to discover a more interesting group of rules.

Rule Discovery with Evolutionary Algorithm

IF-THEN rules are one of the most popular forms of knowledge representation. The different types of rules are: decision rules, association rules, classification rules, prediction rules, causal rules, optimization rules, etc. Prediction rules are used here and Table 4.1 1 shows the precise format of the rules in EBNF (Extended Backus Naur Form).

The objective is to predict an objective attribute depending on the values of another group of attributes. They are popular since they represent discovered

knowledge at a high level of abstraction and it can be used directly in the decision making process. The rule discovery is carried out by (GBGP) to assess the quality of the rules. Evolutionary Algorithm based Programming is represented by trees derived from a grammar defined by the user to specify the solutions of the problem. Though plenty of metrics available to evaluate the quality of the rules, not a single one clearly surpasses the others in all application domains. For this reason this problem has been set out like a multi-objective optimization problem . Several functions need to perform the optimization at the same time. It can be dealt by evolutionary algorithms: the first one uses the aggregation function, while the second one uses the Pareto Front concept. In the Pareto Front there is a vector of objectives to optimize within each individual, and the purpose is to make the solution for the individual converge towards the group of the best solutions.

The evolutionary algorithm consists of the steps: Initialization, Evaluation, Selection and Reproduction repeated until finalization condition.

Table 1: If-Then Rule Format in EBNF

<rule> ::= IF<antecedent>THEN<consequent>

<antecedent>::=<antecedent>AND<condition> | <condition>

<consequent> ::= <condition>

<condition> ::= <level-attribute>=<level-value>|

<time-attribute> = <time-value>|

<success-attribute> = <success-value>

<level-attribute> ::= LEVEL.Name of a valid level attribute

<time-attribute> ::= TIME.Name of a valid time attribute

<success-attribute>::=SUCCESS.Name of valid success attribute

<level-value> ::= BEGINNER | NORMAL | PROFESSIONAL

<time-value> ::=HIGH | MEDIUM | LOW

<success-value>::=YES | NO is fulfilled.

- Initialization consists of generating a group of initial rules specified by the user. The general format of the rules going to discover in Backus Naur Form (BNF) is shown in Table 4.1 1.
- Evaluation consists of calculating the fitness of the current rules. It is made up of a 3-valued vector to measure three aspects as comprehensible, interestingness and accuracy. The metrics are certainty factor measure, interestingness measure and simplicity measure.

- Selection chooses rules from the population to crossover by rank based selection. The worst will have fitness 1, and then worst 2, etc. and best will have fitness N. Parents are selected according to their fitness.
- Reproduction consists of creating new rules, mutating and crossing over current rules. Mutation is creation of a new rule, starting from an older rule. In recombination the antecedent of a rule is joined to the consequent of another rule in order to form a new rule and vice versa.
- Finalization is the number of steps or generations to be applied to the genetic process.

EPRULES Tool

The EPRules tool [3] is a visual tool to discover prediction rules. It lets the user add new rule discovering algorithms and new rule evaluation measures. The graphic interface consists of four main windows:

- *Data input*: In this window you can open an existing database or create a new one and add new students to it, discretize the time attribute by assigning three values as HIGH, MEDIUM and LOW.
- *Data view*: Every student's pre-processed usage data as access times; correct answers and knowledge skills by students for the different web pages can be visualized.
- *Rule discovery*: The implemented algorithms are: for decision tree - ID3, for association rule discovery - Apriori, for induction rules - grammar based genetic programming algorithm with or without multi-objective optimization (Pareto). Select the algorithm, parameters and subjective restrictions that rules to fulfill.
- *Results*: This window appears automatically after finishing the algorithm execution and lets us visualize all the discovered prediction rules.

Experimental Results

To carry out the tests use the log information of 25 students. The course was developed with the AHA System [4]. The objective is to compare the number of discovered rules in each case and the quality of them based on the previously set out measures about accuracy, interestingness and comprehensibility. Three different tests have been carried out: first using all data, then only the frequent data (those with a support higher than 0.5) and finally, the range data (those with a support higher than 0.2 and lower than 0.9).

The obtained results are shown in Table 2,3,4 Evolutionary algorithms generate a lower number of rules but with higher interest than classic algorithms, making them more suitable to be used on-line. Classic algorithms, and specially Apriori, produce very exact rules, but fail when generating rules with a higher interest.

Table 2: Number of Discovered Rules

Algorithm	All	Range	Frequent
ID3	474	131	89
Apriori	5960	491	70
AE-GBGP	198	162	51

Table 3: Percentage of Exact Rules

Algorithm	All	Range	Frequent
ID3	46,0	51,9	60,3
Apriori	84,3	90,0	93,0
AE-GBGP	76,5	86,1	96,3

Table 4: Percentage of Interesting Rules

Algorithm	All	Range	Frequent
ID3	1,5	7,6	15,6
Apriori	3,6	7,9	53,1
AE-GBGP	21,9	60,4	76,6

Description of the discovered information

The objective is to discover a group of interesting rules and to present them to the teacher to take decisions about how to improve the course. The discovered rules express the following relationships:

IF Level | Time | Success AND ...

THEN Level | Time | Success

where Level, Time and Success are expressions referring to users' attained knowledge state (BEGINNER, NORMAL, PROFESSIONAL), the reading time for pages (HIGH, MEDIUM, LOW), and to students' successes and failures in the test (YES, NO). Taking the discovered rules as a basis the teacher can decide which of the expressed associations are desirable or undesirable, and what can be done to reinforce or decline them.

Using these discovered associations a teacher can make decisions to augment the associations or on the contrary to eliminate the relationship. The meaning and the possible use of a discovered rule is as follows:

IF LEVEL . Advanced I/O - Record Locking - high = PROFESSIONAL THEN

LEVEL . IPC – Semaphore - medium = PROFESSIONAL

(Interest=0.57, Factor Certainty=0.75, Simplicity=1)

This rule shows that the knowledge level obtained in the evaluation activities of

the two concepts have been simultaneously very high (PROFESSIONAL). This indicates that the concepts (Record Locking, with HIGH difficulty level in the Advanced I/O chapter, and Semaphore, with MEDIUM difficulty level in the IPC chapter) may be related to each other. The evaluator should check the presentation content of both concepts and find what the reason of the relationship is and join both concepts in the same chapter, setting them to the same level of difficulty, correcting the rules that assign levels. If the levels refer to initial or final test, it can be concluded that the chapters are related and can join the chapters, or put them one after the other.

Conclusions and Future Work

In this paper a visual tool is introduced to discover knowledge in the form of prediction rules to help evaluators to improve adaptive on line e- courses. Particularly, Grammar-based genetic programming with multi-objective techniques is proposed. The quality of the results, depending on the number of rules obtained and their interestingness, accuracy and comprehensible factor of the rules, is higher than for a number of classic algorithms that use only one measure or a composition of some of them to evaluate the rules. A specific tool, named EPRules, has been used to simplify the process of knowledge discovery. This tool carries out the pre-processing of usage data in on line courses, the selection of restrictions on the type of information to be discovered as well as the application of data mining algorithms to extract the rules and to show them.

Complete automation of the knowledge discovery process in ASWEs, so that the discovered rules can be applied directly on the course, without manual intervention by the evaluator or author of the course, except for accepting or rejecting the changes proposed by the rules.

In future, parameter-free rule mining algorithms can be used so there is no need to pose to the courseware authors for the specific values of the algorithm's parameters, and they don't need to understand what the role of these parameters is in the data mining process.

References

[1] Brusilovsky, P., Adaptative educational systems on the world-wideweb: A review of available technologies. Int. Conf. on Intelligent Tutoring System, 1998.

[2] Zaiane, O.R., Web usage mining for a better web-based learning environment. Technical report, 2001.

[3] Romero, C., Ventura, S. & de Bra, P., Discovering prediction rules in AHA courses. 9th Int.Conf. on User Modeling, Johnstown, PA, USA, pp. 25–34, 2003.

[4] de Bra, P. & Ruiter, J., AHA adaptive hipermedia for all. Proc. of the WebNet Conference, pp. 262–268, 2001.

[5] Wang, F., On analysis and modeling of student browsing behavior in web-based asynchronous learning environments. Int. Conf. on Web-based Learning, HongKong, pp.69–80, 2002.
[6] Yu, P.,Own, C. & Lin, L., On learning behavior analysis of web based interactive environment. Int. Conf. ICCEE, Oslo, 2001.
[7] Minaei-Bidgoli, B. & Punch, W., Predicting student performance: an application of data mining methods with the educational web-based system lon-apa. IEEE Frontiers in Education, p. 1–6, 2003.
[8] Romero, C., Ventura, S. & de Bra, P., Knowledge discovery with genetic programming for providing feedback to courseware author. User Modeling and User-dapted Interaction: The Journal of Personalization Research, 14(5), pp. 425–465, 2005.

Author Biography

R. Kavitha is from Tamilnadu, India. She did MCA (Bharathiar University, Tamilnadu), MPhil., in Computer Science (Bharathidasan University, Tamilnadu). Also, she got PGDHET, a degree in Higher Education Technology from Bharathidasan University which is meant exclusively for academicians working in colleges. She is a rank holder in under graduate.

She is having 9 years of teaching experience in colleges. She is currently working as Assistant Professor in Department of MCA, Aloysius Institute of Management and Information Technology (AIMIT), Mangalore, Karnataka, India. Her research area of interest is educational data mining. She presented 4 papers in National Conferences and 2 papers in International Conferences. Now, she is involved in preparing an e-learning content for the subject of Data Structures.