Expert Systems

Article

DOI: 10.1111/j.1468-0394.2012.00627.x

Association rule mining using genetic programming to provide feedback to instructors from multiple-choice quiz data

Cristóbal Romero, Amelia Zafra, Jose María Luna and Sebastián Ventura

Department of Computer Sciences and Numerical Analysis, University of Córdoba,14071 Córdoba, Spain
Email:cromero@uco.es

Abstract: *This paper proposes the application of association rule mining to improve quizzes and courses. First, the paper shows how to preprocess quiz data and how to create several data matrices for use in the process of knowledge discovery. Next, the proposed algorithm that uses grammar-guided genetic programming is described and compared with both classical and recent soft-computing association rule mining algorithms. Then, different objective and subjective rule evaluation measures are used to select the most interesting and useful rules. Experiments have been carried out by using real data of university students enrolled on an artificial intelligence practice Moodle's course on the CLIPS programming language. Some examples of these rules are shown, together with the feedback that they provide to instructors making decisions about how to improve quizzes and courses. Finally, starting with the information provided by the rules, the CLIPS quiz and course have been updated. These innovations have been evaluated by comparing the performance achieved by students before and after applying the changes using one control group and two different experimental groups.*

Keywords: Educational data mining, computer-based testing, association rule mining, grammar-guided genetic programming

1. Introduction

Computer-based testing (CBT), also known as computer-based assessment (CBA) or quiz systems, is one of the most widely used and well-developed tools in education (Brusilovsky & Miller, 1998) for administering tests in which the responses are electronically recorded, assessed, or both. On one hand, CBT may be a stand-alone system such as QuestionMark[1], Webassesor[2], or HotPotatoes[3], or a part of a Virtual Learning Environment (VLE) or Learning Management System (LMS) such as Blackboard[4] or Moodle[5]. On the other hand, there are different types of CBT but the most popular is multiple-choice questions (MCQs) in which students are asked to select the best possible answer (or answers) from the choices provided on a list. They have the following advantages: rapid feedback, automatic evaluation, perceived objectivity, reuse of questions as required, easily computed statistical analysis of test results, and the possibility of generating data that can provide a better understanding of their learning process (Kuechler & Simkin, 2003). In fact, MCQs provide a large amount of data about student interaction with the test such as: students' answers, scores obtained for each question, calculated final score, execution times, etc. (Romero *et al.*, 2009). Therefore, they offer potentially useful information that could be very valuable for providing feedback to instructors. The problem is that discovering interesting information to improve the quiz by hand or using only statistical information may be a difficult task, because these systems can generate a great amount of information about student interaction with the test (Romero *et al.*, 2008). Although there are conventional quiz report analysis tools that provide statistics information about students and items/questions performance, they don't give suggestions or feedback regarding how to improve student learning performance, for example, by modifying the quiz or the own course evaluated by the quiz.

One solution to this problem is to use educational data mining (EDM), a new research area specifically oriented to analysis of these types of data (Romero & Ventura, 2010). EDM can be defined as the application of different data mining (DM) techniques to educational data. Of all the DM techniques, association rule mining (ARM) is one of the most popular. Its objective is to discover relationships among attributes in datasets (Ceglar & Roddick, 2006). ARM algorithms have been applied to a wide range of educational problems and tasks (Romero & Ventura, 2010) such as making recommendations for students and teachers, student modelling, predicting student performance, etc. Little attention has been paid, however, to how to apply the ARM techniques to analyse students' questionnaire data. On the one hand, there are only a few examples of the use of classical ARM based on the Apriori algorithm (Agrawal & Srikant, 1994) for quiz data. For example, it is used in questionnaire data for mining open answers in order to extract

[1] http://www.questionmark.com/

[2] http://www.webassessor.com/

[3] http://hotpot.uvic.ca/

[4] http://www.blackboard.com/

[5] http://moodle.org/

characteristics of individual analysis targets as well as the relationships among those characteristics (Yamanishi & Li, 2001), and in real online exams (in the form of a multiple-choice test) in order to discover how well an elaborated quiz was designed or tailored towards the individual needs of the students (Pechenizkiy *et al.*, 2008). On the other hand, some fuzzy rule association algorithms have also been applied to quiz data. For example, they have been used in a learning diagnosis approach for providing students with personalized learning suggestions by analysing their test results (Chu *et al.*, 2006), and in closed questionnaire data in order to identify the various data types that may appear in a questionnaire and to discover rule patterns (Chen & Weng, 2009).

Our work differs from these previous studies in two ways. First, although evolutionary algorithms (EAs) have been applied successfully in many areas, they have not been applied yet in any work about ARM in quiz data. In this paper, an EA that uses a grammar-guided genetic programming (G3P) approach is applied to quiz data, providing a good performance and rule expression capacities. Second, the goal of this work is to discover interesting relationships to aid the instructor (courseware author, quiz ware author, etc.) in decision making about how to improve both the quiz and the corresponding course that contains the concepts evaluated by the quiz. The present study not only shows the discovered rules as other works do but goes one step further in that the results obtained are applied in a real course where the improvements achieved are evaluated. Thus, first, by the rules the information discovered is applied to introduce a list of updates in a specific quiz and course. Then, it is evaluated whether the changes introduced really do improve the results achieved by the students. To this end, different groups of students who use the quiz and course before and after application of these updates are compared.

The rest of this paper is organized as follows. Section 2 introduces the DM process proposed for improving quizzes and courses. Section 3 describes the experiments and results obtained when a novel grammar-based algorithm for mining association rules is compared versus other classic and soft-computing algorithms. This section also shows the most interesting types of relationships for making decisions and some rules discovered by our algorithm. Section 4 evaluates the list of updates carried out in a Moodle quiz and course starting with feedback provided by the rules. Finally, some conclusions are drawn and future work outlined.

2. Proposed DM process for providing feedback

The task of designing, developing, and evaluating a quiz may be arduous and laborious, because instructors or authors must make important decisions such as: how many questions must a quiz contain? What are the most appropriate questions for evaluating each concept of the course? How much time should students be allowed in order to do the quiz? What are the most discriminatory questions? What behaviour may explain failing or passing the quiz? Are the current contents of the concepts evaluated by the questions appropriate? Owing to all these issues, it is difficult to determine the most suitable quiz for evaluating a specific course. In fact, it is very likely that different authors would propose different quizzes for the same course. DM can be used for a deeper evaluation of students' interaction with the quiz in order to help to answer some of these issues. In this paper, ARM is proposed for providing feedback to instructors and educational designers (see Figure 1).

The proposed DM process can be viewed as a quiz and course evaluation cycle (see Figure 1) in which the quiz's own usage data are used for discovering updates in the quiz and also in the course. In this process, there are two main actors/users: students and instructors. On the one hand, students use courses for learning concepts and then do quizzes evaluating their knowledge of these concepts. On the other hand, instructors pre-process quiz data and transform them to suitable data files for ARM. The rules discovered by ARM algorithms are then post-processed in order to detect the most interesting rules for helping in decision making about how to update and improve quizzes and courses. In the following sections, each step of the DM process is described in more detail.

2.1. Pre-processing quiz data

Quiz data can be gathered from different sources (information provided by online quiz report analysis tool, information provided by paper-and-pencil, and additional other information provided by the own quiz author or instructor, etc.). So, different matrixes or tables can be obtained for doing DM such as: a score matrix provided directly by the used quiz system, or a relationships matrix provided by instructors, and a knowledge matrix automatically generated from the two previous matrices (see Figure 2).

- **Score matrix**. This is a traditional student-rating data matrix in which each row represents a student and each column represents an item/question (see top left of Figure 2). A_{ij} represents the answer or score of $Item_j$ rating obtained by the $Student_i$. Finally, two new columns were added: the time used (T_i) and the final score (S_i) of each student.
- **Relationships matrix.** This matrix is similar to the Q-matrix (Barnes, 2005), which shows relationships between items/questions, the degree of association between them, and the concepts evaluated by the quiz whereby one item can evaluate or be related to one or several concepts (see

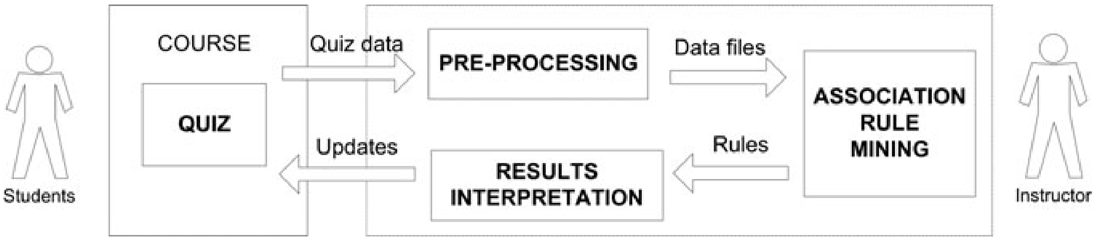


Figure 1: *Data mining process for providing feedback.*

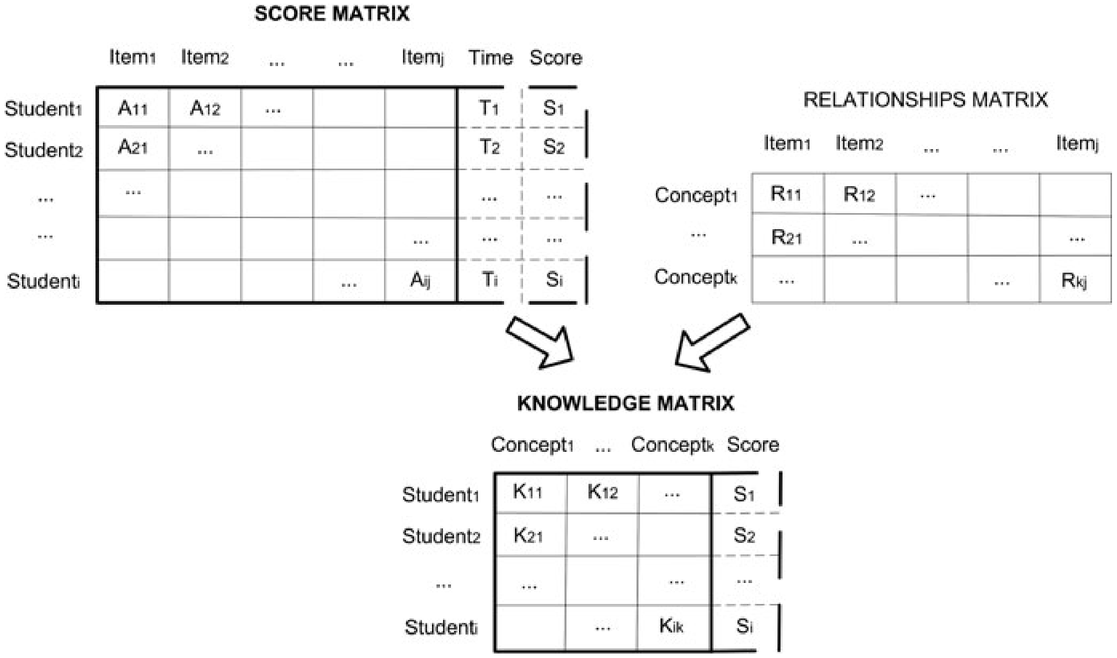


Figure 2: *Score, relationship, and knowledge matrixes.*

the top of the right-hand-side of Figure 2). R_{kj} represents the probability that $Item_i$ has of being related to $Concept_k$.

- **Knowledge matrix**. This matrix shows the level of knowledge that students have about each concept evaluated by the quiz. It has been created automatically from the two previous matrices (see bottom of Figure 2). K_{ik} represents the addition of the answers to each one of the items related to $Concept_k$ by each $Student_i$, that is$K[i][k] = \sum_j A[i_i][j] \times R[k][j]$. Finally, a new column with the final score obtained by each student (S_i) has also been added.

Then, all the previous numerical data have been transformed into categorical values that are more user friendly than precise magnitudes and ranges because they provide a much more comprehensible view of the data (Dougherty *et al.*, 1995). Finally, matrices have been saved into different data files for applying ARM.

2.2. ARM using quiz data

The problem of discovering association rules can be defined following the original definition (Agrawal & Srikant, 1994) as: Let $I = \{i_1; i_2; i_3;\ldots; i_n\}$ be the set of items and D be the set of all transactions in the relation, an association rule is an IF-THEN statement or implication of the form $A \rightarrow C$ where $A \subset I$, $C \subset I$, and $A \cap C = \emptyset$, that is both the antecedent A and the consequent C are item sets having no items in common. The meaning of an association rule is that if the antecedent A is satisfied, then it is highly probable that the consequent C will be also satisfied. In an association rule, both the antecedent and consequent must satisfy a user-specified minimum support and a minimum confidence at the same time (Ceglar & Roddick, 2006). The support of the rule is the proportion of the number of transactions T including A and C in D. In a formal way, the support is defined mathematically as follows.

$$Support(A \rightarrow C) = \frac{|\{A \cup C \subseteq T, T \in D\}|}{|D|}$$

The confidence of an association rule is the proportion of transactions T in D containing A that also contain C, that is the proportion of the number of transactions T that include A and C among all the transactions that include A. The confidence is formally defined as follows.

$$Confidence(A \rightarrow C) = \frac{|\{A \cup C \subseteq T, T \in D\}|}{|\{A \subseteq T, T \in D\}|}$$

A wide variety of ARM algorithms has already been proposed (Romero *et al.*, 2011). On the one hand, there are a great number of classical ARM algorithms that are mainly variations or improvements of the Apriori algorithm (Agrawal & Srikant, 1994), which is the first, simplest, and most common ARM algorithm, such as FP-Growth (Han *et al.*, 1999), ECLAT (Zaki, 2000), etc. In the process of extracting association rules, these classical ARM algorithms follow an exhaustive search methodology, first mining frequent patterns and then, they use these patterns for discovering reliable association rules. The mining process is hindered by these two steps, requiring a high computational time and large amounts of memory. On the other hand, and in order to face up to the exhaustive search approaches' computational and memory requirements, many researchers focused the ARM problem under an EA perspective and, specially, using genetic algorithms and fuzzy logic such as EARMGA (Yan *et al.*, 2009) and GAFuzzyApriori (Hong *et al.*, 2006). A novel ARM algorithm based on G3P (Mckay *et al.*, 2010), called G3PARM (Grammar-Guided Genetic Programming Association Rule Mining), was recently developed by our research group (Luna *et al.*, 2011), obtaining very promising results. Therefore, an adaptation of G3PARM to the problem described in this paper is carried out. This algorithm is based on genetic programming (GP), an EA methodology that represents the individuals in a tree form. This algorithm follows a Michigan approach (each individual encodes a single rule), and uses an external elite population. This means that two different groups of individuals/rules are used: one represents the current population in each generation, and the other represents the elite population with the best individuals/rules, those that exceed a support and a confidence threshold. In each generation of the EA, a set of individuals is selected (via a binary tournament) as parents to obtain new individuals by applying genetic operators (crossover and mutation). For the sake of discovering better individuals, the crossover genetic operator swaps the lower support condition within a parent with the higher support condition within the other parent. In such a way, it is highly probable to obtain new individuals with a better support. On the other hand, and focusing on the mutation genetic operator, the lower support condition within each parent is changed. This genetic operator provides two possibilities of changing a condition: (1) to obtain new complete condition or (2) to obtain a new value for the attribute of the condition. Once a new population is obtained by crossover and mutation, the next step is to update the elite population with the best individuals of the current population, those that exceed a support and confidence threshold. This process is repeated until a maximum number of generations have been reached, returning to the user the set of rules kept in the elite population.

3. Experiments and results

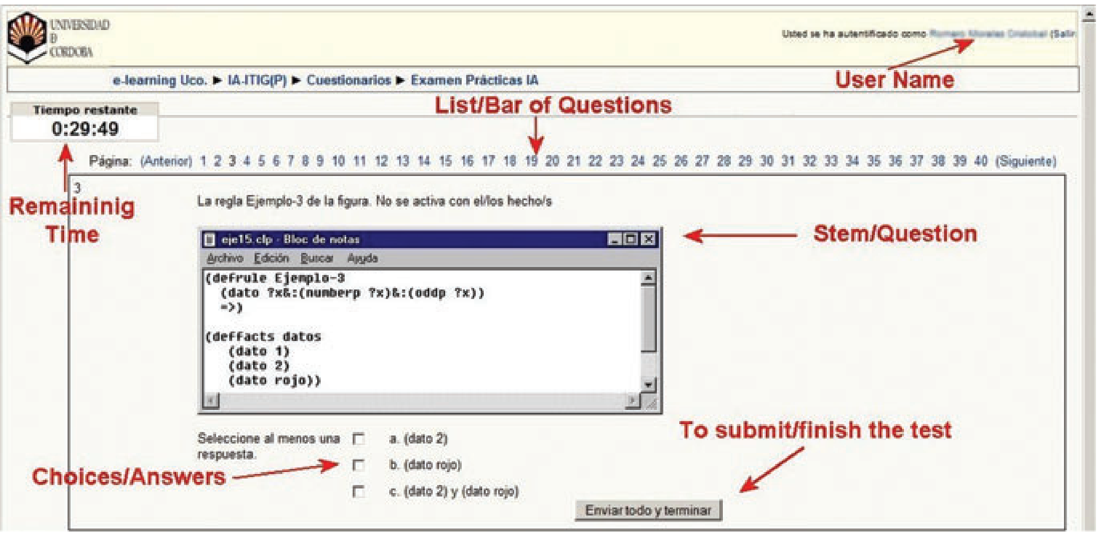
3.2. Data

Moodle LMS has been used in this work because it is used in our university and also it provides a powerful quiz module tool (Romero *et al.*, 2008). A Moodle multiple-choice quiz was as a final exam on CLIPS programming language. All the concepts evaluated by the quiz were explained in an artificial intelligence practice course during the 2nd year of the computer science degree at the University of Cordoba in Spain. This course had traditional in-class lectures and also used Moodle to provide learning content, additional resources, and online activities such as a multiple-choice quiz. Students took the quiz at the end of the course in a computer classroom and three instructors supervised the execution of the exam. This online quiz consisted of 40 items/questions with three possible options/answers, only one of which was correct (see Figure 3). The time limit or maximum total time for doing the test was set at 35 min although students could finish their quiz ahead of time. Only one question was displayed per page and shuffle questions were used, that is, questions and answers were shown to each student randomly. Students had only one attempt at the test but they had the possibility of answering the questions in a flexible order. In this way, they could revisit earlier answers and revise them. Finally, it is important to mention that Moodle provides some security features for quizzes in official exams. For example, the quiz can be shown in a separate full-screen window that pops up and covers all other windows and does not have any navigation controls. Moreover, students cannot use functions like copy, paste, and print. Students can be required to enter a password before they are permitted to take the quiz, and access to a quiz can be restricted to particular subnets on a network to ensure that only students in a certain room are able to access the quiz. The total number of students who take the exam during the academic year 2008–2009 was 104 that will be used for discovering association rules.

3.3. Pre-processing settings

In our case, starting with the information provided by the instructor and the Moodle quiz report analysis tool about this exam, the three previous described matrices were obtained.

- **Score matrix**. This matrix was directly calculated by Moodle and provided each one of the 40 answers given by the 104 students. In our case, the answer or score of one particular item can be 0 for an incorrect answer or 1 for a correct answer, the total time taken by each student is a real value between 0 and 35 min. The final score obtained by each student is a real value automatically calculated by adding all the individual answers for each question/item and then normalizing between 0 and 10 points.
- **Relationships matrix.** This matrix was designed by hand by the instructor. Thus, the instructor first specified the list of concepts that compose the domain taught by the course and that are evaluated by the quiz. In our case, the specific domain is the CLIPS language that consists of the following 13 concepts that correspond to chapters of our course that are explained to the students in the following order: (1) *rule-based systems*, (2) *CLIPS basic aspects,* (3) *ordered facts*, (4) *non-ordered facts*, (5) *initial facts*, (6)


UNIVERSIDAD
D
CORDOBA
Usted se ha autentificado como [illegible] (Salir
e-learning Uco. ▶ IA-ITIG(P) ▶ Cuestionarios ▶ Examen Prácticas IA
User Name
List/Bar of Questions
Tiempo restante
0:29:49
Página: (Anterior) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 (Siguiente)
Remaininig Time
Stem/Question
eje15.clp - Bloc de notas
Archivo Edición Buscar Ayuda
To submit/finish the test
Choices/Answers
Enviar todo y terminar

3

La regla Ejemplo-3 de la figura. No se activa con el/los hecho/s

```
(defrule Ejemplo-3
  (dato ?x&:(numberp ?x)&:(oddp ?x))
  =>)

(deffacts datos
   (dato 1)
   (dato 2)
   (dato rojo))
```

Seleccione al menos una respuesta.

□ a. (dato 2)
□ b. (dato rojo)
□ c. (dato 2) y (dato rojo)

Figure 3: *Moodle user interface of CLIPS's multiple-choice quiz.*

variables and wildcards, (7) *rules definition*, (8) *rules execution*, (9) *rule patterns*, (10) *rule conditional element*, (11) *modules*, (12) *functions and actions,* and (13) *commands.* Then, the instructor specified a value for each possible relation between all the questions and concepts, in our case, 0 shows no relation, 1 shows full relation, and 0.5 shows a half relation between item/question and concept. On the other hand, a hill-climbing algorithm (Barnes, 2011) can be also used for obtaining automatically the relationships matrix. This method is useful when the number of concepts and/or items increase due to it is more difficult to create the matrix by hand.
- **Knowledge matrix**. This matrix has been created automatically and each value is an integer value between 0 and 5, since in our case each concept has a maximum of five full related items/questions. Then, these values have been normalized to a real value between 0 and 10.

Next, all continuous values (integer and real values) of the score and knowledge matrices have been converted into labels (nominal values) in the following way:

- **Item answers:** The *CORRECT* label is assigned if the value is 1, whereas the *INCORRECT* label is used if the value is 0.
- **Concept knowledge level:** The *LOW* label is set if the value is lower than 5, the *NORMAL* label if the value is higher than or equal to 5 and less than or equal to 7, and the *HIGH* label if the value is higher than 7. These specific cut-offs (5 and 7) have been used due to that the Spanish universities use a similar 10-point grading scale for assessing the student's performance (0–4.9: FAIL, 5–6.9: PASS, 7–10: VERY GOOD or EXCELLENT).
- **Used time:** The *ALL* label denotes if the value is higher than or equal to 34 (students that take all or almost all the time provided) and the *NOT-ALL* label if the value is lower than 34 (students that have time left).
- **Final score:** The *FAIL* label is used if the value is lower than 5, the *PASS* label if the value is higher than or equal to 5 and lower than 7, and the *EXCELLENT* label if the value is higher than 7. Again, these specific cut-offs (5 and 7) are based on the Spanish grading scale.

Finally, the score matrix and knowledge matrix have been saved in two different data files with a Comma-Separated Values (CSV) format that is a standard file format for interchanging data used in most spreadsheets and DM frameworks.

3.4. G3PARM parameters

In G3PARM (Luna *et al.*, 2011), each individual is evaluated by a fitness function based on support measure. In this paper, however, a new fitness function is applied in order to maximize both support and confidence measures, so the new fitness function is defined as: $Fitness = w_1 \times Sup + w_2 \times Conf$, where *Sup* and *Conf* are respectively the support and confidence of the individual/rule and w_1 and w_2 are parameters to weight support and confidence, respectively. Notice that $w_1 + w_2 = 1$. Both parameters should be established based on the importance of support and confidence in the resultant individual. A high value of w_1 allows to discover rules having high support and, therefore, having high confidence. On the other hand, a high value of w_2 allows to discover rules having high confidence. However, the mere fact of having a high confidence does not guarantee a high support. To study the behaviour of these parameters, a series of experiments was carried out by using values from 0.1 to 0.9 because values 0 and 1 are not permitted. Notice that a 0 value implies the suppression of the parameter, whereas a 1 value implies the suppression of the opposite parameter. Since our goal is the discovery of both frequent and reliable association rules, the results of our experiments show that support and confidence must be equally weighted so $w_1 = w_2 = 0.5$.

Table 1: *IF-THEN rule syntax expressed in Extended BNF notation*

```
<rule> :: = IF <antecedent> THEN <consequent>
<antecedent> :: = <condition> +
<consequent> :: = <condition> +
<condition> :: = <attribute> = <value>
<attribute> :: = Each one of the possible attributes set
<value> :: = Each one of the possible values of the attributes set
```

A major feature of G3PARM, which makes it different from the existing EA, is the use of a context-free grammar (CFG). This CFG (see Table 1) allows to adapt and apply the algorithm to each specific problem or domain. Furthermore, the use of a CFG restricts the search space, defining the syntax of a general from of association rules.

It should be noted that more constrained forms of association rules can be easily represented by use of a more restrictive grammar. For example, to represent rules with a predefined number of conditions in the antecedent, consequent or both the antecedent and consequent. More specifically, the use of a grammar to define rules allows to represent rules with a specific attribute (items, scores, times, or concepts in this problem) in the rule antecedent or consequent. Therefore, instructors can focus the search for specific types of patterns by only predefining a different grammar. None of the existing algorithms, neither classical approaches nor genetic ones, provides this advantage, G3PARM being a pioneer algorithm in this sense.

3.5. Executions and comparisons

In order to verify the performance of our algorithm, G3PARM has been applied on the two previously preprocessed quiz data files and its results have been compared with those of other ARM algorithms. Despite G3PARM provides the possibility of mining quantitative association rules, numerical attributes were discretized since categorical association rules are more comprehensible and this preprocessing step is required by classical ARM algorithms. The algorithms used in the comparison are Apriori, ECLAT, and FP-Growth and EA algorithms such as GAFuzzyApriori, EARMGA, and our proposed G3PARM algorithm. All of them are implemented on Java Language. In this experiment, all these algorithms have been executed with the pre-processed score matrix and knowledge matrix data files. For all the algorithms, the support and confidence threshold values used with the score matrix are 0.6 and 0.9, respectively, and the support and confidence threshold values used

Table 2: *Association rule mining results from the score matrix data file*

Algorithm	*Number of rules*	*% Average support*	*% Average confidence*	*% Coverage*	*Average lenght*	*Time*
Apriori	166	64.75%	94.92%	100%	2.957	0.487
ECLAT	166	64.75%	94.92%	100%	2.957	0.323
FP-Growth	166	64.75%	94.92%	100%	2.957	0.275
GAFuzzyApriori	135	65.08%	95.52%	100%	2.965	5.740
EARMGA	132	65.12%	95.85%	100%	2.970	4.694
G3PARM	125	66.77%	96.27%	100%	3.070	4.098

Table 3: *Association rule mining results from the knowledge matrix data file*

Algorithm	*Number of rules*	*% Average Support*	*% Average confidence*	*% Coverage*	*Average length*	*Time*
Apriori	120	32.76%	97.43%	100%	3.179	0.213
ECLAT	120	32.76%	97.43%	100%	3.179	0.197
FP-Growth	120	32.76%	97.43%	100%	3.179	0.178
GAFuzzyApriori	102	33.02%	98.16%	100%	3.261	4.992
EARMGA	98	33.56%	98.30%	100%	3.264	3.272
G3PARM	82	34.63%	98.97%	100%	3.280	3.018

with the knowledge matrix are 0.3 and 0.9, respectively. For genetic algorithms, the specific parameters used are the same for both matrices: 500 as the number of generations, 0.7 as crossover probability, and 0.1 as the mutation probability. All these parameters and threshold values have been established after a previous experimental study for obtaining these optimal values when using our specific data. Focusing on support and confidence threshold values, it should be noted that they do not significantly influence on the evolutionary proposals. However, classical algorithms follow an exhaustive search methodology, and the fact of using low threshold values in support and confidence could imply a huge increment in both their execution time and number of rules discovered. Therefore, these thresholds depend on the dataset used. That was the reason because we had to use two different support thresholds for the knowledge matrix and for the score matrix, that is 0.3 and 0.6, respectively.

Tables 2 and 3 show the results obtained by each algorithm. Both tables show the total number of rules discovered, the average support (percentage of instances that contain both antecedent and consequent among all instances in the dataset), the average confidence (percentage of instances that contain the consequent among instances that contain the antecedent), the percentage of instances in the dataset covered by the rules mined, the average length of the rules (number of items in antecedent and consequent), and the execution time (in seconds).

Similar conclusions can be obtained after analysis of both Tables 2 and 3. Our G3PARM algorithm discovers a lower number of rules than classic Apriori-based algorithms and the other Fuzzy and Genetic algorithms. It also discovers rules with the best support and confidence. Focusing on the coverage and length of the rules, all the algorithms cover all the data (100%) and all the discovered rules have almost the same length. Finally, classic algorithms are the fastest, as was expected from a small dataset like the two current datasets used. Our G3PARM algorithm, however, is faster than the other two EA algorithms. However, it should be noted that the runtime of the exhaustive search algorithms for ARM increases faster than the runtime of the EA when the size of the dataset grows. With the increment of the dataset size, the search space increment and, therefore, exhaustive search algorithms become untenable. The robustness of our approach is that it does not suffer with the increment of the dataset size. Since G3PARM is an EA, its runtime scales quite linearly as dataset size and the number of attributes increases up.

In general, classic algorithms are exhaustive searcher algorithms and so they discover the same rules and the only difference between them is the execution time. EA algorithms discover different numbers of rules as they use different search methods; for example, EAs use different codification schemes and fitness functions. In this case, the proposed G3PARM algorithm has shown the best performance when using the data files and parameters mentioned above, discovering the lowest number of high-quality rules in an acceptable time.

3.6. *Results interpretation for provision of feedback*

The objective of the result interpretation is to analyse the set of discovered rules in order to provide feedback for improving quiz and course. ARM algorithms normally discover a huge number of rules but not all are interesting for the user. In our case, not all the rules discovered by the G3PARM algorithm have the same interest. For example, there are some misleading rules that show relationships about items or concepts that contain opposite values, such as to get right one or several questions but get others wrong or obtain a high level of knowledge in one or several concepts but obtain a low level in other concepts. Two examples of rules obtained that show this type of relations are the next:

IF *Item-Num.19* = *CORRECT* **THEN** *Item-Num.8* = *INCORRECT* (Support = 0.640, Confidence = 0.916)

IF *Rules-Definition* = *HIGH* **THEN** Ordered-Facts = *LOW* (Support = 0.301, Confidence = 0.905)

Logically, this type of relationships does not provide us interesting information about how to improve the quiz or the course, and so, these rules are ruled out. Then, working with the remaining rules, we want to find which ones are the most interesting and useful for our objective. However, the usefulness and interestingness of rules are subjective concepts that are difficult to quantify effectively. Traditionally, two different approaches or measures have been used in order to evaluate rule interest: objective and subjective (Geng & Hamilton, 2006).

On the one hand, objective data-driven measures such as support and confidence are based only on the raw data and use the numerical or structural properties of a rule. The interest of mining association rules over frequent patterns lies in the extraction of rules that satisfy many dataset instances. However, the discovery of rules with maximal support, for example, those that cover all the dataset instances, does not provide any new information and so is normally not interesting. Another well-known objective measure that has been used with success in educational environment as a measure of interestingness is the interest factor or lift (Merceron & Yacef, 2008), which is defined as:

$$Lift(A \rightarrow C) = \frac{Confidence(A \rightarrow C)}{Support(C)}$$

The lift measure is used to represent the reliability of the rule by calculating how many times more often the antecedent and consequent are related in a dataset than would be expected if they were statistically independent. Lift values less than or equal to the unity provide misleading rules since the consequent is more frequent than when the antecedent appears in the rule. On the contrary, lift values greater than the unity provide a positive dependence between the antecedent and consequent, and so, the rule is potentially useful for predicting the consequent in future data sets. In our experiments using G3PARM with the previous matrixes and parameters, all the rules discovered had a lift value greater than the unity, so all the rules discovered have been considered as interesting rules.

On the other hand, subjective and semantic user-driven measures and approaches incorporate a user's background knowledge or domain knowledge in order to obtain information about the utility of the rules. End users have a crucial role in this form of post-mining since they can guide the search for the best rules. In our case, three experts in the CLIPS domain have identified five general patterns and 10 different types of interesting relations in order to help instructors to find the most interesting rules that match these patterns and relations. This post-mining process has been carried out in the next way. First, the three experts identified the types of general patterns considering the information provided by the matrices. Then, they identified the types of interesting relations analysing all the rules discovered by the G3PARM algorithm. Finally, they identified the best obtained rules for each type of relation. Next, some examples of the rules previously discovered by our G3PARM algorithm are classified by type of pattern and relation together with information about the feedback provided by each one. Rules obtained by means of the score matrix show relationships between items, times, and scores, so three types of patterns of rules can be distinguished.

Item-item pattern. This pattern shows relationships between several items/questions. Experts have distinguished the following two types of relations:

(1) Relations between to get several right questions. They show that if students get one right item then they also get another one or more right items. An example of an obtained rule of this type is:

IF *Item-Num.12 = CORRECT* ***AND*** *Item-Num.29 = CORRECT*

THEN *Item-Num.38 = CORRECT*

(Support = 0.631, Confidence = 0.902, Lift = 1.177)

(2) Relations between to get several wrong questions. They show that if students get one wrong item then they also get another one or more wrong items. An example of an obtained rule of this type is:

IF *Item-Num.24 = INCORRECT* ***AND*** *Item-Num.35 = INCORRECT*

THEN *Item-Num.8 = INCORRECT*

(Support = 0.640, Confidence = 0.985, Lift = 1.193)

In general, these two types of relationship show items/questions that could evaluate the same or closely related concepts. Relation type 1 could identify easy questions that could have a low level of difficulty, so the instructor should check the content of these questions in order to increase or not their difficulty, if necessary. In a similar way, relation type 2 could identify difficult questions that could have a high level of difficulty or could contain an error or typo. The instructor must check the content of these questions in order to decrease their difficulty if necessary or to correct the possible error. Moreover, both type of relations also could show a evidence of multiple items tapping the same construct, or pre-requisite relationships.

Item-score pattern. This pattern shows relationships between items and scores and experts have distinguished two very interesting subtypes:

(3) Relations between to get wrong items/questions and fail the exam. For example, the next rule shows that if students make a mistake in items 1 and 29, then they obtain a *FAIL* score.

IF *Item-Num.1 = INCORRECT* ***AND*** *Item-Num.29 = INCORRECT*

THEN *Score = FAIL*

(Support = 0.631, Confidence = 0.984, Lift = 1.102)

(4) Relations between to get right items/questions and pass the exam with a high score. For example, the next rule shows the opposite relation to the previous one, showing that getting correct answers is related to an *EXCELLENT* score.

IF *Item23 = CORRECT* ***AND*** *Item31 = CORRECT*

THEN *Score = EXCELLENT*

(Support = 0.621, Confidence = 0.941, Lift = 1.227)

The instructor could select the specific questions that appear in these rules as good discriminatory items of *FAIL* and *EXCELLENT* scores, respectively. For example, the instructor could create a new and shorter version of the quiz made up of only this type of items because they are the most discriminatory ones.

Item-time-score pattern. It shows relationships between items, times, and scores, and experts have distinguished the following two types of interesting relations:

(5) Relation between to get wrong items, the score obtained, and the time spent. For example, the following rule shows

that students who get wrong item 40 and also obtain a *FAIL* score then use *ALL* the time provided.

IF *Item-Num.40 = INCORRECT* ***AND*** *Score = FAIL*

THEN *Time = ALL*

(Support = 0.611, Confidence = 0.926, Lift = 1.084)

(6) Relation between to get right items, the score obtained, and the time used. For example, the following rule shows that students who get right item number 26 and do not use all the time provided obtain a *GOOD* score.

IF *Item-Num.26 = CORRECT* ***AND*** *Time = LESS*

THEN *Score = GOOD*

(Support = 0.660, Confidence = 1.0, Lift = 1.051)

Again, the instructor can use questions that appear in these rules as good discriminatory items of the final score obtained. The instructor could also consider providing less or more time to execute the quiz because of the relationships between to use (or not) all the time provided and the score obtained.

With respect to rules obtained using the knowledge matrix shows relationships between concepts and scores, so two different types of patterns can be distinguished.

Concept-concept pattern. It shows relationships between different concepts and experts have distinguished two types of relations:

(7) Relations between to obtain a low level of knowledge in several concepts. For example, the following rule shows that if students have a *LOW* knowledge level in the *Rules Definition* and *Rule Conditional Element* concepts, then they also have a *LOW* knowledge level in the *Rules Execution* concept:

IF *Rules-Definition = LOW* ***AND*** *Rule-Conditional Element = LOW*

THEN *Rules Execution = LOW*

(Support = 0.466, Confidence = 0.960, Lift = 1.177)

(8) Relations between to obtain a high level of knowledge in several concepts. For example, the following rule shows that students who obtain a *HIGH* knowledge level in *Initial Facts*, *Rule Conditional Element,* and *Rules Execution* concepts also obtain a *HIGH* knowledge level in the *Functions and Actions* concept.

IF *Initial-Facts = HIGH* ***AND*** *Rule-Conditional-Element = HIGH* ***AND*** *Rules-Execution = HIGH*

THEN *Functions-And-Actions = HIGH*

(Support = 0.388, Confidence = 0.952, Lift = 1.167)

These two relations show two instructor concepts that are closely related. The first relation, however, can be used to detect concepts that could have brief or unsuitable contents in the course. The instructor should check the contents of these chapters in the course in order to decide if they should be modified or extended for improvement. The second relation can be used to detect good concepts that are related. The instructor must check if these chapters are located together in the course in order to decide if they could be placed closer together or even combine in only one concept.

Concept-score pattern. It shows relationships between concepts and scores, and experts have distinguished the following two types of interesting relations:

(9) Relations between to obtain a high level of knowledge in one or several concepts and a high score in the exam. For example, the following rule shows that if students have a *HIGH* knowledge level in *Rule Conditional Element* and *Rules Execution* concepts, then they obtain an *EXCELLENT* score.

IF *Rule-Conditional-Element = HIGH* ***AND*** *Rules-Execution = HIGH*

THEN *Score = EXCELLENT*

(Support = 0.320, Confidence = 1.0, Lift = 1.907)

(10) Relations between to obtain a low level of knowledge in one or several concepts and to obtain a *LOW* score in the exam. For example, the following rule shows that students who have a *LOW* knowledge level in *Initial Facts* and *Variables and Wildcard* concepts also obtain a *FAIL* score.

IF *Initial-Facts = LOW* ***AND*** *Variables-And-Wildcards = LOW*

THEN *Score = FAIL*

(Support = 0.310, Confidence = 1.0, Lift = 1.226)

These relations show the instructor the most influential concepts for obtaining a good or a bad score. The instructor must check the content of these chapters in order to decide if they should be modified or extended for improvement.

4. Evaluating updates done in quiz and course

A pilot experiment was conducted to evaluate the effect of applying updates in the CLIPS quiz and course starting with the feedback provided by the previously discovered rules. The hypothesis was that the updated quiz and course would have a beneficial effect on student performance. The list of specific updates in the CLIPS quiz and in the course is as follows:

- The contents of some questions have been modified starting with information provided by relations of types 1 and 2. The instructor checked and realized that two questions (numbers 24 and 35) were really very difficult questions and four questions (numbers 12, 14, 17, and 38) were really very easy questions. The instructor therefore decided to modify these questions briefly in order to decrease or increase their level of difficulty. The instructor also checked and realized that question number 9 had an error in the answer and so it was modified to correct it.
- Some questions have been removed from the quiz starting with the information provided by relations of types 1, 2, 3, 4, and 5. In fact, some questions did not appear in any of the rules obtained and it could indicate that they did not affect significantly to the final score. The instructor decided, after reviewing the content of these questions, to remove five questions (3, 15, 20, 22, and 30) as they treated specific CLIPS anecdotes and they were not important questions.

- The time available to respond to each question has been increased from the information of relations of types 5 and 6. Although the maximum total time provided to students was the same (35 min), the average time to respond to each question was increased (from 52.5 to 60 s) because the number of questions was reduced (from 40 to 35).
- The contents of some concepts in the course have been improved. The instructor decided, after reviewing the content of chapters referenced by the discovered rules of type 7, 8, 9, and 10, that the next four chapters/concepts should be extended in order to improve their contents: *Variables and Wildcards, Rules Definition, Rules Execution*, and *Rule Conditional Element.*
- Some chapters have been moved from their location in the course. Starting from the relationships of types 7 and 8, the instructor decided that chapter 5, *Initial Facts*, should be moved to chapter 6 and chapter 12, *Function And Actions*, to 11, in order that these concepts could be closer to the other specific chapters about rules (chapters 7–10).

It is important to notice that the post-mining process that we have used to select the rules is subjective because it is based on three experts, that is, other authors could identify different interesting relations and rules. So, the list of updates in the CLIPS quiz and in the course could also vary depending upon the experts used.

After application of all these specific updates, two new groups of students took the updated course and quiz during the 2009–2010 and 2010–2011 academic years. The effectiveness of these updates was evaluated in the light of the performance by the students, that is, by comparing the score obtained with this updated quiz and course versus the original quiz and course. Therefore, there were three groups of students: one control and two experimental groups. On the one hand, a total of 104 students (control group) took the original Moodle CLIPS course and quiz during the 2008–2009 academic year. On the other hand, 98 students (experimental group one) and 102 students (experimental group two) took the updated course and quiz during the 2009–2010 academic year. All these groups of students had similar characteristics (age, previous experience, and knowledge in computer science, etc.) because all were students on the same course (second year) of the computer science engineering degree at Cordoba University.

Two experimental studies are carried out to analyse if there were differences between the scores obtained (from 0 to 10 points) by the different groups of students in order to be able to evaluate if really the application of the updates have improved the course and the quiz. On one hand, the first study evaluates both the changes on the quiz and the course. On the other hand, the second study evaluates only the changes on the course. For both studies was considered a statistical analysis. First of all, it was checked that the values of the obtained scores in all the groups were normally distributed in order to decrease the risk of error. Histograms of the scores obtained showed a bell-shaped curve for all the groups. Then, descriptive and comparative statistics were calculated (see Table 4) such as: the number of questions in each exam/quiz (Q), the number of students in each group (N), the score average value (*Mean*), the mean difference between two groups (*Mean Differences*), the standard deviation (*SD*) that quantifies score variability, standard error of the mean (*SEM*) that gives an idea of the accuracy of the mean, and the Student t-Test p value (p) that compares the means of two groups.

In the first study (see comparison 1 and 2 in Table 4), the three groups were compared in pairs (control versus experimental), that is, the scores obtained in 1 year were compared versus the scores obtained in the other year. As we have commented, the objective was to test whether the updates done both in the quiz and in the course had an effect (positive or negative) on the student scores. So, all items of the quiz (40 and 35, respectively) have been used for obtaining the final score. Table 4 shows that the differences between the control group and the two experimental groups can be considered statistically significant with a confidence level of 99% ($p < 0.01$). In fact, there is a mean difference of 0.68 points (in a scale from 0 to 10 points), with 2009–2010 students scoring better than 2008–2009 students, and 0.71 points with 2010–2011 students scoring better than 2008–2009. So, updates done in the quiz and the course can be considered as very positive to obtain better scores.

In the second study (see comparison 3 and 4 in Table 4), the three previous groups but using the same quiz were compared in pairs (control versus experimental), that is, the student scores were obtained starting on only the common 29 items in both quizzes. In this case, the six modified questions/items and the five deleted questions were not used to obtain the student scores. The objective was to test whether the updates done only in the course had an effect (positive or negative) on the student scores. Table 4 shows that the differences can be considered statistically significant with a confidence level of 95% ($p < 0.05$). Again, 2009–2010 and 2010–2011 students scoring better than 2008–2009 students, with a similar but less mean difference of 0.65 and 0.68 points, respectively. So,

Table 4: *Descriptive statistics of each group and pairwise comparison between control and experimental groups (*only the common items in both quizzes are used)*

Number of comparison	*Group*	*Q*	*N*	*Mean*	*Mean difference*	*SD*	*SEM*	*p*
1	Control2008–09	40	104	5.9469	0.6844	1.8074	0.1772	0.0060
	Exp2009–10	35	98	6.6313		1.6758	0.1702	
2	Control2008–09	40	104	5.9469	0.7113	1.8074	0.1772	0.0041
	Exp20010–11	35	102	6.6583		1.7093	0.1692	
3	Control2008–09*	29	104	5.9904	0.6502	1.8607	0.1825	0.0109
	Exp2009–10*	29	98	6.6406		1.7156	0.1742	
4	Control2008–09*	29	104	5.9904	0.6804	1.8607	0.1825	0.0103
	Exp20010–11*	29	102	6.6708		1.9112	0.1892	

updates done only in the course can be also considered as positive on the performance of the students.

5. Conclusions

In this paper, ARM is applied to provide the instructor with interesting relationships that can be useful for decisions whether or not to update and improve a Moodle CLIPS quiz and course. Some important conclusions have been obtained. On the one hand, two new data matrixes were used together with the traditional score matrix in order to reveal more useful information about quiz and course. On the other hand, a grammar-based GP approach was very useful for discovering interesting rules. Experimental results using real data from computer science students at the University of Cordoba who performed a multiple-choice test on CLIPS programming language showed that our G3PARM algorithm discovered a lower number of rules with higher quality than other approaches. In order to find the most interesting rules, subjective and objective measures were used. In fact, the lift value is shown together with the support and confidence for each obtained rule. The most useful types of patterns and relationships were detected for three experts. Finally, a pilot experiment was conducted to evaluate the effect of applying updates in the quiz and course starting from the feedback provided by the previously discovered rules. The results of the statistical analysis of two comparisons of a control and an experimental group of students show that the updates to quizzes and courses provided by our algorithm had a positive and statistically significant impact on student learning. Overall, the results from the current study suggest that ARM can be very useful for providing feedback about how to enhance online quizzes and courses.

In future work, it would be interesting to repeat the experimentation using more data from different types of courses (other engineering fields or other disciplines) and also different settings (elementary, junior high, and high school) in order to test if the same type of rules are discovered, or if the same differences in the means between control and experiment groups are obtained and if they can also be considered statistically significant. It would be also very useful to do experiments using more experts in order to investigate its effect in the post-mining process and to test different approaches to identify and select interesting relations and rules. For example, in our case, the three experts had an in-person meeting but if there is more number of experts involved in the process, it could be useful to use an online communication and using a voting approach (García *et al.*, 2009). Finally, it is important to mention that although the quiz data used in this paper are gathered from the Moodle quiz module, other quiz data gathered from different learning management and quizzing systems are very similar and so they can also be pre-processed and mined in a similar way to that described in this paper.

Acknowledgement

This research is supported by projects of the Regional Government of Andalucia and the Ministry of Science and Technology, P08-TIC-3720 and TIN-2011-22408, respectively, and FEDER funds. Furthermore, this work has been supported by the Spanish Ministry of Education under the FPU grant AP2010-0041.

References

AGRAWAL, R. and R. SRIKANT (1994) Fast algorithms for mining association rules. *International Conference on Very Large Data Bases*, Santiago de Chile, Chile, 487–499.

BARNES, T. (2005) The q-matrix method: mining student response data for knowledge, *Proceedings of the AAAI-2005 Workshop on Educational Data Mining*, Pittsburgh, PA, 1–8.

BARNES, T. (2011) *Novel derivation and application of skill matrices: The q-matrix method. Handbook of educational data mining*. In C. Romero, S. Ventura, M. Pechenizkiy, R.S.J.d. Baker (eds.), Chapman & Hall/CRC Press, New York, 159–172.

BRUSILOVSKY, P. and P. MILLER (1998) Web-based testing for Distance Education, *World Conference of WWW and Internet*, Orlando, Florida, 149–154.

CEGLAR, A. and J. RODDICK (2006) Association mining. *ACM Computing Surveys*, **38**, 1–42.

CHEN, Y. and C. WENG (2009) Mining fuzzy association rules from questionnaire data. *Knowledge-Based Systems Journal*, **22**, 46–56.

CHU, H.C., G.J. HWANG, J.C.R. TSENG and G.H. HWANG (2006) A computerized approach to diagnosing student learning problems in health education. *Asian Journal of Health and Information Sciences*, **1**, 43–60.

DOUGHERTY, J., M. KOHAVI and M. SAHAMI (1995) Supervised and unsupervised discretization of continuous features, *International Conference on Machine Learning*, Tahoe City. CA, 194–202.

GARCIA, E., C. ROMERO, S. VENTURA and C. CASTRO (2009) An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, **19**, 99–132.

GENG, L. and H.J. HAMILTON (2006) Interestingness measures for data mining: a survey. *ACM Computing Surveys*, **38**, 1–32.

HAN, J., J. PEI and Y. YIN (1999) Mining frequent patterns without candidate generation, *ACM-SIGMOD International Conference on Management of Data*. 85–93.

HONG, T.P., C.H. CHEN, Y.L. WU and Y.C. LEE (2006) A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. *Soft Computing*, **10**, 1091–1101.

KUECHLER, W. L. and M.G. SIMKIN (2003) How well do multiple choice tests evaluate student understanding in computer programming classes? *Journal Information System Educaction*, **14**, 389–399.

LUNA, J.M., J. R. ROMERO and S. VENTURA (2011) Design and behavior study of a grammar-guided genetic programming algorithm for mining association rules, Knowledge and Information Systems, pp. 1–24, DOI: 10.1007/s10115-011-0419-z.

MCKAY, R., N. HOAI, P. WHIGHAM, Y. SHAN and M. O'NEILL (2010) Grammar-based genetic programming: a survey. *Genetic Programming and Evolvable Machines*, **11**, 365–396.

MERCERON, A. and K. YACEF (2008) Interestingness measures for association rules in educational data, *International Conference on Educational Data Mining*, Montreal, 57–66.

PECHENIZKIY, M., T. CALDERS, E. VASILYEVA and P. DE BRA (2008) Mining the student assessment data: lessons drawn from a small scale case study, *International Conference on Educational Data Mining*, Cordoba, Spain, 187–191.

ROMERO, C. and S. VENTURA (2010) Educational data mining: a review of the state-of-the-art. *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, **40**, 601–618.

ROMERO, C., S. VENTURA and E. SALCINES (2008) Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, **51**, 368–384.

ROMERO, C., S. VENTURA and P. DE BRA (2009) Using mobile and web-based computerized tests to evaluate university students. *Computer Applications in Engineering Education*, **17**, 435–447.

ROMERO, C., J.M. LUNA, J.R. ROMERO and S. VENTURA (2011) RM-Tool: a framework for discovering and evaluating association rules. Advances in Engineering Software, **42**, 566–576.

YAMANISHI, K. and H. LI (2001) Mining from open answers in questionnaire data, *Proceedings of the Seventh ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, 443–449.

YAN, X., C.H. ZHANG and S. ZHANG (2009) Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. *Expert Systems with Applications*, **36**, 3066–3076.

Zaki, M.J. (2000) Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, **12**, 372–390.

The authors

Cristóbal Romero

Cristóbal Romero is currently an Associate Professor in the Department of Computer Science and Numerical Analysis at the University of Córdoba. He received his BSc and PhD degrees in computer science from the University of Granada, Spain, in 1996 and 2003, respectively. He has published more than 40 international papers, 15 of which have been published in international journals. He is the co-editor of two books specifically regarding EDM. His current research interest is focussed on the application of data mining in e-learning systems. Dr. Romero is a member of the IEEE Computer Society, the International EDM Working Group and the steering committee of several conferences about education, personalisation, and data mining.

Amelia Zafra

Amelia Zafra received the BS degree and PhD degrees in Computer Science from the University of Granada, Spain, in 2005 and 2009, respectively. She is currently an Assistant Professor with the Department of Computer Science and Numerical Analysis, University of Cordoba, Spain. Her research labour is developed as a member of the Knowledge Discovery and Intelligent Systems Research Laboratory, and it is focused on the fields of soft computing, data mining, and machine learning with evolutionary algorithms and their applications.

Jose María Luna

Jose María Luna received a BSc degree from the University of Córdoba in 2007, and an MSc degree from the same university in 2009, both in Computer Science. Since 2009, he has been with the Department of Computer Science and Numerical Analysis, at the University of Córdoba, where he is currently working towards obtaining a PhD, as well as at research tasks. His research interests include the application of evolutionary computation, association rule mining, and its applications. José María Luna is a Student Member of the IEEE Computer, Computational Intelligence and Systems, Man and Cybernetics societies.

Sebastián Ventura

Sebastián Ventura is currently an Associate Professor in the Department of Computer Science and Numerical Analysis at the University of Córdoba, where he heads the Knowledge Discovery and Intelligent Systems Research Laboratory. He received his BSc and PhD degrees in sciences from the University of Córdoba, Spain, in 1989 and 1996, respectively. He has published more than 90 international publications, 35 of which have been published in international journals. He has also been engaged in 11 research projects (being the coordinator of three of them) supported by the Spanish and Andalusian governments and the European Union. His main research interests are in the fields of soft-computing, machine learning, data mining, and their applications. Dr. Ventura is a senior member of the IEEE Computer, the IEEE Computational Intelligence and the IEEE Systems, Man and Cybernetics Societies, as well as the Association of Computing Machinery (ACM).