Author Proof

Web Usage Mining for Predicting Final Marks of Students That Use Moodle Courses

ROMERO CRISTOBAL,[1] PEDRO G. ESPEJO,[2] ZAFRA AMELIA,[3] ROMERO JOSE RAUL,[2] VENTURA SEBASTIAN[2]

[1] *Universidad*Q2 *de Cordoba, Informatica, Cordoba, Spain*

[2] *Cordoba University, Computer Science, Cordoba, Spain*

[3] *Universidad de Cordoba, Computer Science, Cordoba, Spain*

Received 10 March 2009; accepted 3 May 2010

ABSTRACT: This paper shows how web usage mining can be applied in e-learning systems in order to predict the marks that university students will obtain in the final exam of a course. We have also developed a specific Moodle mining tool oriented for the use of not only experts in data mining but also of newcomers like instructors and courseware authors. The performance of different data mining techniques for classifying students are compared, starting with the student's usage data in several Cordoba University Moodle courses in engineering. Several well-known classification methods have been used, such as statistical methods, decision trees, rule and fuzzy rule induction methods, and neural networks. We have carried out several experiments using all available and filtered data to try to obtain more accuracy. Discretization and rebalance pre-processing techniques have also been used on the original numerical data to test again if better classifier models can be obtained. Finally, we show examples of some of the models discovered and explain that a classifier model appropriate for an educational environment has to be both accurate and comprehensible in order for instructors and course administrators to be able to use it for decision making. © 2010 Wiley Periodicals, Inc. Comput Appl Eng Educ 9999: 1–1, 2010; Published online in Wiley InterScience (www.interscience.wiley.com); DOI 10.1002/cae.20456

Keywords: educational data mining; classifying students; predicting marks; learning management systems

INTRODUCTION

The use of web-based education systems or e-learning systems has grown exponentially in the last years, spurred by the fact that neither students nor teachers are bound to any specific location and that this form of computer-based education is virtually independent of a specific hardware platform [1]. In particular, collaborative and communication tools are also becoming widely used in educational contexts and as a result. Learning Management Systems (LMSs) are becoming much more common in universities, community colleges, schools, and businesses, and are even used by individual instructors in order to add web technology to their courses and supplement traditional face-to-face courses [2]. LMSs can offer a great variety of channels and workspaces to facilitate information sharing and communication among participants in a course. They let educators distribute information to students, produce content material, prepare assignments and tests, engage in discussions, manage distance classes and enable collaborative learning with forums, chats, file storage areas, news services, etc. Some examples of commercial systems are Blackboard [3] and TopClass [4] while some examples of free systems are Moodle [2], Ilias [5], and Claroline [6]. Nowadays, one of the most commonly used is Modular Object Oriented Developmental Learning Environment (Moodle), a free learning management system enabling the creation of powerful, flexible and engaging online courses and experiences [7].

LMSs accumulate a vast amount of information which is very valuable for analyzing students' behavior and could create a gold mine of educational data [8]. They can record any student activities involved, such as reading, writing, taking tests, performing various tasks, and even communicating with peers [9]. They normally also

Correspondence to Romero Cristobal (Campus de Rabanales, Edificio C2, Cordoba 14071, Spain. cromero@uco.es).
© 2010 Wiley Periodicals, Inc.

Author Proof

provide a database that stores all the system's information: personal information about the users (profile), academic results and users' interaction data. However, due to the vast quantities of data that these systems can generate daily, it is very difficult to manage manually. Instructors and courseware authors request tools to assist them in this task, preferably on a continual basis. Although there are platforms that offer some reporting tools, it gets harder for a tutor to extract useful information when there are a great number of students [10]. Unfortunately, these platforms do not provide specific tools to allow educators to thoroughly track and assess all learners' activities while evaluating the structure and contents of the course and its effectiveness in the learning process [11]. The use of data mining is a very promising means to achieve these ends [12].

In the last few years, researchers have begun to apply data mining methods to help instructors, courseware authors, administrators, etc. to improve educational systems [13]. Educational Data Mining (EDM) is an emerging interdisciplinary research area that deals with the application of Data Mining (DM) techniques to educational data [14]. Data mining or knowledge discovery in databases (KDD) is the automatic extraction of implicit and interesting patterns from large data collections [15]. Data mining is a multidisciplinary area in which several computing paradigms converge: decision tree construction, rule induction, artificial neural networks, instance-based learning, Bayesian learning, logic programming, statistical algorithms, etc. And some of the most useful data mining tasks and methods are: statistics, visualization, clustering, classification, and association rule mining. These methods uncover new, interesting and useful knowledge based on students' usage data. Some examples of e-learning problems or tasks that data mining techniques have been applied to [16] are: dealing with the assessment of students' learning performance, providing course adaptation and learning recommendations based on the students' learning behavior, dealing with the evaluation of learning material and educational web-based courses, providing feedback to both teachers and students of e-learning courses, and detection of atypical students' learning behavior. In most of these educational tasks or problems it is necessary to predict/classify a student's performance [17], in fact, one of the most useful and oldest DM tasks in education is classification.

A classifier is a mapping from a (discrete or continuous) feature space X to a discrete set of labels Y [18]. This is supervised classification which provides a collection of labeled (pre-classified) patterns, the problem being to label a newly encountered, as of yet unlabelled pattern. In classification, the goal is to learn a model to predict the class value, given other attribute values. There are different objectives and applications for using classification in an educational environment, such as: discovering potential student groups with similar characteristics and reactions to a particular pedagogical strategy [19]; detecting students' misuse or game-playing [20]; grouping students who are hint-driven or failure-driven, and find common misconceptions that students possess [21]; identifying learners with low motivation and finding remedial actions to lower drop-out rates [22]: predicting/classifying students when using intelligent tutoring systems [23], etc. There are different types of classification methods and artificial intelligent algorithms that have been applied to predict student outcome, marks or scores. Some examples are: predicting students' grades (classifying them in five classes: A, B, C, D, and E or F) from test scores using neural networks [24]; predicting student academic success (classes that are successful or not) using discriminant function analysis [25]; classifying students using genetic algorithms to predict their final grade [26]; predicting a student's academic success (to classify as low, medium, and high risk classes) using different data mining methods [27]; predicting a student's marks (pass and fail classes) using regression techniques in Hellenic Open University data [28] or using neural network models from Moodle logs [29].

This paper compares different data mining techniques for classifying students (predicting final marks obtained in the course) based on student usage data in a Moodle course. We have also developed a specific Moodle data mining tool to make this task easier for instructors and course administrators. The paper is arranged in the following order: second section describes the background of the main classification methods and its application in education; third section describes the Moodle data mining tool; fourth section presents the detailed comparison of classification techniques with different datasets; finally, the conclusions and further research are outlined.

BACKGROUND

Classification is one of the problems most frequently studied by DM and machine learning (ML) researchers. It consists of predicting the value of a categorical attribute (the class) based on the values of other attributes (the predicting attributes). In ML and DM fields, classification is usually approached as a supervised learning task. A search algorithm is used to induce a classifier from a set of correctly classified data instances, called the training set. Another set of correctly classified data instances, known as the testing set, is used to measure the quality of the classifier obtained after the learning process. Different kinds of models can be used to represent classifiers, and there are a great variety of algorithms available for inducing classifiers from data. The following paragraphs give a brief description of the classification algorithms used in our work.

- In statistical classification, individual items are placed into groups based on the quantitative information of characteristics inherent in the items (referred to as variables, characters, etc.) and based on a training set of previously labeled items. Statistical approaches are generally characterized by the existence of an explicit underlying probability model, which provides the probability of being in each class rather than simply a classification. Some examples of statistical algorithms are: linear discriminant analysis [30], where the sample space is divided by a series of hyperplanes determined by linear combinations of variables in such a way that the examples belonging to each class are most clearly split; least mean square quadratic [31], the generalization of the previous method in which quadratic surfaces are employed; kernel methods [30] approach the problem by mapping the data into a high dimensional feature space, where each coordinate corresponds to one feature of the data items, transforming the data into a set of points in a Euclidean space. In that space, a variety of methods can be used to find relationships in the data. *K* nearest neighbors [32] is a method in which a data set is used as a reference to classify new instances with the help of a suitable distance measure. In order to classify a new data instance, its *k* nearest neighbors are found, the number of instances of each class is counted for that subset of *k*, and the example to be classified is assigned to the class with the highest count. Statistical methods have a long tradition and have been applied to practical classification problems in different domains, such as biology [33], meteorology [34], finance [35], etc.
- A decision tree is a set of conditions organized in a hierarchical structure [36] that contains zero or more internal nodes and one

Author Proof

or more leaf nodes. All internal nodes have two or more child nodes and contain splits, which test the value of an expression of the attributes. Arcs from an internal node to its children are labeled with distinct outcomes of the test at the internal node. Each leaf node has a class label associated with it. The decision tree is a predictive model in which an instance is classified by following the path of satisfied conditions from the root of the tree until reaching a leaf, which will correspond to a class label. A decision tree can easily be converted into a set of classification rules. Some of the most well-known decision tree algorithms are C4.5 [36] and CART [37]. The main difference between these two algorithms is the splitting criterion used at the internal nodes: C4.5 uses the information gain ratio, and CART employs the gini index. Decision trees have been employed successfully in different domains like agriculture [38], medicine [39], networking [40], etc.

- Rule Induction is an area of machine learning in which IF-THEN production rules are extracted from a set of observations [29]. Rules are a simple and easily comprehensible way to represent knowledge. A rule has two parts, the antecedent and the consequent. The rule antecedent (IF part) contains a combination of conditions with respect to the predicting attributes. Typically, conditions form a conjunction by means of the AND logical operator, but in general any logical operator can be used to connect elemental conditions, also known as clauses. The rule consequent (THEN part) contains the predicted value for the class. This way, a rule assigns a data instance to the class pointed out by the consequent if the values of the predicting attributes satisfy the conditions expressed in the antecedent, and thus, a classifier is represented as a rule set. The algorithms included in this paradigm can be considered as a heuristic state-space search. In rule induction, a state corresponds to a candidate rule and operators correspond to generalization and specialization operations that transform one candidate rule into another. Examples of rule induction algorithms are: CN2 [41], in which antecedent conditions are built in several stages, eliminating the instances covered in each stage; AprioriC [42] is based on the frequency of appearance of each variable in the training set. Some of the rule induction methods employed in this work are evolutionary algorithms (EAs) based on the use of probabilistic search algorithms inspired by certain points of the Darwinian theory of evolution. The essential features shared by all EAs are: the use of a population of candidate solutions; a generational inheritance method including the application of genetic operators like mutation and crossover; and a fitness function used to measure the quality of each individual. The EAs employed in our work are XCS [43], the Supervised Inductive Algorithm (SIA) [44], a genetic algorithm using real-valued genes (Corcoran) [45] and a Grammar-based genetic programming algorithm (GGP) [46]. Different rule-based classification approaches have been applied to medicine [47], networking [48], engineering [23], etc.
- Fuzzy rule induction applies fuzzy logic in order to interpret the underlying data linguistically [49]. To describe a fuzzy system completely, a rule base (structure) and fuzzy partitions have to be determined (parameters) for all variables. Some fuzzy rule learning methods are MaxLogitBoost [50], a boosting-based genetic method (boosting algorithms are statistical additive modeling techniques that combine different low-quality classifiers to obtain a compound classifier that performs better than any of its components); Grammar-based genetic Programming (GP) [51], a hybrid grammar-based genetic programming/genetic algorithm method (GAP) [30], a hybrid simulated annealing/genetic programming algorithm (SAP) [30] (simulated annealing is a method in which each step of the algorithm replaces the current solution by a random nearby solution, chosen with a probability that depends on a parameter called the temperature, that is gradually decreased during the process); and an adaptation of the Wang-Mendel algorithm (Chi) [52] in which fuzzy rules are weighed by the algorithm. Fuzzy classification rules have been used in engineering [53], biology [54], documentation [55], etc.
- An artificial neural network (ANN) is a computational model inspired in biological neural networks. It consists of an interconnected group of artificial neurons, and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Examples of neural network algorithms are: multilayer perceptron (with conjugate gradient-based training) [56], a feedforward ANN which uses three or more layers of neurons with nonlinear activation functions; a radial basis function neural network (RBFN) [57], which enhances multilayer perceptrons, avoiding being trapped in local minima by means of employing distance-based functions; a hybrid genetic algorithm/neural network (GANN) [58]; and neural network evolutionary programming (NNEP) [17], another hybridization of ANNs with an EA (evolutionary programming). Different types of neural networks have been applied to engineering [59], medicine [60], agronomy [61], etc.

We have described these 21 classification algorithms because these specific algorithms are going to be used in our experiment, although there are some other classification techniques such as support vector machine, bayesian networks, etc.

With respect to applying classification in education, there are two main types of educational environments: the traditional educational environment (offline or classroom education) and e-learning or the web-based environment (on-line education).

- Traditional classroom environments are the most widely used educational systems. They are based on face-to-face contact between educators and students and organized in lectures. There are a lot of different subtypes: private and public education, elementary and primary education, adult education, higher, tertiary and academic education, special education, etc. In conventional classrooms, educators attempt to enhance instructions by monitoring student's learning processes and analyzing their performances using paper records and observation. They can also use information about student attendance, course information, curriculum goals, and individualized plan data. In traditional educational environments, classification has been applied for many tasks, some examples are: predicting student success using multiple regression equations in an introductory programming course, with hopes of better counseling for students [62]; explaining and predicting the student final grade predicted by a neural network [63]; predicting performance from test scores using neural networks [24]; in a gifted education program selecting the weaker students for remedial classes by using association rules [64]; predicting student outcomes using discriminant function analysis and identifying variables to predict success in specific courses [15]; predicting academic performance to determine what registration factors determine academic success in universities [65]; using several data mining techniques, before the first session of exams, to classify students into three groups according to their probability of success, in order to identify the students who require aid, and thus propose specific remedial actions in time [27]; predicting a student's academic performance using neural networks, decision trees and linear regression [66].
- Web-based education is a form of distance education delivered over the Internet. Today, there are a lot of terms used to refer to web-based education such as e-learning, e-training, online instruction, web-based learning, web-based training, etc. All these systems normally record the student's accesses in web logs that provide a raw tracking of the learners' navigation on the site. However, there are two types of web-based educational

Author Proof

systems: well-known learning management systems and adaptive and intelligent web-based educational systems. Learning management systems accumulate a great amount of log data on students' activities in databases and usually have built-in student monitoring features. Adaptive and intelligent web-based educational systems (AIWBES) are the result of a joint evolution of intelligent tutoring systems (ITS) and adaptive hypermedia systems (AHS). AIWBES have data from students in the domain model, student model and user interaction log files. In web-based education, classification has been applied to many tasks, for example: discovering potential student groups with similar characteristics and reactions to a specific pedagogical strategy by applying decision tree and data cube technology [19]; predicting student performance and the number of errors a student will make through the use of Neural Networks [67]; predicting student performance and their final grades using genetic algorithms [38]; detecting student misuse or students playing using a Latent Response Model [20]; predicting student performance as well as assessing the relevance of attributes involved, using different machine learning techniques [32]; integrating prior problem solving and knowledge sharing histories of a group to predict future group performance using a combination of machine learning probabilistic tools [68]; grouping hint-driven students or failure-driven ones and finding common student misconceptions [21]; predicting course success using different machine learning methods [69]; classifying student performance according to their accumulated knowledge in an e-learning platform using C4.5 algorithm [70]; finding which usage features are best at predicting online student marks and explaining mark prediction in the form of simple and interpretable rules using Fuzzy Inductive Reasoning [71]; engagement prediction using Bayesian Networks and identifying learners with low motivation, as well as finding remedial actions in order to lower drop-out rates [72].

MOODLE MINING TOOL

Nowadays, there are a variety of general data mining tools, both commercial—such as DBMiner, SPSS Clementine, SAS Enterprise Miner, IBM Intelligent Miner—as well as open sources, like Weka [73], RapidMiner [74], and KEEL [75]. However, all these tools are not specifically designed for pedagogical/educational purposes and it is cumbersome for an educator to use these tools which are normally designed more for power and flexibility than for simplicity. In order to resolve this problem, we propose to use an educational data mining tool integrated into the educational environment like other traditional author tools (for creating and designing courses and their contents). In this way all data mining processes could be carried out by the instructor himself in a single application, and the feedback and results obtained can be applied directly to the educational environment. In fact, we have developed a specific Moodle data mining tool oriented for the use of two different types of users:

- Instructors and courseware authors/administrators. They are normally beginners not yet expert in data mining. In order to facilitate the use of data mining to this type of users, we provide both default algorithms (i.e., the instructor only has to select a method or task and then automatically a default algorithm is proposed for use) and default parameters (i.e., the instructor does not have to provide appropriate values for algorithm parameters because the tool also proposes default parameters). In this way, we try to simplify the configuration and execution of data mining algorithms for non-expert users.
- Data mining and educational researchers. They are normally expert users of data mining, so they apply several algorithms to do a task/method and also modify the default parameters in order to try to obtain better performances. And if they are expert at programming, they can even also add completely new algorithms (implemented in Java language) to the tool by using a specific Application Programming Interface (API) that we provide for these special users who want to develop their own algorithms.

We have chosen Moodle [2] because it is one of the most frequently used LMS and enables the creation of powerful, flexible and engaging online courses and experiences. It is important to point out that Moodle keeps a record of all the activities that students perform in detailed logs in a database.

Our Moodle mining tool has a simple interface (see Fig. 1) to facilitate the execution of data mining algorithms. We have integrated this tool into the Moodle environment itself. In this way, users can both create/maintain courses and carry out all data mining processing with the same interface. Likewise, they can directly apply feedback and the results obtained by data mining back into Moodle courses. This tool has been implemented in Java using the KEEL framework [76] which is an open source framework for building data mining models including classification (all the previously described algorithms in the Background Section), regression, clustering, pattern mining, and so on.

As seen in Figure 1, Moodle, like most LMSs, records all the students' usage information not only in log files but also directly in a database. In fact, the Moodle database has about interrelated 145 tables. But all this information is not required and so it is also necessary to convert the useful data to the required format used by our data mining. For this reason, Moodle data has to be pre-processed to convert it into KEEL data files. Then, data mining algorithms (classification algorithms in our case) can be executed to discover hidden knowledge within the data of interest for the instructor (classification models in our case). Finally, the results or models obtained are saved into result files that must be interpreted by the teacher to make decisions about the students and Moodle course activities. So, this mining process consists of the same three steps of a general knowledge discovery process: pre-processing, data mining, and post-processing. Next there is a description in more detail about how to carry out these steps using our mining tool.

- Pre-processing: First of all, users create data files starting from the Moodle database. The pre-processing task has to be done when instructors or courseware authors have enough information available about the students, and normally this happens after the end of a semester or after the end of a course. Then, they can choose between creating a data file from one particular Moodle table or creating a summary data file from different Moodle tables. Summarization files integrate the most important student information since student and interaction data are spread over several Moodle tables (see Table 1). The process for creating a summarization file has three steps (see Fig. 2). In the first step, instructors choose which specific courses (from among all the Moodle courses) merit using mining. In the second step, users have to add the marks obtained in the final exam by the students in the course selected. And in the third step, the users have to select which specific attributes they want to use in the summarization file, and they must name the data file. This data file is a text file in KEEL format [76]. Our mining tool also allows them to split data files into training and test data files.
- Data mining: Next, users have to select one of the available mining algorithms, the data file and a location for the output directory (where the model obtained and the results will be saved). Our mining tool has a lot of algorithms grouped by methods/tasks: statistics, clustering, association and classification (all the algo-

Author Proof

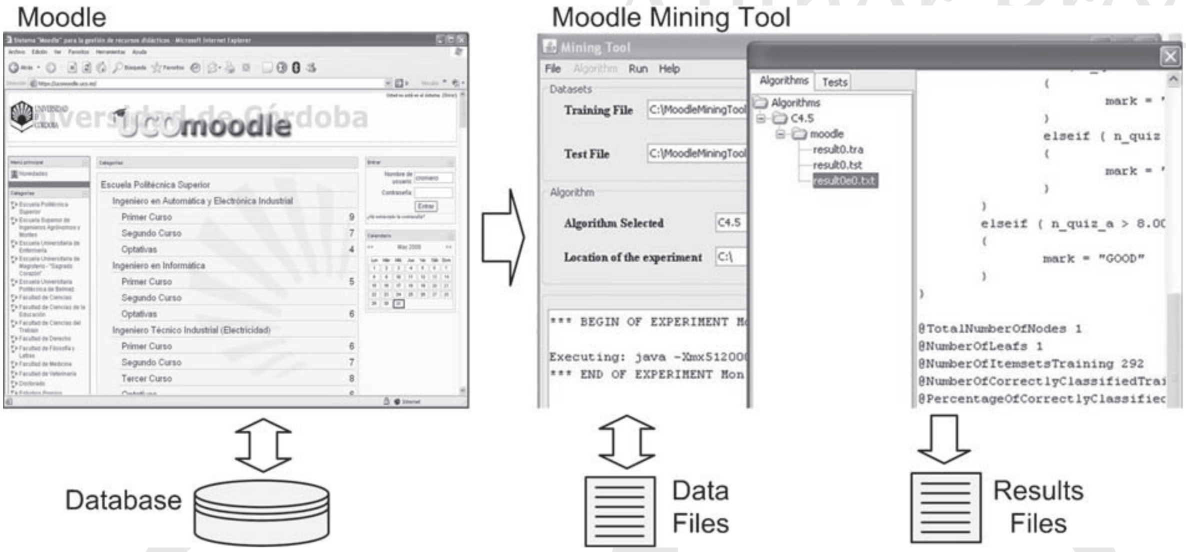

Figure 1 Moodle mining tool working.

Table 1 Attributes Used by Each Student in Summary File

Name	Type	Description
Course	Input attribute	Identification number of the course
n_assigment	Input attribute	Number of assignments done
n_quiz_a	Input attribute	Number of quizzes passed
n_quiz_s	Input attribute	Number of quizzes failed
n_posts	Input attribute	Number of messages sent to the forum
n_read	Input attribute	Number or messages read on the forum
total_time_assignment	Input attribute	Total time used on assignments
total_time_quiz	Input attribute	Total time used on quizzes
total_time_forum	Input attribute	Total time used on forum
Mark	Class	Final mark the student obtained in the course

rithms previously described in the Background Section). So, the user has to first select the data mining method and then one of the available algorithms. Our tool recommends one algorithm by default for each method in order to facilitate the selection of algorithms for beginners. For example, the default algorithm for doing classification is C4.5 [36]. However, the user can change the current selection to any other algorithm in order to test its performance with current data. For example, Figure 1 shows the execution of the C4.5 algorithm over a summary file and the decision tree obtained. The resulting files (.tra and .test files that contain partial classification results, and the .txt file that contains the obtained model) appear in a new window (see Fig. 1 on the right hand side). In this case, the model obtained is a decision tree that uses IF-THEN rules and also shows a summary with the number of nodes and leaves on the tree, the number and percentage of correctly and incorrectly classified instances, etc.

- Post-Processing: Finally, users can use the classification model obtained for decision making about the suitability of Moo-

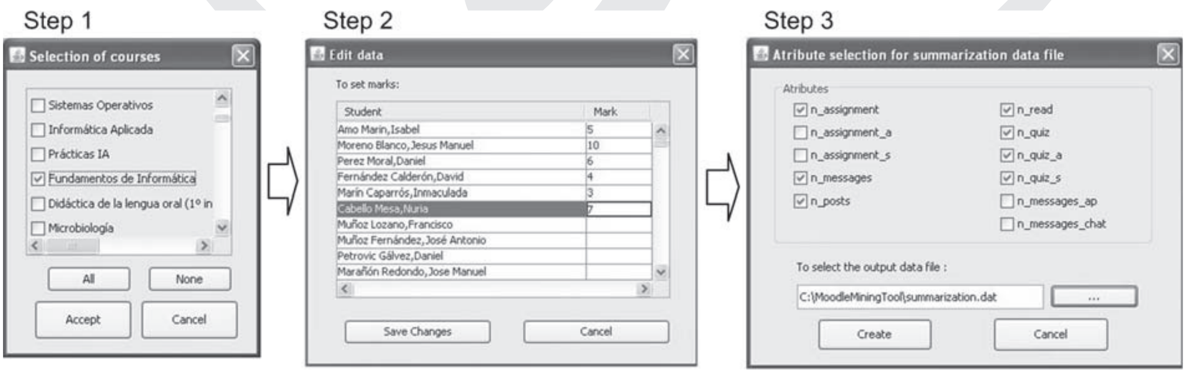

Figure 2 Steps for pre-processing Moodle data and creating a summarization file.

Author Proof

dle activities for each specific course, to classify new students depending on the course usage data, to detect students with problems, etc.

EXPERIMENTAL RESULTS

We have carried out two experiments in order to evaluate the performance and usefulness of different classification algorithms to predict students' final marks based on information in the students' usage data in LMS. Our objective is to classify students with similar final marks into different groups depending on the activities carried out in a web-based course. We have chosen the data of 438 Cordoba University students in 7 Moodle engineering courses concerning: health and safety at work, projects, engineering firm, programming for engineering, computer science fundamentals, applied computer science, and scientific programming. Starting from these courses and using our Moodle mining tool, a summary table (see Table 1) has been created which integrates the most important information for our objective (Moodle activities and the final marks obtained in the course). Table 1 summarizes row by row all the activities done by each student in the course (input variables or attributes) and the final mark obtained in this course (class or output attribute). Although there are many factors that can affect effectiveness in e-learning [77], this study is based on the information gathered about the following three online activities:

- Quizzes are a useful tool for students to test their level of knowledge and review each of the subjects studied [78]. They are great for giving students rapid feedback on their performance and for gauging their comprehension of materials. In our study, both passed and failed quizzes are taken into consideration.
- Assignments are a tool for collecting student work [79]. It is an easy way to allow students to upload digital content for grading. They can be asked to submit essays, spreadsheets, presentations, web pages, photographs, or small audio or video clips.
- Forums are a powerful communication tool [80]. They allow educators and students to communicate with each other at any time, from anywhere with an Internet connection. Forums create many opportunities to replicate the conversations you have in class, to formulate discussions between groups of students or to bring the best ideas and questions from the forum into your classroom. In our study, we use both the messages sent and read to/on the forum.

The whole summarization table has been divided into 10 pairs of training and test data files. In this way, each algorithm can be evaluated using stratified 10-fold cross-validation [81]. That is, the dataset is randomly divided into 10 disjointed subsets of equal size in a stratified way (maintaining the original class distribution). The algorithm is executed 10 times and in each repetition, one of the 10 subsets is used as the test set and the other 9 subsets are combined to form the training set. Finally, the mean accuracy is calculated.

The first experiment compares all the classification algorithms (described in the Background Section) using three different datasets: all the available data, filtered data by rows and filtered data by columns.

- The first dataset (all available data) consists of 438 instances/students with 9 input attributes for each instance, that is, the entire student data obtained from Moodle in the summary table.
- The second dataset (filtered data by rows) consists of 135 instances/students with 9 input attributes for each instance. In this case, these specific students have been selected/filtered by hand (135) because they are the only ones who completed all the Moodle activities proposed in each course. Our objective was to clean incomplete data.
- The third dataset (filtered data by columns) consists of 438 instances/students with only 4 input attributes (course, n_assigment, n_quiz_a, and total_time_quiz) for each instance. In this case, these 4 specific input attributes have been selected because they had been previously chosen by such attribute selection algorithms as: CfsSubsetEval, FilteredSubsetEval and ChiSquaredAttributeEval, all available in Weka [82] software. Attribute selection algorithms try to remove irrelevant attributes from data. In many practical situations there are far too many attributes for learning schemes to handle, and some of them can be irrelevant or redundant. Our objective was to reduce the dimensionality of the data.

So, this first experiment was to test if better classification accuracy could be obtained using filtered data instead of the original data. In order to do so, the three previously described sets of 10-fold data files were used; one execution was carried out with all deterministic algorithms and 5 executions with nondeterministic ones.

Table 2 shows the global percentage of the accuracy rate with test data (the averages of 10 executions). The global percentage of those correctly classified (global PCC) shows the accuracy of the classifiers.

Table 2 shows that a great number of algorithms (14 out of 21) obtain their highest accuracy values using original data, and all the rest of the algorithms (7 out of 21) obtain them using the data filtered by columns. This can be due to the fact that some algorithms themselves try to select attributes appropriately and ignore irrelevant and redundant ones, while others do not [70]. The best algorithms (more than 65% global PCC) with original data are CART, GAP, GGP, and NNEP. The best algorithms (over 50% global PCC) using filtered data by row are PolQuadraticLMS, KNN, and XCS. The best algorithms (over 64% global PCC) with filtered data by column are CART, SAP, and GAP.

The conclusion of this first experiment is that the best accuracy results (over 65% global PCC) are obtained when all available data (all students and all attributes) are taken into consideration versus filtering. In fact, when filtering by row, 303 students that had not done some activities were eliminated from the original data. And, it is logical to obtain worse accuracy when less data has been used. On the other hand, when filtering by attribute selection, only 4 of the 10 available input attributes have been used. And in some algorithms, the effect of decreasing the number of attributes was a decrease in classification accuracy. So, in our case we recommend using all available data.

The second experiment again compared all the classification algorithms (described in the Background Section) using all the available data but now applying two pre-processing tasks to the data: discretization and rebalancing. Thus, three different datasets are going to be used again: the full original numerical data, categorical data (discretizing the original data) and rebalanced data (rebalancing the original data).

- All the numerical values of the summary table have been discretized into a new summarization table. Discretization divides the numerical data into categorical classes that are easier for the teacher to understand. It consists of transforming continuous attributes into discrete attributes that can be treated as categorical attributes. There are different discretization methods. Concretely, the manual method (where cut-off points have to

Author Proof

Table 2 Classification Results With All Available Data and Filtered Data (Global Percentage of Correctly Classified)

Method	Algorithm	All available data	Filtered data by row	Filtered data by column
Statistical Classifier	ADLinear	59.82	28.13	63.49
Statistical Classifier	PolQuadraticLMS	64.30	50.93	63.03
Statistical Classifier	Kernel	54.79	33.18	58.00
Statistical Classifier	KNN	59.38	49.12	60.51
Decision Tree	C45	64.61	45.16	63.01
Decision Tree	CART	65.77	40.71	64.15
Rule Induction	AprioriC	60.04	35.60	59.82
Rule Induction	CN2	64.17	39.28	63.46
Rule Induction	Corcoran	62.55	36.86	58.91
Rule Induction	XCS	62.80	47.30	62.34
Rule Induction	GGP	65.51	63.44	43.86
Rule Induction	SIA	57.98	44.94	61.19
Fuzzy Rule Learning	MaxLogitBoost	64.85	43.68	63.23
Fuzzy Rule Learning	SAP	63.46	38.62	64.16
Fuzzy Rule Learning	GAP	65.99	43.46	64.15
Fuzzy Rule Learning	GP	63.69	36.20	63.48
Fuzzy Rule Learning	Chi	57.78	34.06	58.91
Neural Networks	NNEP	65.95	44.50	63.49
Neural Networks	RBFN	55.96	26.92	54.13
Neural Networks	GANN	60.28	42.03	61.43
Neural Networks	MLPerceptron	63.91	45.93	62.34

be specified) has been applied to the mark attribute, where four intervals and labels have been used (FAIL: if value is <5; PASS: if value is >5 and <7; GOOD: if value is >7 and <9; and EXCELLENT: if value is >9). To all the other attributes, the equal-width method has been applied [83]; here three intervals have been used (each one depending on the range of each attribute) and the same three labels (LOW, MEDIUM, and HIGH).

- Another problem taken into consideration has been learning from imbalanced data. Data is said to be imbalanced when some classes differ significantly from others with respect to the number of instances available. The problem with imbalanced data arises because learning algorithms tend to overlook less frequent classes (minority classes), paying attention just to the most frequent ones (majority classes). As a result, the classifier obtained will not be able to correctly classify data instances corresponding to poorly represented classes. Our data presents a clear imbalance since its distribution is: EXCELLENT 3.89%, GOOD 14.15%, PASS 22.15%, FAIL 59.81%. One of the most frequent methods used to learn from imbalanced data consists of re-sampling the data, either by over-sampling the minority classes or under-sampling the majority ones, until every class is equally represented [84]. When we deal with balanced data, the quality of the induced classifier is usually measured in terms of classification accuracy, defined as the fraction of correctly classified examples. However accuracy is known to be unsuitable to measure classification performance with imbalanced data. An evaluation measure well suited to imbalanced data is the geometric mean of accuracies per class (g-mean), defined as $g - \text{mean} = \sqrt[n]{\prod_{i=1}^{n} \frac{\text{hits}_i}{\text{instances}_i}}$, where n is the number of classes, hits_i is the number of instances of class i correctly classified and instances_i is the number of instances of class i. Our work has used random over-sampling, a technique consisting of copying randomly chosen instances from minority classes in the dataset until all classes have the same number of instances; the geometric mean is used to measure the quality of the classifiers induced.

The objective of this second experiment is to test if better classification accuracy can be obtained using discretized or rebalanced data instead of the original numerical data. In order to do so, the three previously described sets of 10-fold data files have been used; one execution has been carried out with all the deterministic algorithms and 5 executions with the nondeterministic ones.

Table 3 shows the global percentage of the accuracy rate and geometric mean with test data (the averages of 10 executions). With respect to accuracy about half of the algorithms (12 out of 21) obtain their highest values using the original numerical data, and the other algorithms (13 out of 21) obtain it using the categorical data. This can be due to the nature and implementation of each algorithm which might be more appropriate for using numerical or categorical data. As seen above, it is easier to obtain a high accuracy rate when data are imbalanced, but when all the classes have the same number of instances it becomes more difficult to achieve a good classification rate. The best algorithms (over 65% global PCC) with original data (numerical) are CART, GAP, GGP, and NNEP. The best algorithms (over 65% global PCC) using categorical data are the two decision tree algorithms: CART and C4.5. The best algorithms (over 60% global PCC) with balanced data are Corcoran, XCS, AprioriC, and MaxLogicBoost.

The geometric mean tells us about the effect of rebalancing on the performance of the classifiers obtained, since the geometric mean offers a better view of the classification performance in each of the classes. Table 2 shows that the behaviour depends to a great extent on the learning algorithm used. There are some algorithms which are not affected by rebalancing (Kernel, KNN, AprioriC, and Corcoran); the two decision tree methods (CART and C4.5) give worse results with rebalanced data (C4.5) but most of the algorithms (all the rest, 17 out of 25) obtain better results with rebalanced data. Thus we can see that the rebalancing of data is generally beneficial for most of the algorithms, and also that many algorithms obtain a value of 0 in the geometric mean. This is because some algorithms do not classify any of the students correctly into a specific group. It is interesting to see that it only happens to the group of EXCELLENT students (EXCELLENT students are incorrectly classified as GOOD and PASS students).

The conclusion of the second experiment is that the best accuracy results (more than 65% global PCC) are obtained both with

Author Proof

Table 3 Classification Results With Numerical, Categorical, and Rebalanced Data (Global Percentage of Correctly Classified/Geometric Mean)

Method	Algorithm	Numerical data	Categorical data	Rebalanced data
Statistical Classifier	ADLinear	59.82/0.00	61.66/0.00	59.82/0.00
Statistical Classifier	PolQuadraticLMS	64.30/15.92	63.94/18.23	54.33/26.23
Statistical Classifier	Kernel	54.79/0.00	56.44/0.00	54.34/0.00
Statistical Classifier	KNN	59.38/10.15	59.82/7.72	54.34/10.21
Decision Tree	C45	64.61/41.42	65.29/18.10	53.39/9.37
Decision Tree	CART	65.77/39.25	65.86/24.54	47.51/34.65
Rule Induction	AprioriC	60.04/0.00	59.82/0.00	61.64/0.00
Rule Induction	CN2	64.17/0.00	63.47/3.52	50.24/15.16
Rule Induction	Corcoran	62.55/0.00	64.17/0.00	61.42/0.00
Rule Induction	XCS	62.80/0.00	62.57/0.00	60.04/23.23
Rule Induction	GGP	65.51/1.35	64.97/1.16	52.91/12.63
Rule Induction	SIA	57.98/0.00	60.53/0.00	56.61/15.41
Fuzzy Rule Learning	MaxLogitBoost	64.85/0.00	61.65/0.00	62.11/8.83
Fuzzy Rule Learning	SAP	63.46/0.00	64.40/0.00	47.23/3.20
Fuzzy Rule Learning	GAP	65.99/0.00	63.02/0.00	52.95/26.65
Fuzzy Rule Learning	GP	63.69/0.00	63.03/0.00	53.19/11.97
Fuzzy Rule Learning	Chi	57.78/10.26	60.24/0.00	41.11/14.32
Neural Networks	NNEP	65.95/0.00	63.49/0.00	54.55/12.70
Neural Networks	RBFN	55.96/3.23	54.60/0.00	37.16/4.00
Neural Networks	GANN	60.28/0.00	61.90/4.82	53.43/17.33
Neural Networks	MLPerceptron	63.91/9.65	61.88/4.59	53.21/17.16

numerical and categorical data, and that the best values of the geometric mean are obtained with rebalanced data.

On the other hand, in our educational problem it is also very important for the classification model obtained to be user friendly, so that instructors or course administrators can make decisions about some students and the on-line course to improve the students' learning. In general, models obtained using categorical data are more comprehensible than when using numerical data because categorical values are easier for a teacher to interpret than precise magnitudes and ranges. Nonetheless, some models are more interpretable than others:

- Decision trees are considered to be easily understood models because a reasoning process can be given for each conclusion. It is a so-called white-box model that allows an interpretation of model parameters. That is, it can provide an explanation for the classification result. In fact, a decision tree can be directly transformed into a set of IF-THEN rules that are one of the most popular forms of knowledge representation, due to their simplicity and comprehensibility. So, these algorithms are simple for instructors to understand and interpret. An example of part of a decision tree obtained by C4.5 is shown below. With respect to its applicability and usefulness, the decision tree obtained can be used for predicting/classifying new students depending on the activities done and decisions made because the attributes and values that are used for classification are also shown in a tree form. For example, this decision tree shows the instructors that students who pass fewer than 7 quizzes can pass the final exam if they spend over 25 min (1494 s) on the forums of courses, or they fail the final exam if they spend under 25 min in forums, and also students who pass more than seven quizzes will then pass the final exam if they do under five assignments or they will obtain an excellent marks if they do more than five assignments.
  IF (n_quiz_a < = 7) THEN
  {
  IF (total_time_forum <= 1494) THEN {mark = FAIL}
  ELSEIF (total_time_forum > 1494) THEN {mark = PASS}
  }
  ELSEIF (n_quiz_a > 7) THEN
  {
  IF (n_assignment < = 10) THEN {mark = PASS}
  ELSEIF (n_assignment > 10) THEN {mark = EXCELLENT}
  }
  ELSEIF...
- Rule induction algorithms are normally also considered to produce comprehensible models. It is also a so-called white-box model that discovers a set of IF-THEN classification rules which have a high-level knowledge representation and can be used directly for decision making. Some specific algorithms such as GGP have a higher expressive power that allows the user to determine the specific format of the rules, such as their number of conditions, operators, etc. using a grammar. Some examples of rules obtained by the GGP algorithm are shown below. Regarding its applicability and usefulness, these rules can be used to detect both good students (students who obtain pass, good and excellent marks) and those with problems (students who obtain fail marks), and which activities are more related to good and bad marks. For example, the first rule shows that if students do fewer than six assignments, then they fail in the final exam. The second rule shows that if students do more than 10 assignments and they read more than nine messages on the forum, then they obtain an excellent mark in the final exam. The third rule shows that only students in course 29 who do not pass any quizzes fail in the final exam. The fourth rule shows that only students in course 110 who pass more than seven quizzes obtain good marks.
  IF n_assignment < 6 THEN mark = FAIL
  IF n_assignment > 10 AND n_read > 9 THEN mark = EXCELLENT
  IF course = 29 AND n_quiz_a = 0 THEN mark = FAIL
  IF course = 110 AND n_quiz_a > 7 THEN mark = GOOD
- Fuzzy rule algorithms are also white-box models that obtain a special type of IF-THEN rules. These rules use linguistic terms that make them more comprehensible/interpretable by humans. So, this type of rules is very intuitive and easily understood by problem-domain experts like teachers. Some examples of fuzzy rules obtained by the MaxLogitBoost algorithm are shown below. Regarding its applicability and usefulness, these fuzzy rules can also be used to detect both good students and students with problems, as well as those activities related to obtaining good or bad marks. For example, the first rule shows that if

Author Proof

students pass a very low number of quizzes, then they fail in the final exam. The second is the opposite rule that shows that if students pass a very high number of rules, then they obtain excellent marks. The third rule shows that only students in course 110 who do many assignments and send a high number of messages to the forum obtain a good mark. The fourth rule shows that only students in course 29 who read a very low number of messages on the forum fail in the final exam.
IF n_quiz_a = Very low THEN mark = FAIL
IF n_quiz_a = Very high THEN mark = EXCELLENT
IF course = 110 AND n_assignment = High AND n_posts = High THEN mark = GOOD
IF course = 29 AND n_read = Very low THEN mark = FAIL

- Statistical methods and neural networks are deemed to be less suitable for data mining purposes. This rejection is due to their lack of comprehensibility. Knowledge models obtained under these paradigms are usually considered to be black-box mechanisms, able to attain very good accuracy rates but very difficult for people to understand. Nevertheless, their discriminating power is often significantly better than that of white-box models, which may explain their popularity in domains where classification performance is more important than model interpretation. However, some algorithms of this type do obtain models people can understand. For example, ADLinear, PolQuadraticLMS, Kernel, and NNEP algorithms obtain functions that express possibly strong interactions among the variables. An example of discriminant function obtained by NNEP algorithm is shown below, where $x_1, x_2, \ldots, x_9$ are the input variables or attributes: course, n_assigment, ..., total_time_forum and F1 is the predicted output value. Regarding its applicability and usefulness, this function can be used to predict the classification of new students depending on the activities done. In fact, for each new student, it calculates its output value of the function (predicted mark) using the input attribute values of each new student.
$F1 = -7.44 * (x1^{-}0.58 * x2^{0}.90 * x5^{-}1.13 * x6^{1}.96)$
$-1.50 * (x1^{-}2.76 * x3^{2}.05 * x5^{2}.22 * x6^{0}.90 * x7^{1}.72 * x9^{-}1.01)$
$-4.48 * (x1^{1}.95 * x2^{1}.47 * x4^{3}.43 * x9^{0.16})$

Finally, in our educational problem the final objective of using a classification model is to show the instructor interesting information about student classification (prediction of marks) depending on the usage of Moodle courses. Then, the instructor can use this discovered knowledge for decision making and for classifying new students. For example, some of the rules discovered show that the number of quizzes passed in Moodle was the main determiner of the final marks, but there are some others that could help the teacher to decide whether to promote the use of some activities to obtain higher marks, or on the contrary, to decide to eliminate some activities because they are related to low marks. It could also be possible for the teacher to detect new students with learning problems in time to remedy them (students classified as FAIL). The teacher could use the classification model in order to classify new students and detect in time if they will have learning problems (students classified as FAIL) or not (students classified as GOOD or EXCELLENT).

CONCLUSIONS

In this paper we have compared the performance and usefulness of different data mining techniques for classifying university students through the use of different datasets from Moodle courses. We have also developed a specific mining tool integrated into the Moodle system in order to facilitate the execution of data mining algorithms both for non-expert users, such as instructors, and expert users, such as data mining researchers.

Our experiments show, on the one hand, that in general there is not one single algorithm that obtains the best classification accuracy in all cases (with all datasets). On the other hand, some pre-processing task like filtering, discretization or rebalanzing can be very important to obtain better or worse results. In fact, most of the algorithms are seen to improve their classification performance when using all the available data (without filtering); some of them do when pre-processing tasks like discretization and rebalancing data are applied; others do not at all. However, we have seen that the accuracy obtained is not very high (in the range of 65%) and it shows that it is a very difficult task to predict the students' final marks starting from their web usage data. One first possible solution can be to try to fix/set more appropriate parameters by doing exhaustive experimentation which varies and tests progressively the value of each parameter's ranges of values. But, we have not obtained better classification accuracy in this way. A different and promising solution can be to use more and different students' attributes as input attributes. For example, we could use not only online information about students (as we have done in this paper) but also offline information such as classroom attendance, punctuality, participation, attention, predisposition, etc. But, it is important to notice that all this offline information is not obtained automatically and as easily as we have obtained the online information provided by using Moodle. Thus instructors will have to provide the values of these new attributes by themselves which could be a difficult chore.

We have also shown that, in our problem, a good classifier model has to be both accurate and comprehensible for instructors. So, from among all the proposed methods, we recommend using decision trees, rule induction and fuzzy rule algorithms because they are white-box models that provide comprehensible results, allow an interpretation to be made of the model obtained and can be used for making decisions.

In future experiments, we would like to can measure the level of comprehensibility of each classification model in order to select the best algorithm. But due to it is a subjective measure, we can use for example a group of experts in data mining for evaluating the interpretability and comprehensibility of each specific algorithms. We also want to do more experiments using data with more information about the students (i.e., profile and curriculum) and to use more amounts of data (data from other courses and years). This could measure how the quantity and quality of the data can affect the performance of the algorithms. Finally, the tool should also be tested by teachers in real pedagogical situations to prove its acceptability.

REFERENCES

[1] P. Brusilovsky and C. Peylo, Adaptive and intelligent web-based educational systems, Int J Artif Intell Educ 13 (2003), 156–169.
[2] J. Cole, Using Moodle. O'Reilly, 2005.
[3] BlackBoard. www.blackboard.com [Accessed 23 March 2010].
[4] TopClass. www.topclass.nl [Accessed 23 March 2010].
[5] Ilias. www.ilias.de [Accessed 23 March 2010].
[6] Claroline. www.claroline.net [Accessed 23 March 2010].
[7] W. H. Rice, Moodle E-learning course development. A complete guide to successful learning using moodle[Q3], Packt Publ (2006).
[8] J. Mostow and J. Beck, Some useful tactics to modify, map and mine data from intelligent tutors, Nat Lang Eng 12 (2006), 195–208.

Author Proof

[9] J. Mostow, J. Beck, H. Cen, A. Cuneo, E. Gouvea, and C. Heiner, An educational data mining tool to browse tutor-student interactions: Time will tell, In: Proceedings of the Workshop on Educational Data Mining, 2005, pp 15–22.

[10] L. Dringus and T. Ellis, Using data mining as a strategy for assessing asynchronous discussion forums, Comput Educ J 45 (2005), 141–160.

[11] M. E. Zorrilla, E. Menasalvas, D. Marin, E. Mora, and J. Segovia, Web usage mining project for improving web-based learning sites, In Web Mining Workshop (2005), 1–22[Q4].

[12] O. Zaïane and J. Luo, Web usage mining for a better web-based learning environment, In: Proceedings of the Conference on Advanced Technology for Education, 2001, pp 60–64.

[13] C. Romero and S. Ventura, Educational data mining: A survey from 1995 to 2005, Expert Syst Appl 33 (2007), 135–146.

[14] C. Romero and S. Ventura, Educational data mining: A review of the state-of-the-art, IEEE Trans Syst Man Cybern C in press[Q5].

[15] W. Klosgen and J. Zytkow, Handbook of data mining and knowledge discovery, Oxford University Press[Q6], 2002.

[16] F. Castro, A. Vellido, A. Nebot, and F. Mugica, Applying data mining techniques to e-learning problems, In: L.C. Jain, R.A. Tedman, D.K. Tedman, (Eds.), Evolution of teaching and learning paradigms in intelligent environment. Studies in computational intelligence, Vol. 62, 2007, Springer[Q7].

[17] F. J. Martínez, C. Hervás, P. A. Gutiérrez, A. C. Martínez, and S. Ventura, Evolutionary Product-Unit Neural Networks for Classification, In: Conference on Intelligent Data Engineering and Automated Learning, 2006, pp 1320–1328.

[18] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification, Wiley Interscience[Q8], 2000.

[19] G. Chen, C. Liu, K. Ou, and B. Liu, Discovering decision knowledge from web log portfolio for managing classroom processes by applying decision tree and data cube technology, J Educ Comput Res 23 (2000), 305–332.

[20] R. Baker, A. Corbett, and K. Koedinger, Detecting student misuse of intelligent tutoring systems, Intelligent Tutoring Systems Alagoas (2004), pp 531–540.

[21] M. V. Yudelson, O. Medvedeva, E. Legowski, M. Castine, D. Jukic, and C. Rebecca, Mining student learning data to develop high level pedagogic strategy in a medical ITS, AAAI Workshop Educ Data Min[Q9](2006), 1–8.

[22] M. Cocea and S. Weibelzahl, Can log files analysis estimate learners' level of motivation? In: Proceedings of the Workshop Week Lernen—Wissensentdeckung—Adaptivität, Hildesheim, 2006, pp 32–35.

[23] A. Gandhe, S.-H. Yu, R. Mehra, and R. E. Smith, XCS for fusing multi-spectral data in automatic target recognition, Stud Comput Intell 125 (2008), 147–167.

[24] L. Fausett and W. Elwasif, Predicting performance from test scores using backpropagation and counterpropagation, In: IEEE Congress on Computational Intelligence, 1994, pp 3398–3402.

[25] D. Martínez, Predicting student outcomes using discriminant function analysis, In: Annual Meeting of the Research and Planning Group, California, 2001, pp 163–173.

[26] B. Minaei-Bidgoli and W. Punch, Using genetic algorithms for data mining optimization in an educational web-based system. Genetic and Evolutionary Computation Conference, Part II, 2003, pp 2252–2263.

[27] J. F. Superby, J. P. Vandamme, and N. Meskens, Determination of factors influencing the achievement of the first-year university students using data mining methods[Q10], Workshop Educ Data Min (2006), 37–44.

[28] S. B. Kotsiantis and P. E. Pintelas, Predicting Students Marks in Hellenic Open University, In: Conference on Advanced Learning Technologies. IEEE, 2005, pp 664–668.

[29] M. Delgado, E. Gibaja, M. C. Pegalajar, and O. Pérez, Predicting Students' Marks from Moodle Logs using Neural Network Models, Current Developments in Technology-Assisted; Education, Badajoz, Spain, 2006, pp 586–590.

[30] G. J. McLachlan, Discriminant analysis and statistical pattern recognition, John Wiley and Sons[Q11], 2004.

[31] J. S. Rustagi, Optimization techniques in statistics, Academic Press[Q12], 1994.

[32] S. B. Kotsiantis, C. J. Pierrakeas, and P. E. Pintelas, Predicting students' performance in distance learning using machine learning techniques, Appl Artif Intell 18 (2004), 411–426.

[33] J. P. Vert, Classification of biological sequences with kernel methods, In: Proceedings of the 8th International Colloquium in Grammatical Inference—Algorithms and Applications, 2006, pp 7–18.

[34] L. Gómez-Chova, G. Camps-Valls, L. Bruzzone, and J. Calpe-Maravilla, Mean map kernel methods for semisupervised cloud classification, IEEE Trans Geosci Remote Sens 48 (2010), 207–220.

[35] Q. Yu, A. Sorjamaa, Y. Miche, A. Lendasse, E. Séverin, A. Guillen, and F. Mateo, Optimal pruned K-nearest neighbors: OP-KNN—Application to financial modeling, In: Proceedings of the 8th International Conference on Hybrid Intelligent Systems, 2008, pp 764–769.

[36] J. R. Quinlan, C4.5: Programs for machine learning, Morgan Kaufman[Q13], 1993.

[37] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and regression trees, Chapman & Hall, New York, 1984.

[38] M. Lee, A study on[Q14]the application of Classification and Regression Trees (CART) via an example of sweet potato data, J Agr Assoc China 173 (1996), 61–68.

[39] Y. G. Ma, Y. X. Bi, H. Yan, L. N. Deng, W. F. Liang, B. Wang, and X. L. Zhang, The application of decision tree in the research of anemia among rural children under 3-year-old, Chinese J Prev Med 43 (2009), 434–437.

[40] P. Xu and S. Lin, Internet traffic classification using C4.5 decision tree, J Softw 20 (2009), 2692–2704.

[41] P. Clark and T. Niblett, The CN2 induction algorithm, Mach Learn 3 (1989), 261–283.

[42] V. Jovanoski and N. Lavrac, Classification Rule Learning with APRIORI-C, In: Progress in Artificial Intelligence, Knowledge Extraction, Multi-agent Systems, Logic Programming and Constraint Solving, 2001, pp 44–51.

[43] S. W. Wilson, Classifier fitness based on accuracy, Evolut Comput 3 (1995), 149-175.

[44] G. Venturini, SIA A Supervised Inductive Algorithm with Genetic Search for Learning Attributes based Concepts, In: Conference on Machine Learning, 1993, pp 280–296.

[45] A. L. Corcoran and S. Sen, Using real-valued genetic algorithms to evolve rule sets for classification, In: Conference on Evolutionary Computation, Orlando, 1994, pp 120–124.

[46] P. G. Espejo, S. Ventura, and F. Herrera, A survey on the application of genetic programming to classification, IEEE Trans Syst Man Cybern C 40 (2010), 121–144.

[47] J. A. Sanandrés-Ledesma, V. Maojo, J. Crespo, M. García-Remesal, and A. Gómez De La Cámara, A performance comparative analysis between rule-induction algorithms and clustering-based constructive rule-induction algorithms—Application to rheumatoid arthritis, In: Proceedings of the 5th International Symposium on Biological and Medical Data Analysis, 2004, pp 224–234.

[48] M. Panda and M. R. Patra, Ensembling rule based classifiers for detecting network intrusions, In: Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing, 2009, pp 19–22.

[49] F. Herrera, Genetic fuzzy systems: Taxonomy, current research trends and prospects, Evol Intell 1 (2008), 27–46.

[50] L. Sánchez and J. Otero, Boosting fuzzy rules in classification problems under single-winner inference, Int J Intell Syst 22 (2007), 1021–1034.

[51] L. Sánchez, I. Couso, and J. A. Corrales, Combining GP operators with SA search to evolve fuzzy rule based classifiers, Inform Sci 136 (2001), 175–191.

Author Proof

[52] Z. Chi, H. Yan, and T. Pham, Fuzzy algorithms: With applications to image processing and pattern recognition. World Scientific, Singapore, 1996.
[53] T. W. Chua and W. W. Tan, Genetically evolved fuzzy rule-based classifiers and application to automotive classification, In: Proceedings of the 7th International Conference on Simulated Evolution and Learning, 2008, pp 101–110.
[54] E. G. Mansoori, M. J. Zolghadri, and S. D. Katebi, Protein superfamily classification using fuzzy rule-based classifier, IEEE Trans Nanobiosci 8 (2009), 92–99.
[55] P. Dziwiński and D. Rutkowska, Algorithm for generating fuzzy rules for WWW document classification, In: Proceedings of the 8th International Conference on Artificial Intelligence and Soft Computing, 2006, pp 1111–1119.
[56] M. F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993), 525–533.
[57] D. S. Broomhead and D. Lowe, Multivariable functional interpolation and adaptative networks, Complex Syst 11 (1988), 321–355.
[58] X. Yao, Evolving artificial neural networks, Proc IEEE 87 (1999), 1423–1447.
[59] Z.-Y. Li, F.-Q. Zhang, and S.-T. Wan, RBFN Based on two levels iteration cluster algorithm and its application in generator fault diagnosis, In: Proceedings of the International Conference on Machine Learning and Cybernetics, Vol. 2, 2009, pp 1183–1187.
[60] F. Li, R. Wang, and L. Jiang, A HMM-RBFN hybrid classifier for surface electromyography signals classification, Chinese J Rehabil Med 21 (2006), 1016–1018.
[61] D. E. Guyer and X. Yang, Identifying apple defects by utilizing spectral imaging, fluorescence and genetic neural networks, In: Proceedings of the ASAE Annual International Meeting, Technical Papers: Engineering Solutions for a New Century, Vol. 1, 2000, pp 1159–1171.
[62] T. Hostetler, Predicting student success in an introductory programming course, ACM SIGCSE Bull 15 (1983), 40–43.
[63] T. Gedeon and H. Turner, Explaining student final grades predicted by a neural network, In: Proceedings of the International Conference on Neural Network, Detroit, USA, 1993, pp 609–612.
[64] Y. Ma, B. Liu, C. Wong, P. Yu, and S. Lee, Targeting the right students using data mining, In: Proceedings of the Conference on Knowledge Discovery and Data Mining. Boston, USA, 2000, pp 457–464.
[65] P. Golding and S. NcNamarah, Predicting academic performance in the school of computing & information technology, In: Proceedings of the ASEE/IEEE Frontiers in Education Conference, Indianapolis, USA, 2005, pp 16–20.
[66] Z. Ibrahim and D. Rusli, Predicting student's academic performance: Comparing artificial neural networks, decision tree and linear regression. Annual SAS Malaysia Forum. Kuala Lumpur, Malaysia, 2007, pp 1–6.
[67] T. Wang and A. Mitrovic, Using neural networks to predict student's performance, In: Proceedings of the International Conference on Computers in Education. Washington DC, USA, 2002, pp 969–974.
[68] R. Stevens, A. Soller, A. Giordani, L. Gerosa, M. Cooper, and C. Cox, Developing a framework for integrating prior problem solving and knowledge sharing histories of a group to predict future group performance, In: International Conference on Collaborative Computing: Networking, Applications and Worksharing. San Jose, USA, 2005, pp 1–9.
[69] W. Hämäläinen and M. Vinni, Comparison of machine learning methods for intelligent tutoring systems, In: Conference Intelligent Tutoring Systems, Taiwan, 2006, pp 525–534.
[70] C. Mihaescu and D. Burdescu, Testing attribute selection algorithms for classification performance on real data, In: Proceedings of the IEEE Conference Intelligent Systems, London, UK, 2006, pp 581–586.
[71] T. Etchells, A. Nebot, A. Vellido, P. Lisboa, and F. Múgica, Learning what is important: Feature selection and rule extraction in a virtual course, European Symposium on Artificial Neural Networks, Bruges, Belgium, 2006, pp 401–406.
[72] M. Cocea and S. Weibelzahl, Eliciting motivation knowledge from log files towards motivation diagnosis for adaptive systems, In: Proc 11th Int Conference on User Modelling, Corfu, Greece, 2007, pp 197–206.
[73] Weka. www.cs.waikato.ac.nz/ml/weka [Accessed 23 March 2010].
[74] RapidMiner. rapid-i.com [Accessed 23 March 2010].
[75] KEEL. www.keel.es [Accessed 23 March 2010].
[76] J. Alcalá, L. Sánchez, S. García, M. del Jesus, et al[Q15]. KEEL a software tool to assess evolutionary algorithms to data mining problems, Soft Comput[Q16](2007).
[77] E. Martínez, Factors affecting effectiveness in e-learning: An analysis in production management courses, Comput Appl Eng Educ in press[Q17].
[78] C. Romero, S. Ventura, and P. de Bra, Using mobile and web-based computerized tests to evaluate university students, Comput Appl Eng Educ 17 (2009), 435–447.
[79] C. Romero, S. Ventura, and E. García, Data mining in course management systems: Moodle case study and tutorial, Comput Educ 51 (2008), 368–384.
[80] C. Romero, P. Gonzalez, S. Ventura, M. J. del Jesus, and F. Herrera, Evolutionary algorithms for subgroup discovery in e-learning: A practical application using Moodle data, 36 (2009), 1632–1644.
[81] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995, pp 1137–1143.
[82] I. H. Witten and E. Frank, Data mining. Practical machine learning tools and techniques with java implementations, Morgan Kaufmann[Q18], 1999.
[83] J. Dougherty, M. Kohavi, and M. Sahami, Supervised and Unsupervised Discretization of Continuous Features, In: Conference on Machine Learning, San Francisco, 1995, pp 194–202.
[84] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, Strategies for learning in class imbalance problems, Pattern Recogn 36 (2003), 849–851.

Q1: Please check the suitability of the short title on the odd-numbered pages. It has been formatted to fit the journal's 45-character (including spaces) limit.
Q2: Please check the author names, affiliations and correspondence address.
Q3: Please provide the volume number and page range.
Q4: Please provide authors' biographies and photos.
Q5: Please update.
Q6: Please provide the publisher location.
Q7: Please provide the publisher location.
Q8: Please provide the publisher location.
Q9: Please provide the volume number.
Q10: Please provide the volume number.
Q11: Please provide the publisher location.
Q12: Please provide the publisher location.
Q13: Please provide the publisher location.
Q14: Please check the volume number.
Q15: Please provide the list of all the authors.
Q16: Please provide the volume number and page range.
Q17: Please update.
Q18: Please provide the publisher location.