# Data Mining Algorithms to Classify Students

Cristóbal Romero, Sebastián Ventura, Pedro G. Espejo and César Hervás
{cromero, sventura, pgonzalez, chervas}@uco.es
Computer Science Department, Córdoba University, Spain

Abstract. In this paper we compare different data mining methods and techniques for classifying students based on their Moodle usage data and the final marks obtained in their respective courses. We have developed a specific mining tool for making the configuration and execution of data mining techniques easier for instructors. We have used real data from seven Moodle courses with Cordoba University students. We have also applied discretization and rebalance preprocessing techniques on the original numerical data in order to verify if better classifier models are obtained. Finally, we claim that a classifier model appropriate for educational use has to be both accurate and comprehensible for instructors in order to be of use for decision making.

## 1 Introduction

The ability to predict/classify a student's performance is very important in web-based educational environments. A very promising arena to attain this objective is the use of Data Mining (DM) [26]. In fact, one of the most useful DM tasks in e-learning is classification. There are different educational objectives for using classification, such as: to discover potential student groups with similar characteristics and reactions to a particular pedagogical strategy [6], to detect students' misuse or game-playing [2], to group students who are hint-driven or failure-driven and find common misconceptions that students possess [34], to identify learners with low motivation and find remedial actions to lower drop-out rates [9], to predict/classify students when using intelligent tutoring systems [16], etc. And 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 (to classify in five classes: A, B, C, D and E or F) from test scores using neural networks [14]; predicting student academic success (classes that are successful or not) using discriminant function analysis [19]; classifying students using genetic algorithms to predict their final grade [21]; predicting a student's academic success (to classify as low, medium and high risk classes) using different data mining methods [30]; predicting a student's marks (pass and fail classes) using regression techniques in Hellenic Open University data [18] or using neural network models from Moodle logs [11].

In this paper we are going to compare different data mining techniques for classifying students based on both students' usage data in a web-based course and the final marks obtained in the course. We have also developed a specific Moodle data mining tool for making this task easier for instructors. The paper is arranged in the following way: Section 2 describes the background of the main classification methods and algorithms; section 3 describes the Moodle data mining tool; section 4 details the comparison of the classification techniques; finally, the conclusions and further research are outlined.

## 2 Background

Classification is one of the most frequently studied problems 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). There are different classification methods, such as:

- Statistical classification is a procedure in which 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 labelled items [23]. Some examples of statistical algorithms are linear discriminant analysis [21], least mean square quadratic [27], kernel [21] and k nearest neighbors [21].

- A decision tree is a set of conditions organized in a hierarchical structure [25]. It 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 to a set of classification rules. Some of the most well-known decision tree algorithms are C4.5 [25] and CART [4].

- Rule Induction is an area of machine learning in which IF-THEN production rules are extracted from a set of observations [11]. 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 [8], AprioriC [17], XCS [32], Supervised Inductive Algorithm (SIA) [31], a genetic algorithm using real-valued genes (Corcoran) [10] and a Grammar-based genetic programming algorithm (GGP) [15].

- Fuzzy rule induction applies fuzzy logic in order to interpret the underlying data linguistically [35]. 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 LogitBoost [23], MaxLogitBoost [29], AdaBoost [12], Grammar-based genetic Programming (GP) [28], a hybrid Grammar-based genetic Programming/genetic Algorithm method (GAP) [28], a hybrid Simulated Annealing/genetic Programming algorithm (SAP) [28] and an adaptation of the Wang-Mendel algorithm (Chi) [7].

- Neural Networks can also be used for rule induction. A neural network, also known as a parallel distributed processing network, is a computing paradigm that is loosely modeled after cortical structures in the brain. It consists of interconnected processing elements called nodes or neurons that work together to produce an output function. Examples of neural network algorithms are multilayer perceptron (with conjugate gradient-based training) [22], a radial basis function neural network (RBFN) [5], incremental RBFN [24], decremental RBFN [5], a hybrid Genetic Algorithm Neural Network (GANN) [33] and Neural Network Evolutionary Programming (NNEP) [20].

## 3 Moodle Data Mining Tool

We have developed a specific Moodle data mining tool oriented for use by on-line instructors. It has a simple interface (see Figure 1) to facilitate the execution of data mining techniques. We have integrated this tool into the Moodle environment itself. In this way, instructors can both create/maintain courses and carry out all data mining processing with the same interface. Likewise, they can directly apply feedback and results obtained by data mining back into Moodle courses. We have implemented this tool in Java using the KEEL framework [1] which is an open source framework for building data mining models including classification (all the previously described algorithms in Section 2), regression, clustering, pattern mining, and so on.

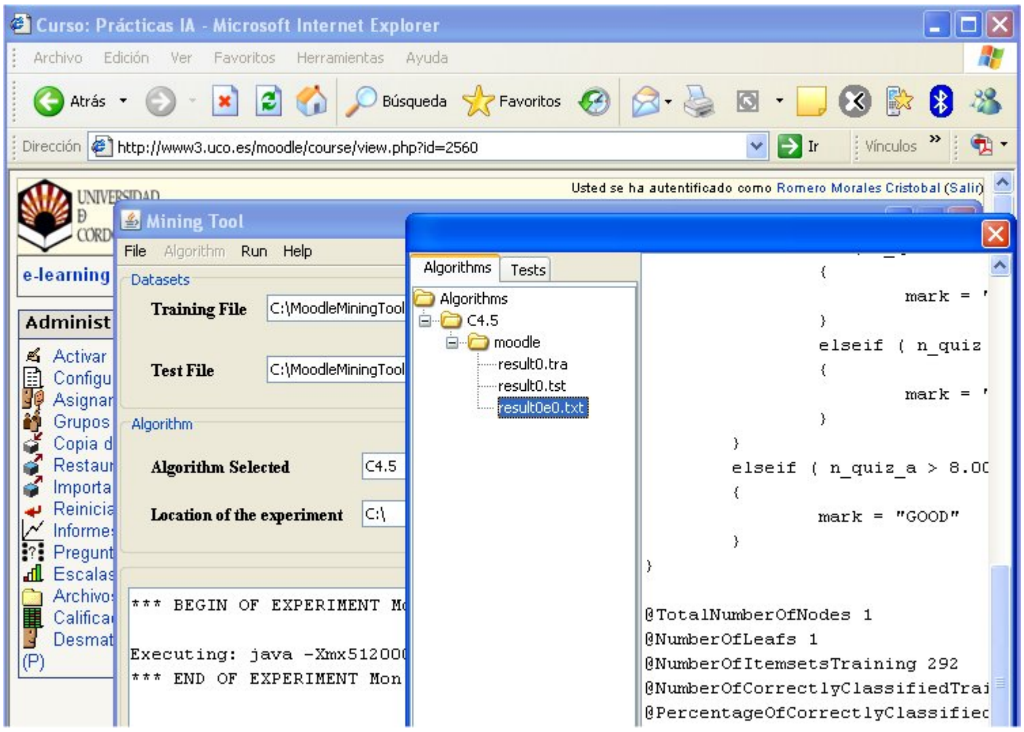


**Figure 1. Moodle Data Mining Tool executing C4.5 algorithm.**

In order to use it, first of all the instructors have to create training and test data files starting from the Moodle database. They can select one or several courses and one Moodle table (mdl_log, mdl_chat, mdl_forum, mdl_quiz, etc.) or create a summary table (see Table 1). Then, data files will be automatically preprocessed and created. Next, they only have to select one of the available mining algorithms and the location of the output directory. For example, in Figure 1, we show the execution of the C4.5 algorithm over a summary file and the decision tree obtained. We can see that the results files (.tra and .test files with partial results and .txt file with the obtained model) appear in a new window (see Figure 1 down in the right hand corner). Finally, instructors can use this model for decision making concerning the suitability of the Moodle activities in each specific course and also to classify new students depending on the course usage data.

## 4 Experimental Results

We have carried out some experiments in order to evaluate the performance and usefulness of different classification algorithms for predicting students' final marks based on information in the students' usage data in an e-learning system. Our objective is to classify students with equal 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 courses (security and hygienee in the work, projects, engineering firm, programming for enginnering, computer science basis, applied computer science, and scientific programming). Moodle (http://moodle.org) is one of the most frequently used free Learning Content Management Systems (LCMS). Moodle keeps detailed logs of all activities that students perform in a data base. Information is available about the use of Moodle activities and resources (assignments, forums and quizzes). We have preprocessed the data in order to transform them into a suitable format to be used by our Moodle data mining tool. First, we have created a new summary table (see Table 1) which integrates the most important information for our objective (Moodle activities and the final marks obtained in the course). Using our Moodle mining tool a particular teacher could select these or other attributes for different courses during the data preprocessing phase. The Table 1 summarises row by row all the activities done by each student in the course (input variables) and the final mark obtained in this course (class).

**Table 1.** **Attributes used by each student.**

| Name | Description |
|---|---|
| course | Identification number of the course. |
| n_assigment | Number of assignments done. |
| n_quiz | Number of quizzes taken. |
| n_quiz_a | Number of quizzes passed. |
| n_quiz_s | Number of quizzes failed. |
| n_posts | Number of messages sent to the forum. |
| n_read | Number or messages read on the forum. |
| total_time_assignment | Total time used on assignments. |
| total_time_quiz | Total time used on quizzes. |
| total_time_forum | Total time used on forum. |
| mark | Final mark the student obtained in the course. |

Secondly, we have discretized all the numerical values of the summary table 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. Discretization is also a requirement for some algorithms. We have applied the manual method (in which you have to specify the cut-off points) to the mark attribute. We have used four intervals and labels (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). In addition, we have applied the equal-width method [13] to all the other attributes with three intervals and labels (LOW, MEDIUM and HIGH). Then, we have exported both versions of the summary table (with numerical and categorical values) to text files with KEEL format [1]. Next, we have made partitions of whole files (numerical and categorical files) into pairs of training and test files. Each algorithm is evaluated using stratified 10-fold cross-validation. The dataset is randomly divided into 10 disjointed subsets of equal size in a stratified way (maintaining the original class distribution). 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. In this work we also take into consideration the problem of learning from imbalanced data. We say data is 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 resampling the data, either by over-sampling the minority classes or under-sampling the majority ones, until every class is equally represented [3]. 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. But 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 $g-mean=\sqrt[n]{\prod_{i=1}^{n}\frac{hits_i}{ins\tan ces_i}}$, 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 $i$. In our work, we have used random over-sampling, a technique consisting of copying randomly chosen instances of minority classes in the dataset until all classes have the same number of instances, and we use the geometric mean to measure the quality of the induced classifiers.

Finally, we have used three sets of 10-fold data files: the original numerical data, the categorical data and the numerical rebalanced data. We have carried out one execution with all the determinist algorithms and 5 executions with the nondeterministic algorithms. In Table 2 we show the global percentage of the accuracy rate and geometric means (the averages of 5 executions for nondeterministic algorithms). We have used the same default parameters for algorithms of the same type (For example, 1000 iterations in evolutionary algorithms and 4 labels in fuzzy algorithms). We have used these 25 specific classification algorithms due to they are implemented in Keel software, but there are some other classification techniquess such as bayesina networks, logistic regression, etc.

The global percentage of those correctly classified (global PCC) shows the accuracy of the classifiers (see Table 2). More than half of the algorithms obtain their highest values using original numerical data, and the other algorithms obtain them 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 we have 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 (with more than 65% global PCC) with original data (numerical) are CART, GAP, GGP and NNEP. The best algorithms (with over 65% global PCC) using categorical data are the two decision tree algorithms: CART and C4.5. The best algorithms (with over 60% global PCC) with balanced data are Corcoran, XCS, AprioriC and MaxLogicBoost. It is also important to note that no algorithm exceeds 70% global percentage of correctly classified results. One possible reason for this is due to the fact that we have used incomplete data, that is, we have used the data of all the students examined although some students who did not do all the course activities did do the final exam. In particular, about 30% of our students have not used the forum or have not done some quizzes. But we have not eliminated these students from the dataset because it shows a real problem about the students' usage level of e-learning systems. So, we have used all the data although we know that this fact can affect the accuracy of the classification algorithms.

**Table 2. Classification results (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 | 67.02 / 39,25 | 66.86 / 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 | AdaBoost | 62.33 / 0.00 | 60.04 / 0.00 | 50.47 / 0.00 |
| Fuzzy Rule Learning | LogitBoost | 61.17 / 13.05 | 63.27 / 4.64 | 55.70 / 13.95 |
| 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 | RBFN Incremental | 53.65 / 9.87 | 58.00 / 14.54 | 30.31 / 18.32 |
| Neural Networks | RBFN Decremental | 50.16 / 3.95 | 53.44 / 5.61 | 35.32 / 8.41 |
| 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 |

The geometric mean tells us about the effect of rebalancing on the performance of the classifiers obtained, since the geometric mean offers us a better view of the classification performance in each of the classes. We can see in Table 2 that the behavior depends to a great extent on the learning algorithm used. There are some algorithms which are not affected by rebalancing (Kernel, KNN, AprioriC, Corcoran, AdaBoost and LogitBoost): 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 the rebalanced data. Thus we can see that the rebalancing of the data is generally beneficial for most of the algorithms. We can also see 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). But in education this is not very dramatic after all since the most important thing is to be able to distinguish perfectly between FAIL students and PASS students (PASS, GOOD and EXCELENT).

On the other hand, in our educational problem it is also very important for the classification model obtained to be user friendly, so that teachers 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 easily understood models because a reasoning process can be given for each conclusion. However, if the tree obtained is very large (a lot of nodes and leaves) then they are less comprehensible. 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, C4.5 and CART algorithms are simple for instructors to understand and interpret.

- Rule induction algorithms are normally also considered to produce comprehensible models because they discover a set of IF-THEN classification rules that are a high-level knowledge representation and can be used directly for decision making. And some algorithms such as GGP have a higher expressive power allowing the user to determine the specific format of the rules (number of conditions, operators, etc.).

- Fuzzy rule algorithms obtain IF-THEN rules that 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.

- Statistical methods and neural networks are deemed to be less suitable for data mining purposes. This rejection is due to the 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. However, some of the algorithms of this type obtain models people can understand easily. For example, ADLinear, PolQuadraticLMS, Kernel and NNEP algorithms obtain functions that express the possible strong interactions among the variables.

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 could use this discovered knowledge for decision making and for classifying new students. 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 be also possible for the teacher to detect new students with learning problems in time (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).

## 5 Conclusions

In this paper we have compared the performance and usefulness of different data mining techniques for classifying students using a Moodle mining tool. We have shown that some algorithms improve their classification performance when we apply such preprocessing tasks as discretization and rebalancing data, but others do not. We have also indicated that a good classifier model has to be both accurate and comprehensible for instructors. In future experiments, we want to measure the compressibility of each classification model and use data with more information about the students (i.e. profile and curriculum) and of higher quality (complete data about students that have done all the course activities). In this way we could measure how the quantity and quality of the data can affect the performance of the algorithms. Finally, we want also test the use of the tool by teachers in real pedagogical situations in order to prove on its acceptability.

**Acknowledgments.** The authors gratefully acknowledge the financial subsidy provided by the Spanish Department of Research under TIN2005-08386-C05-02 projects. FEDER also provided additional funding.

## References

[1] Alcalá, J., Sánchez, L., García, S., del Jesus, M. et. al. KEEL A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. Soft Computing, 2007.

[2] Baker, R., Corbett, A., Koedinger, K. Detecting Student Misuse of Intelligent Tutoring Systems. Intelligent Tutoring Systems. Alagoas, 2004. pp.531–540.

[3] Barandela, R., Sánchez, J.S., García, V., Rangel, E. Strategies for Learning in Class Imbalance Problems. Pattern Recognition 2003, 36(3), pp.849-851.

[4] Breiman, L. Friedman, J.H., Olshen, R.A., Stone, C.J. Classification and Regression Trees. Chapman & Hall, New York, 1984.

[5] Broomhead, D.S., Lowe, D. Multivariable Functional Interpolation and Adaptative Networks. Complex Systems 11, 1988, pp.321-355.

[6] Chen, G., Liu, C., Ou, K., Liu, B. Discovering Decision Knowledge from Web Log Portfolio for Managing Classroom Processes by Applying Decision Tree and Data Cube Technology. Journal of Educational Computing Research 2000, 23(3), pp.305–332.

[7] Chi, Z., Yan, H., Pham, T. Fuzzy Algorithms: with Applications to Image Processing and Pattern Recognition. World Scientific, Singapore, 1996.

[8] Clark, P., Niblett, T. The CN2 Induction Algorithm. Machine Learning 1989, 3(4), pp.261-283

[9] Cocea, M., Weibelzahl, S. Can Log Files Analysis Estimate Learners' Level of Motivation? Workshop on Adaptivity and User Modeling in Interactive Systems, Hildesheim, 2006. pp.32-35.

[10] Corcoran, A.L., Sen, S. Using Real-valued Genetic Algorithms to Evolve Rule Sets for Classification. Conference on Evolutionary Computation, Orlando, 1994, pp.120-124.

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

[12] del Jesus, M.J., Hoffmann, F., Junco, L., Sánchez, L. Induction of Fuzzy-Rule-Based Classifiers With Evolutionary Boosting Algorithms. IEEE Transactions on Fuzzy Systems 2004, 12(3), pp.296-308.

[13] Dougherty, J., Kohavi, M., Sahami, M. Supervised and Unsupervised Discretization of Continuous Features. Conf. on Machine Learning, San Francisco, 1995. pp.194–202.

[14] Fausett, L., Elwasif, W. Predicting Performance from Test Scores using Backpropagation and Counterpropagation. IEEE Congress on Computational Intelligence, 1994. pp.3398–3402.

[15] Espejo, P.G., Romero, C., Ventura, S. Hervas, C. Induction of Classification Rules with Grammar-Based Genetic Programming. Conference on Machine Intelligence, 2005. pp.596-601.

[16] Hämäläinen, W., Vinni, M. Comparison of machine learning methods for intelligent tutoring systems. Conference Intelligent Tutoring Systems, Taiwan, 2006. pp. 525–534.

[17] Jovanoski, V., Lavrac, N. 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.

[18] Kotsiantis, S.B., Pintelas, P.E. Predicting Students Marks in Hellenic Open University. Conference on Advanced Learning Technologies. IEEE, 2005. pp.664- 668.

[19] Martínez, D. Predicting Student Outcomes Using Discriminant Function Analysis. Annual Meeting of the Research and Planning Group. California, 2001. pp.163-173.

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

[21] Minaei-Bidgoli, B., Punch, W. Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. Genetic and Evolutionary Computation, Part II. 2003. pp.2252–2263.

[22] Moller, M.F. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks, 1993, 6(4), pp.525-533.

[23] Otero, J., Sánchez, L. Induction of Descriptive Fuzzy Classifiers with the Logitboost Algorithm. Soft Computing 2005, 10(9), pp.825-835.

[24] Platt, J. A Resource Allocating Network for Function Interpolation. Neural Computation, 1991, 3(2), pp.213-225.

[25] Quinlan, J.R. C4.5: Programs for Machine Learning. Morgan Kaufman. 1993.

[26] Romero, C., Ventura, S. Educational Data Mining: a Survey from 1995 to 2005. Expert Systems with Applications, 2007, 33(1), pp.135-146.

[27] Rustagi, J.S.: Optimization Techniques in Statistics. Academic Press, 1994.

[28] Sánchez, L., Couso, I., Corrales, J.A. Combining GP Operators with SA Search to Evolve Fuzzy Rule Based Classifiers. Information Sciences, 2001, 136(1-4), pp.175-191.

[29] Sánchez, L., Otero, J. Boosting Fuzzy Rules in Classification Problems under Single-winner Inference (in press). International Journal of Intelligent Systems. 2007.

[30] Superby, J.F., Vandamme, J.P., Meskens, N. Determination of Factors Influencing the Achievement of the First-year University Students using Data Mining Methods. Workshop on Educational Data Mining, 2006. pp.37-44.

[31] Venturini, G. SIA A Supervised Inductive Algorithm with Genetic Search for Learning Attributes based Concepts. Conf. on Machine Learning, 1993. pp.280-296.

[32] Wilson, S.W. Classifier Fitness Based on Accuracy. Evolutionary Computation, 1995, 3(2), pp.149-175.

[33] Yao, X. Evolving Artificial Neural Networks. Proceedings of the IEEE, 1999, 87(9), pp.1423-1447.

[34] Yudelson, M.V., Medvedeva, O., Legowski, E., Castine, M., Jukic, D., Rebecca, C. Mining Student Learning Data to Develop High Level Pedagogic Strategy in a Medical ITS. AAAI Workshop on Educational Data Mining, 2006. pp.1-8.