# Predicting Academic Achievement Using Multiple Instance Genetic Programming

Amelia Zafra, Cristóbal Romero, Sebastián Ventura
Department of Computer Science and Numerical Analysis. University of Cordoba
azafra@uco.es, cromero@uco.es, sventura@uco.es

## Abstract

*The ability to predict a student's performance could be useful in a great number of different ways associated with university-level learning. In this paper, a grammar guided genetic programming algorithm, G3P-MI, has been applied to predict if the student will fail or pass a certain course and identifies activities to promote learning in a positive or negative way from the perspective of MIL. Computational experiments compare our proposal with the most popular techniques of Multiple Instance Learning (MIL). Results show that G3P-MI achieves better performance with more accurate models and a better trade-off between such contradictory metrics as sensitivity and specificity. Moreover, it adds comprehensibility to the knowledge discovered and finds interesting relationships that correlate certain tasks and the time devoted to solving exercises with the final marks obtained in the course.*

## 1. Introduction

The design and implementation of the virtual learning environment (VLE) or e-learning platforms have grown exponentially in the last years, spurred by the fact that neither students nor teachers are bound to a specific location and that this form of computer-based education is virtually independent of any specific hardware platforms [5]. These systems can potentially eliminate barriers and provide: flexibility, constantly updated material, student memory retention, individualized learning, and feedback superior to the traditional classroom, thus becoming an essential accessory to support both the face-to-face classroom and distance learning. The use of these applications accumulates a great amount of information because they can record all the information about students' actions and interactions in log files and data sets. Nowadays, there has been a growing interest in analyzing this valuable information to detect possible errors, shortcomings and improvements in student performance and discover how the student's motivation affects the way he or she interacts with the software [7, 9, 13]. All previous studies have used traditional supervised learning to represent the problem. However, such representation generates instances with many missing values because the information about the problem is incomplete. Each course has different types and numbers of activities and each student carries out the number of activities considered most interesting, dedicating more or less time to resolve them. Recently, a Multiple Instance Learning (MIL) representation has appeared that overcomes the shortcomings of supervised learning traditional. This paper presents a grammar guided genetic programming (G3P) algorithm, G3P-MI, to solve this problem from the MIL perspective. The most representative paradigms in MIL are compared to our proposal. Experimental results show that G3P-MI is more effective in obtaining a more accurate model as well as in finding a trade-off between contradictory measurements like sensitivity and specificity. Moreover, it adds comprehensibility to the knowledge discovered, allowing interesting relationships between activities, resources and results to be obtained. The paper is organized as follows. Section 2 introduces multi-instance learning. Section 3 presents the problem of classifying students' performance from a multi-instance perspective. Section 4 presents the G3P-MI algorithm and section 5 reports on experiment results which compare our proposal to the most representative multiple instance learning paradigms. Finally, Section 6 summarizes the main contributions of this paper and suggests some future research directions.

## 2. Multiple Instance Learning

Multiple Instance Learning (MIL) introduced by Dietterich et al. [6] consists of generating a classifier that will correctly classify unseen patterns. The main characteristic of this learning is that the patterns are bags of instances where each bag can contain different numbers of instances. There is information about the bags because a bag receives a special label, but the labels of instances are unknown. Ac-

cording to the standard learning hypothesis proposed by Dietterich et al. [6] a bag is positive if and only if at least one of its instances is positive, and it is negative if none of its instances produce a positive result. The key challenge in MIL is to cope with the ambiguity of not knowing which of the instances in a positive bag is really a positive example and which is not. In this sense, this learning problem can be regarded as a special kind of supervised learning problem where the labeling information is incomplete. This learning framework is receiving growing attention in the machine learning community because numerous real-world tasks can be very naturally represented as multiple instance problems. Among these tasks are text categorization [1], content-based image retrieval [10], drug activity prediction [8, 19], image annotation [11], web index page recommendation [18] and stock selection [8]. In order to solve these problems, an extensive number of methods have been proposed in the literature. If we go through them, we can find specifically developed algorithms for solving MIL problems, such as APR algorithms [6], the first proposal in MIL; Diverse Density (DD) [8], one of the most popular algorithms in this learning; and some variants, such as EM-DD [19], which combine DD with Expectation Maximization (EM) and a more recent proposal by Pao et al. [10]. On the other hand, there are contributions which adapt popular machine learning paradigms to the MIL context, such as multi-instance lazy learning algorithms which extend k nearest-neighbour algorithms (kNN) [15], multi-instance tree learners which adapt classic methods [3], multi-instance rule inducers which adapt the RIPPER algorithm [4], multi-instance neural networks which extend standard neural networks [2], multi-instance kernel methods which adapt classic support vector machines [1, 11] and multi-instance ensembles which show the use of ensembles in this learning [20]. Finally, we can find a multi-instance evolutionary algorithm which adapts G3P to this scenario [18].

# 3. Predicting Students' performance based on the e-learning Platform

Predicting student's performance based on work they have done on the Virtual Learning Platform is an issue under much research. This problem shows interesting relationships that can suggest activities and resources to students and educators that can favour and improve both their learning and effective learning process. Thus, it can be determined if all the additional material provided to the students (web-based homework) helps them to assimilate the concepts and subjects developed in the classroom or if some activities could be used to improve the final results. The problem could be formulated as follows. A student could do different activities in a course to enable him to acquire and strengthen the concepts acquired in class. Later, at the end of the course, there is a final exam. A student with a mark over a fixed threshold passes a module, while a student with a mark lower than that threshold fails that module. With this premise, the problem consists of predicting if the student will pass or fail the module considering the time dedicated and the number and type of activities done for the student during the course. The types of activities considered in this study are quizzes, assignments and forums. There are lots of strategies and studies which show the effectiveness to strengthen the learning by means of the use of these collaborative and cooperative activities.

## 3.1. MIL representation of the problem

In this problem, each student can execute a different number of activities: a hard-working student may do all the activities available but, on the other hand, there can be students who have not done any activities. Moreover, there are some courses which contain only a few activities and others which contains an enormous variety and number of them. MIL allows a representation that adapts itself perfectly to the concrete information available for each student, eliminating the missing values that abound when traditional representation is used. In MIL representation, each pattern represents a student registered in a course. Each student is regarded as a bag which represents the work carried out. Each bag is composed of one or several instances. Each instance represents the different types of work that the student has done. Therefore, each pattern/bag will have as many instances as the different types of activities done by the student. This representation fits the problem completely because general information about the student and course is stored as bag attributes, and variable information is stored as instance attributes.

Each instance is divided into 3 attributes: type of Activity, number of exercises in that activity and the time devoted to completing that activity. Eight activity types are considered. They are, ASSIGNMENT_S, number of assignments that the student has submitted, ASSIGNMENT referring to the number of times the student has visited the activity without submitting finally any file. QUIZ_P, number of quizzes passed by the student, QUIZ_F number of quizzes failed by the student, QUIZ referring to the times the student has visited a survey without actually answering it, FORUM_POST number of messages that the student has submitted, FORUM_READ number of messages that the student has read and FORUM that refers to the times the student has seen different forums without entering them. In addition, the bag contains three attributes, student identification, course identification and the final mark obtained by the student in that course. A summary of the attributes that belong to the bag and to the instances is presented in Figure 1.

| | |
|---|---|
| *User-Id* | Student identifier. |
| *Course* | Course identifier. |
| *FinalMark* | Final mark obtained by the student in this course. |

(a) Information about Bags

| | |
|---|---|
| *TypeActivity* | Type of activity which represents the instance. The type of activities considered are eight: FORUM read, written or consulted, QUIZ passed or failed and ASSIGNMENT submitted or consulted. |
| *timeActivity* | Time spent to complete the tasks of this type of activity. |
| *numberActivity* | Number of activities of this type completed by the student. |

(b) Information about Instances

**Figure 1. Attributes considered on Multiple Instance Learning Representation**

## 4. Multi-Instance Grammar Guided Genetic Programming, G3P-MI

G3P-MI is an extension of traditional GP systems, called grammar-guided genetic programming G3P [16]. G3P facilitates the efficient automatic discovery of empirical laws providing a more systematic way to handle typing by using a context-free grammar which establishes a formal definition of syntactical restrictions. The motivation to include this paradigm is that it retains a significant position due to a flexible representation using solutions of variable length and the low error rates that it achieves both in obtaining classification rules, and in other tasks related to prediction, such as feature selection and the generation of discriminant functions. The design of the system will be examined in more detail in continuation with respect to the following aspects: individual representation, genetic operators, fitness function and evolutionary process.

### 4.1. Individual Representation

We follow an approach where an individual represents IF-THEN rules. These rules determine if a bag should be considered positive (that is, if it is an instance of the concept we want to represent) or negative (if it is not).

**If** ($cond_B$(bag)) **then**
*the bag is an instance of the concept.*
**Else**
*the bag is an instance of the concept.*
**End-If**

where $cond_B$ is a condition that is applied to the bag. Following the Dietterich hypothesis, $cond_B$ can be expressed as:

$$cond_B(bag) = \bigvee_{\forall instance \in bag} cond_I(instance)$$

where $\vee$ is the disjunction operator, and $cond_I$ is a condition that is applied over every instance contained in a given bag. Figure 2 shows the grammar used to represent the conditions of the rules where IN operator represents an interval considering the extreme values and OUT operator represents an interval not considering the extreme values.

⟨S⟩ → ⟨$\text{cond}_\text{I}$⟩
⟨$\text{cond}_\text{I}$⟩ → ⟨cmp⟩
| **OR** ⟨cmp⟩ ⟨$\text{cond}_\text{I}$⟩
| **AND** ⟨cmp⟩ ⟨$\text{cond}_\text{I}$⟩
⟨cmp⟩ → ⟨op-num⟩ ⟨variable⟩⟨value⟩
| ⟨op-cat⟩ ⟨variable⟩⟨value⟩
| ⟨op-int⟩ ⟨variable⟩⟨value⟩⟨value⟩
⟨op-cat⟩ → **EQ**
| **NOT EQ**
⟨op-num⟩ → **GE**
| **LT**
⟨op-int⟩ → **IN**
| **OUT**
⟨variable⟩ → *Any valid attribute in dataset*
⟨value⟩ → *Any valid value*

**Figure 2. Grammar used for representing individuals' genotypes in G3P-MI**

### 4.2. Genetic Operators

G3P-MI uses two genetic operators to generate new individuals in a given generation of the evolutionary algorithm. The operators are based on selective crossover and mutation proposed by [16]. In this section, we briefly describe their basic principles and functioning.

**Crossover Operator.** This operator creates new rules by mixing the contents of two parent rules. To do so, a non-terminal symbol is chosen at random with uniform probability from among the available non-terminal symbols in the grammar and two sub-trees (one from each parent) are

**Table 1. General Information about Data Sets**

| *Course Identifier* | ICT-29 | ICT-46 | ICT-88 | ICT-94 | ICT-110 | ICT-111 | ICT-218 |
|---|---|---|---|---|---|---|---|
| *Number of Students* | 118 | 9 | 72 | 66 | 62 | 13 | 79 |
| *Number of Assignments* | 11 | 0 | 12 | 2 | 7 | 19 | 4 |
| *Number of Forums* | 2 | 3 | 2 | 3 | 9 | 4 | 5 |
| *Number of Quizzes* | 0 | 6 | 0 | 31 | 12 | 0 | 30 |

selected whose roots coincide with the symbol adopted or with a compatible symbol. In order to reduce bloating, if either of the two offspring is too large, it will be replaced by one of its parents.

**Mutation Operator.** This operator randomly selects the node in the tree where the mutation is to take place. If the node is a terminal node, it will be replaced by another compatible terminal symbol. More precisely, two nodes are compatible if they are derivations of the same non-terminal. When the selected node is a non-terminal symbol, the grammar is used to derive a new subtree which replaces the subtree underneath that node. If the new offspring is too large, it will be eliminated to avoid having invalid individuals.

### 4.3. Fitness Function

The fitness function is a measure of the classifier's effectiveness. There are several measures to evaluate different components of the classifier and determine the quality of each rule. Our fitness function combines two commonly used indicators, namely sensitivity (Se) and specificity (Sp). Sensitivity is the proportion of cases correctly identified as meeting a certain condition and specificity is the proportion of cases correctly identified as not meeting a certain condition.

$$sensitivity = \frac{t_p}{t_p + f_n}, \quad specificity = \frac{t_n}{t_n + f_p}$$

where, $t_p$ is the number of positive bags correctly identified, $f_p$ is the number of positive bags not correctly identified, $t_n$ is the number of negative bags correctly identified and $f_n$ is the number of negative bags not correctly identified. The goal of G3P-MI is to maximize both Sensitivity and Specificity at the same time. These two measurements evaluate different and conflicting characteristics in the classification process where a value of 1 in both measurements represents perfect classification. The fitness function combines both measurements with the product to give the same importance to both measurements and penalize those individuals whose value is 0 in any of the measurements.

$$Fitness = sensitivity * specificity$$

### 4.4. Evolutionary Algorithm

The main steps of our algorithm are based on a classical generational and elitist evolutionary algorithm. Initially, a population of classification rules is generated. Once the individuals are evaluated with respect to their ability to solve the problem, the main loop of the algorithm is composed of the following operations:

**Step 1.** Parents are selected by means of binary tournaments.

**Step 2.** The recombination and mutation processes are carried out with a certain probability and the offspring are obtained and evaluated.

**Step 3.** The population is updated by direct replacement with elitism.

**Step 4.** Finally, the procedure is repeated until the algorithm ends if the maximum number of generations defined by the user is reached or the best individual in the population achieves the quality objectives indicated by the user.

## 5. Experimentation and Results

Experiments compare the performance of G3P-MI to other MIL techniques. All experiments are carried out using 10-fold stratified cross validation and the average values of accuracy, sensitivity and specificity are reported below. First, the problem domain is described briefly and then the results are reported and discussed. Finally, the comprehensibility of the rules generated by G3P-MI will be shown.

### 5.1. Problem domain

This study employs the students' usage data from the virtual learning environment at Cordoba University considering Moodle platform [12] and 7 courses with 419 students. The details about the 7 e-Learning courses are given in Table 1. For the purpose of our study, the collection of data was carried out during an academic year from September to June, just before the Final Examinations.

**Table 2. Experimental Results of Methods based on Multiple Instance Learning**

| ALGORITHM BASED ON | ALGORITHM | ACCURACY | SENSITIVITY | SPECIFICITY |
|---|---|---|---|---|
| Supervised Learning (MISimple) | PART | 0.7357 | 0.8387 | 0.5920 |
| | AdaBoostM1&PART | 0.7262 | 0.8187 | 0.5992 |
| Supervised Learning (MIWrapper) | Bagging&PART | 0.7167 | 0.7733 | 0.6361 |
| | AdaBoostM1&PART | 0.7071 | 0.7735 | 0.6136 |
| | PART | 0.7024 | 0.7857 | 0.5842 |
| | SMO | 0.6810 | 0.8644 | 0.4270 |
| | NaiveBayes | 0.6786 | 0.8515 | 0.4371 |
| Boosting | DecisionStump | 0.6762 | 0.7820 | 0.5277 |
| | RepTree | 0.6595 | 0.7127 | 0.5866 |
| Logistic Regression | MILR | 0.6952 | 0.8183 | 0.5218 |
| Diverse Density | MIDD | 0.6976 | 0.8552 | 0.4783 |
| | MIEMDD | 0.6762 | 0.8549 | 0.4250 |
| | MDD | 0.6571 | 0.7864 | 0.4757 |
| Evolutionary Algorithm | **G3P-MI** | **0.7429** | **0,7020** | **0,7750** |

## 5.2. Results and Disscusion

The most relevant proposals based on MIL presented to date are considered to solve this problem and compared to our proposal designed in JCLEC framework [14]. The different paradigms compared include, Methods based on Diverse Density: MIDD, MIEMDD and MDD; Methods based on Logistic Regression: MILR; Methods based on Support Vector Machines: MISMO uses the SMO algorithm for SVM learning in conjunction with an MI kernel; Distance-based Approaches: CitationKNN and MIOptimalBall; Methods based on Supervised Learning Algorithms: MIWrapper using different learners, such as Bagging, PART, SMO, AdaBoost and NaiveBayes; MISimple using PART and AdaBoost as learners and MIBoost. More information about the algorithms considered could be consulted at the WEKA workbench [17] where these techniques are designed. The average results of accuracy, sensitivity and specificity are reported in Table 2.

G3P-MI obtains the most accurate models. Also, this approach achieves a trade-off between the contradictory measurements of sensitivity and specificity. If we observe the results of the different paradigms, it can be seen how they optimize the sensibility measurement in general at the cost of a decrease in the specificity value. This leads to an incorrect prediction of which students will pass the course. This classification problem has an added difficulty since we are dealing with a variety of courses with different numbers and types of exercises which makes it more costly to establish general relationships among them. Nonetheless, G3P-MI in this sense is the one that obtains the best trade-off between the two measurements, obtaining the highest values for sensitivity without a relevant fall of specificity values. Moreover, G3P-MI obtains interpretable rules to find pertinent relationships that could determine if certain activities influence the student's ability to pass, if spending a certain amount of time on the platform is found to be an important contributing factor or if there is any other interesting link between the work of the student and the final results obtained.

## 5.3. Comprehensibility of the Rules

Our system has the advantage of adding comprehensibility and clarity to the knowledge discovery process. G3P-MI generates a learner based on IF-THEN prediction rules which provide a natural extension to knowledge representation. In continuation, we show an example of the rule generated:

**IF [** ((*NumberOfActivity* $\geq$ 3) $\wedge$ (*TypeOfActivity* = QUIZ_P))
$\vee$ ((*NumberOfActivity* $\in$ [3-8]) $\wedge$ *TimeOfActivity* $\in$
( [2554-11602])) $\vee$ (*NumberOfActivity* $\in$ [6-8]) **]**
**THEN** *The student passes the course.*
**ELSE** *The student fails the course.*

According to this rule, we can determine that passing the course requires at least three passed quizzes, or doing between three and eight activities dedicating between 2554 and 11602 seconds to solve them, or finishing from six to eight activities of any type. The most relevant activity is the quizzes that do not require dedicating a certain time and require completing less number of tasks. The other activities imply handing in more tasks to get similar results.

## 6. Conclusions and Future Work

This paper describes the use of G3P-MI to solve the problem of predicting a student's final performance based on his/her work in VLE. To check effectiveness, the most representative paradigm of MIL is applied to solve this problem, and the results are compared. Experiments show that G3P-MI has better performance than the other techniques at an accuracy of 0.743 and achieves a trade-off between sensitivity and specificity at values of 0.702 and 0.775. Moreover it obtains representative information about the problem that is very useful to determine if all the additional material provided to the students (web-based homework) helps them to better assimilate the concepts and subjects developed in the classroom or what activities are more effective to improve the final results. The results obtained are very interesting. However, there are still a few considerations to improve them. For example, the work only considers if a student passes a course or not. It is would be interesting to expand the problem to predict students' grades (classified in different classes) in an e-learning system. Another interesting issue consists of determining how soon before the final exam a student's marks can be predicted. If we could predict a student's performance in advance, a feedback process could help to improve the learning process during the course.

## Acknowledgments

The authors gratefully acknowledge the financial support provided by the Spanish department of Research under TIN2008-06681-C06-03 and P08-TIC-3720 Projects.

## References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS'02: Proceedings of Neural Information Processing System*, pages 561–568., Vancouver, Canada, 2002.

[2] Y.-M. Chai and Z.-W. Yang. A multi-instance learning algorithm based on normalized radial basis function network. In *ISSN'07: Proceedings of the 4th International Symposium on Neural Networks. Lecture Notes in Computer Science*, volume 4491, pages 1162–1172, Nanjing, China, 2007.

[3] Y. Chevaleyre, N. Bredeche, and J. Zucker. Learning rules from multiple instance data : Issues and algorithms. In *IPMU'02: Proceedings of 9th Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 455–459, Annecy, France, 2002.

[4] Y.-Z. Chevaleyre and J.-D. Zucker. Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem. In *AI'01: Proceedings of the 14th of the Canadian Society for Computational Studies of Intelligence, LNCS 2056*, pages 204–214, Ottawa, Canada, 2001.

[5] S. Chou and S. Liu. Learning effectiveness in web-based technology-mediated virtual learning environment. In *HICSS'05: Proceedings of the 38th Hawaii International Conference on System Sciences*, Washington, USA, 2005.

[6] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artifical Intelligence*, 89(1-2):31–71, 1997.

[7] S. Kotsiantis and P. Pintelas. Predicting students marks in hellenic openuniversity. In *ICALT'05: The 5th International Conference on Advanced Learning Technologies*, pages 664–668, Kaohsiung, Taiwan, 2005.

[8] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *NIPS'97: Proceedings of Neural Information Processing System 10*, pages 570–576, Denver, Colorado, USA, 1997.

[9] B. Minaei-Bidgoli and W. Punch. Using genetic algorithms for data mining optimization in an educational web-based system. *Genetic and Evolutionary Computation*, 2:2252–2263, 2003.

[10] H. T. Pao, S. C. Chuang, Y. Y. Xu, and H. . Fu. An EM based multiple instance learning method for image classification. *Expert Systems with Applications*, 35(3):1468–1472, 2008.

[11] X. Qi and Y. Han. Incorporating multiple svms for automatic image annotation. *Pattern Recognition*, 40(2):728–741, 2007.

[12] W. H. Rice. *Moodle e-learning course development*. Pack Publishing, 2006.

[13] J. Superby, J. Vandamme, and N. Meskens. Determination of factors influencing the achievement of the first-year university students using data mining methods. In *EDM'06: Workshop on Educational Data Mining*, pages 37–44, Hong Kong, China, 2006.

[14] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Hervás. JCLEC: A java framework for evolutionary computation soft computing. *Soft Computing*, 12(4):381–392, 2007.

[15] J. Wang and J.-D. Zucker. Solving the multiple-instance problem: A lazy learning approach. In *ICML'00: Proceedings of the 17th International Conference on Machine Learning*, pages 1119–1126, Standord, CA, USA, 2000.

[16] P. A. Whigham. Grammatically-based genetic programming. In *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, pages 33–41, Tahoe City, California, USA, 9 1995.

[17] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. P2nd Edition. Morgan Kaufmann, San Francisco, 2005.

[18] A. Zafra, S. Ventura, C. Romero, and E. Herrera-Viedma. Multi-instance genetic programming for web index recommendation. *Expert System and Applications*, 2009. Published on line.

[19] Q. Zhang and S. Goldman. EM-DD: An improved multiple-instance learning technique. In *NIPS'01: Proceedings of Neural Information Processing System 14*, pages 1073–1080., Vancouver, Canada, 2001.

[20] Z.-H. Zhou and M.-L. Zhang. Solving multi-instance problems with classifier ensemble based on constructive clustering. *Knowledge and Information Systems*, 11(2):155–170, 2007.