Predicting Student Grades in Learning Management Systems with Multiple Instance Genetic Programming

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

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 Multiple Instance Learning (MIL). Computational experiments compare our proposal with the most popular techniques of 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 [1]. 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 [2-4]. 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. In this context, the Multiple Instance Learning (MIL) representation makes possible a more appropriate representation of available information. MIL stores the general information of each pattern by means of bag attributes and specific information about the student's work on each pattern by means of a variable number of instances. This paper tackles the problem from a MIL perspective and presents a grammar guided genetic programming (G3P) algorithm, G3P-MI, to solve it. 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 and section 3 presents the problem of classifying students' performance from a multi-instance perspective. Section 4 reports on experiment results which compare our proposal to the most representative multiple instance learning paradigms. Finally, Section 5 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. [5] 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. According 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. If we go through them, we can find specifically developed algorithms for solving MIL problems [5,6,7] or, on the other hand, contributions which adapt popular machine learning paradigms to the MIL context, such as multi-instance lazy learning algorithms [8], multi-instance tree learners and multi-instance rule inducers [9], multi-instance neural networks [10], multi-instance kernel methods [11], multi-instance ensembles [12] and finally, a multi-instance evolutionary algorithm [13].

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 are not useful 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 final score higher or equal than a minimum required passes a module, while a student with a mark lower than that minimum fails that lesson or module. With this premise, the problem consists of predicting if the student will pass or fail the module considering the time dedicated, 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. They have shown its effectiveness to strengthen the learning in a lot of studies. A summary of the information available for each activity in our study is shown in Table1.

Table1. Information summary considered in our study

ACTIVITY	ATTRIBUTE NAME	ATTRIBUTE DESCRIPTION
Assignment	numberAssignment	Number of practices/tasks done by the user in the course.
	timeAssignment	Total time in seconds that the user has been in the assignment.
Forum	numberPosts	Number of messages sent by the user forum.
	numberRead	Number of messages read by the user forum.
	timeForum	Total time in seconds that the user has been in the forum.
Quiz	numberQuiz	Number of quizzes seen by the user.
	numberQuiz_a	Number of quizzes passed by the user.
	numberQuiz_s	Number of quizzes failed by the user.
	timeQuiz	Total time in seconds that the user has been in the quiz.

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 with only a few activities along with others with 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 it. Eight activity types are considered which 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 Table2.

Table2. Information about bags and information about instances

BAG		INSTANCE	
User-Id	Student identifier.	*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.
Course	Course identifier.	*timeActivity*	Time spent to complete the tasks of this type of activity.
FinalMark	Final mark obtained by the student in this course.	*numberActivity*	Number of activities of this type completed by the student.

4 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 10 different runs for each partition are executed to measure the performance of evolutionary algorithm. First, the problem domain is described briefly. Then, the results are shown and discussed. Finally, the comprehensibility of the rules generated by G3P-MI will be shown.

4.1 Problem domain used in Experimentation

This study employs the students' usage data from the virtual learning environment at Cordoba University that makes use of Moodle platform[14]. The research includes the information for 7 courses with 419 students. The details about the 7 e-Learning courses are given in Table 3. 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. All information about each student for both representations is exported to a text file using Weka ARFF format [15].

Table3. General information about the courses

COURSE IDENTIFIERS	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

4.2 Multi-Instance Grammar Guided Genetic Programming

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.

We follow an approach where an individual represents IF-THEN rules that add comprehensibility to the discovered knowledge and the fitness function to evaluate the rules obtained will be *sensitivity *specificity*. These measurements allow us to consider both successes in the positive and negative class assigning a value of 0 when no example of one class is classified and value of 1 when both classes are full classified.

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 parent selection using a binary tournament selector, then recombination and mutation processes [16] are carried out with a probability of 90% and 10% respectively, and finally, the population is updated by direct replacement with elitism, that is, the offspring replace the present population and the best individual in the population is included. The procedure is repeated until de algorithm reaches a maximum number of one hundred generations or the best individual in the population achieves a full classification (a value of 1 in fitness function).

4.3 Comparison with Multiple Instance Learning techniques

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 [17]. The different paradigms compared included, *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 [15] where these techniques are designed. The average results of accuracy, sensitivity and specificity are reported in Table 4.

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 optimise 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 make it more complicate 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. 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 an important contribution or if there is any other interesting link between the work done and the final results obtained.

Table 4. Results for multiple instance learning algorithms

	Algorithm	Accuracy	Sensitivity	Specificity
Methods Based on supervised learning (Simple)	PART	0.7357	0.8387	0.5920
	AdaBoostM1&PART	0.7262	0.8187	0.5992
Methods Based on supervised learning (Wrapper)	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
Methods Based on Distance	MIOptimalBall	0.7071	0.7218	0.6877
	CitationKNN	0.7000	0.7977	0.5631
Methods Based on Boost	DecisionStump	0.6762	0.7820	0.5277
	RepTree	0.6595	0.7127	0.5866
logistic regression	MILR	0.6952	0.8183	0.5218
Methods Based on 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**

4.4 Comprehensibility in the knowledge discovery process

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. These rules are simple, intuitive, easy to understand and provide representative information. In continuation, we show an example of the rule generated:

IF ((*NumberOfActivities* ≥ 3) AND (*TypeOfActivity EQ* QUIZ_P)) OR
((*NumberOfActivities* IN [3-8]) AND (*TimeOfActivity* IN [2554. 11602])) OR
(*NumberOfActivities* [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. We can conclude that the most relevant activity is the quizzes that do not require dedicating a certain time and require completing less number of tasks. On the contrary, the rest of the activities imply handing in more tasks and spending more time to get similar results.

5 Conclusions and Future Works

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 from MIL perspective. To check effectiveness, the most representative paradigm of multiple instance learning 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. Thus, more interesting relationships could be found between the work done by the student and the precise mark obtained. 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.

References

[1] Chou, S and Liu, S. Learning Effectiveness in Web-based Technology-mediated Virtual Learning Environment. HICSS'05*:* Proceedings of the 38th Hawaii International Conference on System Sciences, Washington, USA, *2005.*

[2] Finaei-Bidgoli, B. and Punch, W. Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. Genetic and Evolutionary Computation, 2, 2252–2263, 2003.

[3] Superby, J.F., Vandamme, J.P., Meskens, N. Determination of Factors Influencing the Achievement of the First-year University Students using Data Mining Methods. EDM'06: Workshop on Educational Data Mining, 37-44, 2006.

[4] Kotsiantis, S.B., Pintelas, P.E. Predicting Students Marks in Hellenic OpenUniversity. ICALT'05: The 5th International Conference on Advanced Learning Technologies, 664-668, 2005.

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

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

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

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

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

[10] Chai, Y.M., Yang, Z.-W. A multi-instance learning algorithm based on normalized radial basis function network. ISSN'07: Proceedings of the 4th International Symposium on Neural Networks. LNCS 4491, Nanjing, China, 1162–1172, 2007.

[11] Han Q.Y., Incorporating multiple SVMs for automatic image annotation, Pattern Recognition, 40(2), 728–741, 2007.

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

[13] Zafra, A., Ventura, S., Romero C., Herrera-Viedma, E. Multi-Instance Genetic Programming for Web Index Recommendation, 2009, doi:10.1016/j.eswa.2009.03.059.

[14] Rice, W. H. Moodle e-learning course development. A complete guide to successful learning using Moodle. Pack Publishing, 2006.

[15] Witten, I.H., Frank, E. Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco, 2005.

[16] Whigham, P. A. Grammatical bias for evolutionary learning, Ph.D. thesis, School of Computer Science, University College, University of New South Wales, Australian Defence Force Academy, Canberra, Australia, 1996.

[17] Ventura, S., Romero, C., Zafra, A., Delgado, J.A., Hervás, C.: JCLEC: A java framework for evolutionary computation soft computing. Soft Computing, 12(4), 381-392, 2008.