Anticipating student's failure as soon as possible

Cláudia Antunes
Dep. Information Systems and Computer Science
Instituto Superior Técnico
Av Rovisco Pais 1, 1049-001
Lisboa – Portugal
+351 218 419 407
claudia.antunes@ist.utl.pt

Abstract. Despite the increase of interest on education and the quantity of existing data about students' behaviors, neither of them, per se, are enough to predict when some student will fail. Anticipating students' failure becomes an even harder task, when the goal is to predict the failure as soon as possible. Indeed, the classification problem doesn't deal easily with the temporal nature of this kind of data, since it considers each instance attribute as important as others. In this paper, it is shown how different data mining techniques (namely classification and pattern mining) can be combined to surpass this difficulty. Experimental results show that the accuracy of these new methods is very promising, when compared with classifiers trained with smaller datasets.

1. Introduction

With the spread of information systems and the increase of interest on education, the quantity of existing data about students' behaviors has exploded in the last decade. Those datasets are usually composed of records about students' interactions with several curricular units. On one hand, these interactions can be related to traditional courses (taught at traditional schools), that reveal the success or failure of each student, on each assessment element of each unit that student has attended. On the other hand, there are the interactions with intelligent tutoring systems (ITS), where each record stores all students' interactions with the system. In both cases, records for

each student have been stored at different instants of time, since both attendance of curricular units, and corresponding assessment elements, and ITS interactions, occur sequentially, in a specific order. Despite this order can be different for each student, the temporal nature of the educational process is revealed in the same way: each student record corresponds to an ordered sequence of actions and results.

Once there are large amounts of those records, one of their possible usages is on the automatic prediction of students' success. Work on this area has been developed, with the research being focused mainly on determining students' models (see for example (Baker & Carvalho, 2008a) and (Beck, 2007)), and more recently on mining frequent behaviors ((Antunes, EDM, 2008) and (Romero, Ventura, Espejo, & Hervas, 2008)). Exceptions to this general scenario are the works by (Superby, Vandamme, & Meskens, 2006), (Vee, Meyer, & Mannock, 2006) and (Antunes, 2009) that try to identify failure causes. For predicting students' success, existing data *per se* is not enough, but combined with the right data mining tools, can lead to very interesting models about students behavior. Classification is the natural choice, since it is able to produce such models, only based on historical data (training datasets as the ones described above). Once these models are created, they can be used as predictive tools for new students' success.

Despite the excellent results of classification tools on prediction in several domains, the educational process presents some particular issues that bring additional challenges to this task. In this chapter, we will describe how traditional classifiers can be adapted to deal with these situations, after they have been trained in full and rich datasets. We cease the opportunity to succinctly explain the classification problem and describe the methodology adopted to create ASAP classifiers (*as soon as possible classifiers*). The chapter will also include a detailed study about the accuracy of these new classifiers when compared with the traditional ones, both

applied on full and reduced datasets.

In order to demonstrate that ASAP classifiers can anticipate students' failure in an interesting time window, the chapter will present a case study about students' performance recorded on the last five years, in a subject of an undergraduate program.

The rest of the chapter is organized as follows: next, the classification problem is described, giving particular attention to the most common measures for their efficacy; in this section, the problem of as soon as possible classification (ASAP classification) is also introduced. In section 3, a methodology for the ASAP classification is proposed. Section 4, presents the case study for evaluating the new methodology, concluding with a deep study on the impact of different factors. The chapter closes with a critical analysis of the achieved results and points out for future directions to solve the problem.

2. The classification problem

a. Problem statement and evaluation criteria

Automatic classification of records has its roots on the area of artificial intelligence and machine learning, and aims to classify one entity in accordance to its attributes and similar historical entities.

In this manner, a *classifier* is just a function *f* from the set of instances *I* to the set of *classes* or *concepts C*, where each instance is characterized by a set of *attributes* or *variables*. Whenever the number of elements in *C* is finite (usually small, say *k*), a classifier is just a partition mechanism that splits the set of instances *I* into *k* subsets, I_k, each one corresponding to one class. Therefore, a classifier can be seen as a *model*, composed by the definition of each concept (class). In order to find such models, classification methods use a set of historical instances, with their own classification (usually known as *training set*). Once the model is discovered, and

assuming that all new instances follow the same probability distribution as the training set, it may be applied to predict the class for any unseen instance.

Several have been the methods applied for discovering (or training) classifiers, all of them needing a specific *generalization language* to represent learnable models (Mitchell 1982). Among the most well known approaches are decision trees (Quinlan 1986), neural networks (Nilsson 1965), Bayesian classifiers, and more recently support vector machines (Vapnik 1995). All of them create a model based on the training set that can be used in classification time, without re-analyzing known instances.

In a general and simplified way, the different learning approaches can be compared following some criteria when applied to a particular problem: i) the time spent on training the classifier; ii) the time spent on classifying an instance; iii) the tolerance to the existence of incoherencies in the training set, and iv) their accuracy. Since, some problems are best solved with some of these approaches, and none of them are better than the others in all situations, the important issue is to know what classifier to apply in a particular domain, and how to assess its quality.

The first issue does not have a consensual answer, but the second one is perfectly defined. Indeed, the quality of a specific classifier is measured by its *accuracy* in an independent dataset (usually known as *testing set*, and represented by D in the following expressions).

The *accuracy* of a classifier h is the percentage of instances in the testing set D that are correctly classified (Figure 1), where $h(x_i)$ is the estimation for x_i's class and $c(x_i)$ its own classification.

$$accuracy = \frac{\#\{h(x_i) = c(x_i) : x_i \in D\}}{\#\{x_i \in D\}}$$

Figure 1 – Formula for accuracy

In order to distinguish the ability to classify instances from different classes, it is usual to use the *confusion matrix*. This is a $k \times k$ matrix (with k the number of classes), where each entry x_{ij}

corresponds to the percentage of instances of class *i* that are classified as in class *j*. Therefore, a diagonal matrix reveals an optimal classifier.

When there are only two classes, it is usual to talk about positive and negative instances: instances that implement the defined concept and the ones that do not implement it, respectively. In this case, it is usual to designate the confusion matrix diagonal entries as *true positives* (TP), and *true negatives* (TN), and the others as *false positives* (FP) and *false negatives* (FN), as depicted in Figure 2.

		Classification	
		Positive	**Negative**
Real	**Positive**	TP	FN
	Negative	FP	TN

Figure 2 – Confusion matrix for binary classification

In the binary case, it is useful to use some additional measures, namely sensitivity and specificity. While the first one reflects the ability to correctly identify positive cases, the second one reveals the ability to exclude negative ones. Hence, *sensibility* is given by the ratio between the number of instances correctly classified as positives (TP) and the number of real positive instances (TP+FN), Figure 3.

$$sensibility = \frac{\#\{h(x_i) = c(x_i) : x_i \in D \cap Positive\}}{\#\{x_i \in D \cap Positive\}} = \frac{TP}{TP + FN}$$

Figure 3 – Formula for sensibility

By the other side, *specificity* is given by the ratio between the number of instances correctly classified as negative (TN) and the number of real negative instances (TN+FP), as shown in Figure 4. In this manner, specificity is just sensibility for the negative class.

$$specificity = \frac{\#\{h(x_i) = c(x_i) : x_i \in D \cap Negative\}}{\#\{x_i \in D \cap Negative\}} = \frac{TN}{TN + FP}$$

Figure 4 – Formula for specificity

b. Anticipating failure as soon as possible

By nature, the educational process begins with students that do not have any knowledge about the topic to be taught, and aims to end with the same student full of that knowledge. At the same time, the process occurs during some time interval, where students are asked to interact with the system, either almost continuously or at specific instants of time. The results of these interactions are usually used to assess students' success and progress. Another important characteristic of the educational process is the fact that the success or failure of the student is determined at a particular instant of time, usually at the end of the process.

Consider for example, the enrolment of a student at an undergraduate subject: he starts to interact with the curricular unit, usually with some small assignments, that have a small weight on the final grade; as the time goes on, student is confronted with harder tasks and at the end of the semester he has to be evaluated on a final exam and has to deliver a final project. His final mark is determined according to a mathematical formula that weights the different interactions. In this case, the best classifier is the one that corresponds to that mathematical formula, which can be represented accurately by all the referred approaches. However, despite the perfect accuracy of this method, the goal is not achieved: it is not able to predict if some student will fail in advance.

To our knowledge, in order to accomplish the goal of predicting students result in advance, the training of classifiers have to suffer some adaptations, namely on weighting the different attributes based on their instance of occurrence. In this manner, oldest attributes have higher weights in the classification than the youngest ones. However, this is against educational process, since the classifier would be almost the reverse of the optimal one.

In the next section, a new approach is proposed, avoiding this contra-nature approach.

3. ASAP classification

The amounts of data needed for training a classifier for achieving a specific accuracy has been study thoroughly, with very interesting results (see for example (Vapnik e Chervonenkis 1971), or more recently (Domingos e Hulten 2000) in other contexts). However, the problem of classifying instances that are partially observable, when classifiers may be trained with fully observable instances, is to our knowledge unstudied. Note that Bayesian networks are trained when such unobservable attributes exist, but this is not the problem defined here. For this reason, the problem statement is presented below.

a. Problem statement

Let I be a set of instances, A a set of attributes and C a set of possible classes; an instance x_i from I is described by an ordered list of m attributes from A, and is represented as $x_i=x_{i1}x_{i2}\ldots x_{im}c_i$, where $c_i \in C$ corresponds to the class of x_i.

Given a set of instances from I, described by m attributes from A, and a number of attributes n, such that $n<m$, the problem of *classifying* an instance x_i *as soon as possible* consists on determining the value of c_i – the class for x_i, only considering its first n attributes, known as the *observable attributes*.

The distinction of this formulation to the traditional formulation of classification, lies on the notion of order among the attributes that characterize an instance, and the fact that the classifier for instances with m attributes have to be able to classify instances described by a fewer number of attributes.

Note, however, that nothing is said about the training process of classifiers. Indeed, the use of a training set, composed of historical records, is compatible with the notion that these historical instances are fully observable, which means, that classifiers can be trained using the traditional

methods without needing any adaptation.

From this point forward, classifiers that work in the context of this formulation are denominated, *as soon as possible classifiers – asap classifiers*, for short.

These new classifiers can be trained using two different strategies: a first one, based on the usage of the entire set of attributes, named the *optimistic strategy*, and a second one, the *pessimistic strategy*, that train the classifier only using the observable attributes.

The pessimistic strategy converts the problem into the traditional problem of classification, by reducing each training instance from its original m-dimensional space, to an n-dimensional space, with $n<m$. Clearly, this strategy does not use all the information available at classification time. Indeed, it wastes the historical values of the unobservable attributes, existing in the training set. For this reason, it is expected that the pessimistic strategy will lead to the creation of less accurate classifiers.

On the other hand, the optimistic strategy needs to train classifiers from m-dimensional instances that can be applied to n-dimensional ones. Again, it is possible to consider two approaches, either to convert the learnt classifier, a function from A^m to C, into a function from A^n *to* C, or to convert the n-dimensional instances into m-dimensional ones.

Note that both approaches require some non-trivial transformation. In the case of the second approach, it tries to enrich non-fully observable instances, which can be achieved with any method capable of estimating unobservable attributes from observable ones. Next, an approach based on Class Associations Rules is described.

b. CAR-based ASAP classifiers

Association analysis is an unsupervised task, which tries to capture existing dependencies among attributes and its values, described as *association rules*. The problem was first introduced in 1993

(R. Agrawal 1993), and is defined as the discovery of "all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence, respectively". An *association rule* corresponds to an implication of the form $A \Rightarrow B$, where A and B are propositions (sets of pairs attribute/value), that expresses that when A occurs, B also occurs with a certain probability (the rule's *confidence*). The *support* of the rule is given by the relative frequency of instances that include A and B, simultaneously. In this case, A and B are named the *antecedent* and the *consequent* of the rule, respectively. In this manner, while confidence measures the effectiveness of a rule, its support accounts for its coverage.

A *Class Association Rule* (*CAR* for short) is an association rule, which consequent is a single proposition related to the value of the class attribute. In this manner, a set of CARs can be seen as a classifier. In this manner, each rule has an *accuracy* value, with the same meaning as introduced before, but that can be estimated as described in (Scheffer, 2001).

Considering the problem of ASAP classification, as described above, and adopting an optimistic strategy, the *CAR-based ASAP classifier*, proposed in this paper, makes use of class association rules for estimating the values of unobservable attributes.

Given a set of training instances D, described by an ordered list of m fully observable attributes and a class label $y \in C$ (the set of class labels), say $a_1 a_2 \ldots a_m y$. The classification of a new unseen instance z described by the same list of attributes, but where only the first n attributes are observable (with $n<m$), by the CAR-based ASAP classifier is done as follows:

- First, a classifier is trained based on the entire training set D, for example using decision trees learners;
- Second, for each unobservable attribute a_i:
    - it creates a subset of D, D_i, with instances characterized by attributes a_1 to a_n,

followed by attribute a_i; this last attribute is set to assume the role of class;

- and then it finds the set of all classification association rules in set D_i, that satisfy chosen levels of confidence, support and accuracy, CAR_i.

– Thirdly, for each unobservable attribute in instance z, z_i:

- it identifies the rules from CAR_i that match instance z, CAR_{zi};
- among the consequents of rules on CAR_{zi}, it chooses the best value for filling in the value of z_i, z_i'.

– Finally, in order to predict the class label of instance z, it creates a new instance with m attributes z', such that $z'=z_1 z_2 \ldots z_n z_{n+1}' \ldots z_m'$ and submits it to the learnt classifier.

Note that a similar methodology can be adopted even when other estimators are used for predicting the value of unobservable attributes. It is only necessary to define what means "the best value". In the case of the CAR-based ASAP classifier, the best value results from combining the different matching rules and choosing the one that better matches the instance. The combination of matching rules is necessary to identify the different possible values and their respective interest, since possibly there are several rules that match with a unique instance.

An instance x is said *to match* a class association rule r, designated by $x \models r$, if and only if

$$x = x_1 x_2 \ldots x_n \; matches \; \mathrm{r} =< a_1 = v_1 \wedge a_2 = v_2 \wedge \ldots \wedge a_n = v_n \Rightarrow \mathrm{Y} = \mathrm{y} >$$

$$\leftrightarrow \forall i \in \{1, \ldots, n\}: x_i \; is \; missing \vee x_i = v_i$$

A rule better matches an instance than another one, if the first is more specific than the second, which means that it has more matching attributes with the instance, and fewer missing values. In this manner, more restrictive rules are preferred. Note that despite variables x_i with $i \leq n$, are observable, some instances may present missing values for those variables, since their values may not be recorded. For example, the rule $a_1 = v_1 \wedge a_2 = v_2 \wedge a_n = v_n \Rightarrow \mathrm{Y} = \mathrm{y}$ is a better match

than $a_1 = v_1 \wedge a_2 = v_2 \Rightarrow Y = y$, since it is more specific than the second one (it has an additional constraint on the value of a_n).

The major disadvantage of the methodology proposed is that the existence of rules that match with some instance is determinant for the success of the final classification. Indeed, if it is not possible to estimate the value for each non observable attribute, then most of the times the classifier cannot improve its accuracy.

One important advantage of CAR-based ASAP classifier is the direct influence of minimum confidence and minimum support thresholds on the number of matching rules. Definitely, with lower levels of support the number of matching rules increase exponentially (a phenomenon well-studied in the area of association analysis), and lower levels of confidence will decrease the certainty of the estimation, increasing the number of errors made. In order to increase the accuracy of the estimation of unobservable attributes, a rule is selected to estimate a value, only if it has an accuracy higher than a user-specified threshold.

4. Case study

In this chapter we claim that traditional classifiers can be beaten by ASAP classifiers on predicting students' failure. In order to support this claim, some results of applying both kinds of classifiers to a specific dataset of an educational process, are described.

The dataset in study stores the results of students enrolled in the five last years, in the subject of *Foundations of Programming* of an undergraduate program at *Instituto Superior Técnico*. The dataset has 2050 instances, each one with 16 attributes. From these, 12 correspond to considered observable attributes: 11 weekly exercises and 1 test. Unobservable attributes are the project (ATT13), the exam (ATT14) and another optional exam (ATT15). All have a classification from *A* to *F* (and *NA* – meaning not evaluated). The last attribute corresponds to the classification

obtained at the end of the semester (*Approved* – A, B, C, D or E, and *Failed* – F).

The initial dataset was split into two sets: the training set with 75% of instances and the test set with 25% of instances. Training and testing sets, named *train* and *full test set*, respectively, from this point forward, were pre-processed creating two different sets: the *small training set* and the *small testing set*. These sets have the same number of instances as the preceding sets, but fewer attributes (in this case the first 12 attributes of the original set). The small training set will be used to train the pessimistic classifier and to identify CARules for each unobservable attribute (by being enriched with its corresponding column); the small testing set will be used to assess the accuracy of pessimist and CAR-based ASAP classifiers.

In order to compare the results, three classifiers were trained using the C4.5 algorithm (Quinlan, 1993), implemented by J48 on the WEKA software (Witten e Frank 2000)): optimal, pessimistic and CAR-based ASAP.

Optimal classifier was trained in the full training dataset and tested in the corresponding testing set (with all attributes: observable and unobservable ones). It serves as a reference line, and gives the best possible classifier for this data using C4.5 (represented on Figure 5 – left).

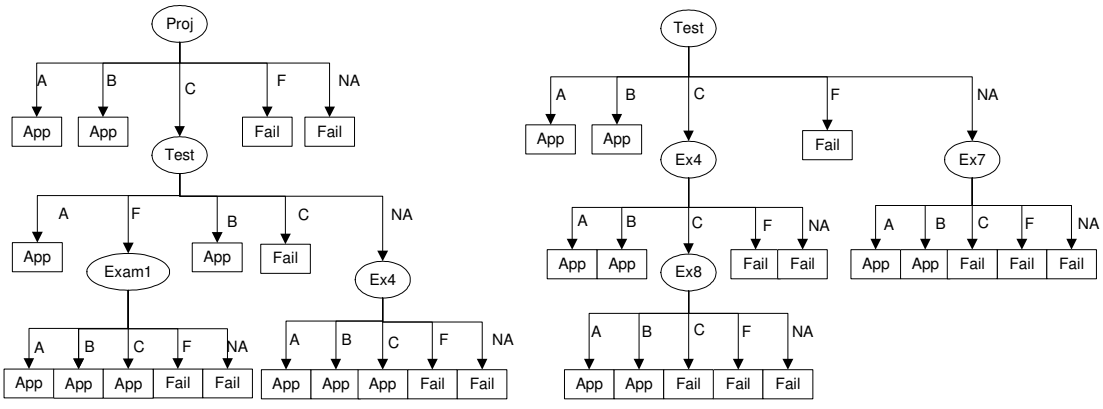

Figure 5 – Discovered decision trees for optimal (left) and pessimistic (right) classifiers

Pessimistic classifier, on the other hand, was trained using small training set and tested in the corresponding small testing set. Again, it serves as a reference line for the best model created

when there is loss of information; the decision tree discovered is on Figure 5 – right.

At last, CAR-based ASAP classifier corresponds to the optimal one discovered and is tested on the estimated testing set. This set is created by extending the small testing set with estimations for unobservable attributes, creating a new full testing set.

Results confirm the expectations. CAR-based ASAP classifier presents relative success, when compared with pessimistic approaches. Indeed, for lower levels of support (0.5%) the accuracy achieved is always above the accuracy of the pessimistic classifier (Figure 6 – left).

It is important to note that for higher levels of support, the discovered rules do not cover the majority of situations of recorded students' results. This is explained by the sparsity of data, resulting from the large number of evaluation moments (variables): there are just a few students with similar behaviors. In order to find them, it is necessary to consider very low levels of support.

Additionally, note that decreasing the level of confidence would decrease the quality of the discovered rules. See that for fixed levels of support the accuracy of the classifier decreases when the confidence also decreases. This is because rules with lower confidence are more generic and less predictive than the ones with larger confidence.

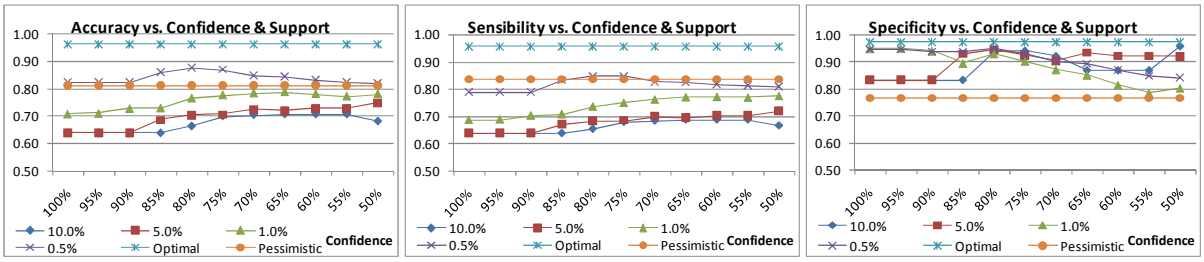

Figure 6 – Accuracy, sensibility and specificity for different levels of support and confidence (for a minimum CAR accuracy of 50%)

Another interesting result is on predicting failure: ASAP classifier (Figure 6 right) has always better specificity levels than the pessimistic one. This reflects the ability of our classifier to cover

both positive and negative cases with the same efficacy, since it both looks for failure and success rules.

A look on the percentage of correct estimation of attribute values with CARs again shows that the accuracy increases with the decrease of support. However, the increase of correct values is not directly correlated with the decrease of missing values. Indeed, when confidence decreases, missing values also decrease but the correct percentage does not (Figure 7).

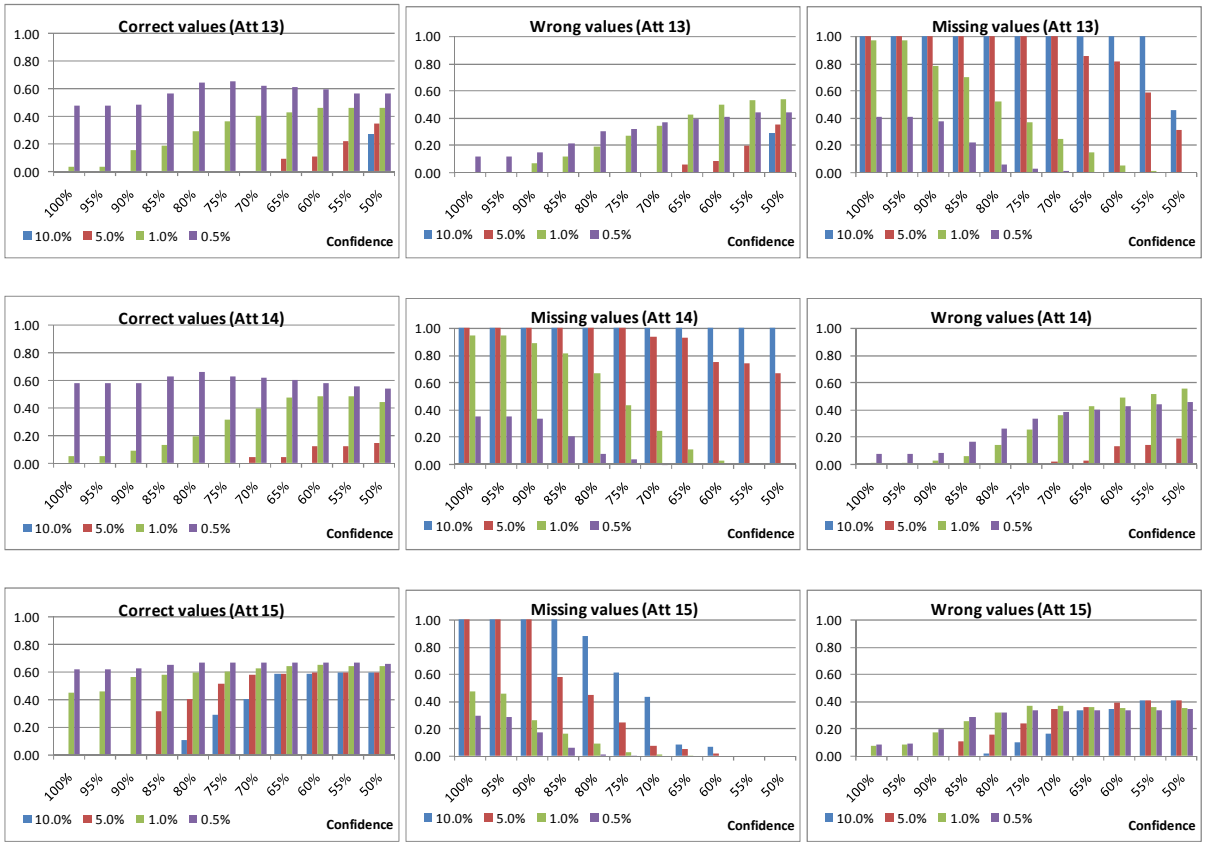

Figure 7 – Impact of support and confidence on the estimation of unobservable attributes (for a minimum CAR accuracy of 50%)

Again, this is due to the different confidence levels of discovered rules: if lower levels of confidence are preferred, then there will be more discovered rules, that would cover more instances, and then they will be used to estimate more unobservable values. However, since confidence is not high, the accuracy of the rule is not enough, and missing values are replaced by

wrong ones. With higher levels of confidence, the accuracy of rules will be higher, and the number of wrong values will be reduced: missing values will prevail.

Note that the most important factor on the accuracy of the CAR-based ASAP classifier is the accuracy of the discovered rules. Indeed, interesting results appear when the minimum cutoff for CAR accuracy does not impair the estimation of values (Figure 8).

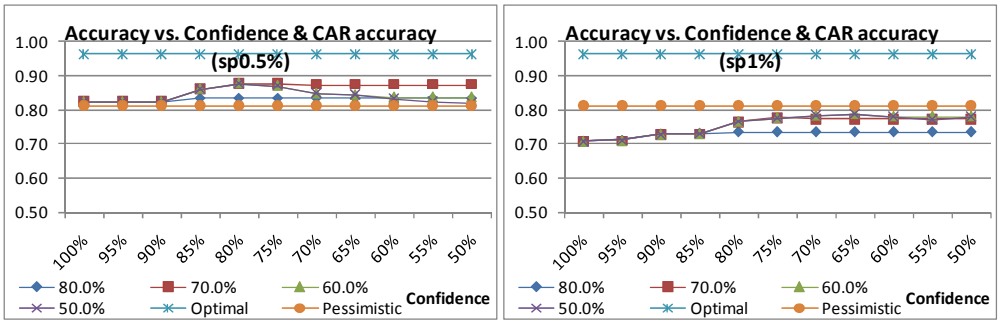

Figure 8 – Impact of accuracy and confidence on the estimation of unobservable attributes

High levels of CAR accuracy exclude most of the discovered rules, which results on attributing missing values for the unobservable attribute. It is important to note that C4.5 in particular presents better results on dealing with missing values that for wrong estimations.

At last, another interesting fact is that the tree discovered with the pessimistic approach has the same number of leaves as the optimistic one (Figure 5). However, while the optimal classifier tests the primary attributes (the ones that have a minimum threshold for a student have success – Project and Exams), the pessimistic one tests the attribute Test and some of the weekly exercises. Little changes on the teaching strategy (changing the order of presenting concepts) will invalidate the pessimistic classifier.

5. Conclusions

In this paper we introduce a new formulation for predicting students' failure, and propose a new methodology to implement it. Our proposal makes use of classifiers, trained as usual, using all available data, and the estimation of unobservable data (attributes) in order to predict failures as

soon as possible. In our case, this estimation is based on the discovery of class association rules. Experimental results show that our methodology can overcome the difficulties of approaches that do not deal with the entire set of historical data, and by choosing the best parameters (confidence, support and rule accuracy) the results become closer to the optimal classifier found in the same data. However, the choice of the best parameters is a hard task, followed by the traditional problems related with the explosion on the number of discovered rules. Other methodologies able to estimate the values of unobservable variables (the EM algorithm (A.P.Dempster 1997) is just one of such possibilities) can also be applied to determine *ASAP classifiers*. A study on their application, advantages and disadvantages is mandatory, in order to understand the total potential of *ASAP classifiers*.

References

A.P.Dempster, N. D. (1997). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series , 39*, 1-38.

Antunes, C. (2008). Acquiring BackgroundKnowledge for Intelligent Tutoring Systems. *Int'l Conf Educational Data Mining*, (pp. 18-27). Montreal, Canada.

Antunes, C. (2009). Mining Models for Failing Behaviors. *Int'l Conf on Intelligent Systems Design and Applications.* IEEE Press.

Baker, R., & Carvalho, A. (2008a). Labeling Student Behavior Faster and More Precisely with Text Replays. *Int'l Conf Educational Data Mining*, (pp. 38-47).

Beck, J. (2007). Difficulties in inferring student knowledge from observations (and why should we care). *Workshop Educational Data Mining – Int'l Conf Artificial Intelligence in Education*, (pp. 21-30).

Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. *ACM SIGKDD Int'l Conf.*

on Knowledge Discovery and Data mining (pp. 71-80). ACM Press.

Mitchell, T. M. (1982). Generalization as Search. *Artificial Intelligence , 18* (2), 223-236.

Nilsson, N. (1965). *Learning Machines: Foundations of Trainable Pattern-Classifying Systems.* New York: McGraw-Hill.

Quinlan, J. (1993). *C4.5 Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning* , 81-106.

R. Agrawal, T. I. (1993). Mining Association Rules between Sets of Items in Large Databases. *ACM SIGMOD Conf. on Management of Data*, (pp. 207-216).

Romero, C., Ventura, S., Espejo, P., & Hervas, C. (2008). Data Mining Algorithms to Classify Students. *Int'l Conf Educational Data Mining*, (pp. 8-17).

Scheffer, T. (2001). Finding Association Rules That Trade Support Optimally against Confidence. *European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'01)* (pp. 424-435). Springer-Verlag.

Superby, J., Vandamme, J.-P., & Meskens, N. (2006). Determining of factors influencing the achievement of first-year university students using data mining methods. *Intelligent Tutoring System (ITS): Educational Data Mining Workshop*, (pp. 37-44).

Vapnik, V. (1995). *The nature of statistical learning theory.* Springer.

Vapnik, V., & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications , 16* (2), 264-280.

Vee, M., Meyer, B., & Mannock, K. (2006). Understanding novice errors and error paths in object-oriented programming through log analysis. *Intelligent Tutoring System (ITS): Educational Data Mining Workshop*, (pp. 13-20).

Witten, I., & Frank, E. (2000). *Data Mining: practical machine learning tools and techniques*

with Java implementations. Morgan Kaufmann.