Merging Decision Trees: A Case Study in Predicting Student Performance

Pedro Strecht, João Mendes-Moreira, and Carlos Soares

INESC TEC/Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
{pstrecht,jmoreira,csoares}@fe.up.pt

Abstract. Predicting the failure of students in university courses can provide useful information for course and programme managers as well as to explain the drop out phenomenon. While it is important to have models at course level, their number makes it hard to extract knowledge that can be useful at the university level. Therefore, to support decision making at this level, it is important to generalize the knowledge contained in those models. We propose an approach to group and merge interpretable models in order to replace them with more general ones without compromising the quality of predictive performance. We evaluate our approach using data from the U. Porto. The results obtained are promising, although they suggest alternative approaches to the problem.

Keywords: prediction of failure, decision tree merging, C5.0.

1 Introduction

Interpretable models for predicting the failure of students in university courses are important to support both course and programme managers. By identifying the students in danger of failure beforehand, suitable strategies can be devised to prevent it. Moreover, those models can give clues about the reasons that lead to student attrition, a topic widely studied in educational data mining [1]. Given the vast amount of data available in university information systems, these models are usually created for each course and academic year separately. This means that a very large number of models is generated, which raises problems on how to generalize knowledge in order to have a global view of the phenomena across the university and not only in the context of a single course.

Additionally, it may be expected that there are different groups of models with very different characteristics. For instance, the performance in courses of different scientific areas is likely to be affected by different factors. We propose an approach for the problem of generalizing the knowledge contained in a large number of models that consists of two phases. In the first one, the models are split into groups. The splitting is done based on domain-specific knowledge (e.g. scientific areas) or is data-driven (e.g. by clustering them). In the second phase, the models in a group are aggregated into a single model that generalizes the knowledge contained in the original models, hopefully with small impact on its

X. Luo, J.X. Yu, and Z. Li (Eds.): ADMA 2014, LNAI 8933, pp. 535–548, 2014.

predictive performance. The aggregation method consists mainly of intersecting the decision rules of pairs of models of a group recursively, i.e., by adding models along the merging process to previously merged ones.

In this paper we compare different methods of grouping models and of defining weights of decision rules as part of a strategy to keep the merged models simple and generic. Throughout the merging process we work only with decision rules and delay the merged decision tree creation to the final step. We define an evaluation procedure to compare performance of merged models relatively to the original ones. The case study used for empirical evaluation uses data from the academic management information system of the University of Porto, Portugal. Due to limitations of space, this paper focuses on the process of merging trees. Therefore, some decisions which were based on domain-specific knowledge and preliminary experiments (e.g. variable selection, parameter setting) as well as some aspects of the results have not been discussed in depth.

The main contributions of this paper are: 1) propose the methodology to generalize the knowledge from a large number of models; 2) identify which are the components of the merging process; 3) define different alternatives for these components; and 4) combine them in a series of experiments to assess the impact in the global predictive performance of the merged models.

The remainder of this paper is structured as follows. Section 2 presents approaches to combine decision trees. Section 3 describes the system architecture and methodology. Section 4 presents results and discussion. Section 5 presents the conclusions and future work.

2 Combining Decision Trees

Combining decision tree models is a topic that has been studied with different approaches. Gorbunov and Lyubetsky [2] address the issue from a mathematical point of view, by formulating the problem of constructing a decision tree that is closest on average to a set of trees. Kargupta and Park [3] present an approach in which decision trees are converted to the frequency domain using the Fourier transform. The merging process consists on summing the spectra of each model and then transform the results back into to the decision tree domain.

Concerning data mining approaches, Provost and Hennessy [4,5] present an algorithm that evaluates each model with data from the other models to merge. The merged model is constructed from satisfactory rules, i.e., rules that are generic enough to be evaluated in the other models. A more common approach is the combination of rules derived from decision trees. The idea is to convert decision trees from two models into decision rules by combining the rules into new rules, reducing their number and finally growing a decision tree of the merged model. Parts of this process are presented in the doctoral thesis of Williams [6] and other researchers have contributed by proposing different ways of carrying out intermediate tasks, such as Andrzejak *et al.* [7], Hall *et al.* [8,9] or Bursteinas and Long [10]. While each approach is different, we identified a set of phases they share in common, described next.

In the first phase, a decision tree is transformed to a set of rules. Each path from the root to the leaves creates a rule with a set of possible values for variables and a class. These have been called "rules set" [8], "hypercubes" [10] or "sets of iso-parallel boxes" [7]. These designations arise from the fact that a variable can be considered as a dimension axis in a multi-dimensional space. The set of values (nominal or numerical) is the domain for each dimension and each rule defines a region. The representation of the regions is required for the next phase. It is worth noting that all regions are disjoint from each other and together cover the entire space.

In the second phase, the regions of both models are combined using a specific method. Andrzejak *et al.* [7] call this "unification" and propose a line sweep algorithm to avoid comparing every region of each model. It is commonly based on sorting the limits of each region and then analysing where merging can be done. However this method only applies to numerical variables. Hall *et al.* [8] compare all regions with each other. Bursteinas and Long [10] have a similar method but separate disjoint from overlapping regions. One potential problem is that combining regions can lead to a class conflict if overlapping regions have different classes. Andrzejak *et al.* [7] propose three strategies to assign a class to the merged region. The first assigns the class with the greatest confidence, the second, the one with the greater probability and a third strategy, which is the more complex, involves more passes over the data. Hall *et al.* [8] explore the issue in greater detail and propose further strategies, e.g., comparing distances to the boundaries of the variables. However, this approach seems suitable only for numerical variables. Bursteinas and Long [10] use a different strategy by retraining the model with examples for the conflicting class region. If no conflict arises, that class is assigned, otherwise the region is removed from the merged model.

The third phase attempts to reduce the number of regions and it is commonly referred to as "pruning". This is carried out to avoid having models with very complex rules. The most direct approach is to identify adjacent regions, i.e., regions sharing the same class and values of all variables except for one. If that variable is nominal, the values of both regions are included, otherwise the join is only possible if the limits overlap. To further reduce the rules set, Andrzejak *et al.* [7] developed a ranking system retaining only the regions with the highest relative volume and number of training examples. Hall *et al.* [8] only carry out this phase to eliminate redundant rules created during the removal of class conflicts. Bursteinas and Long [10] mention the phase but do not provide details on how it is performed.

The fourth phase consists in growing a decision tree from the decision regions representation. Andrzejak *et al.* [7] attempt to mimic the C5.0 algorithm using the values in the regions as examples. One problem with this method is that it is necessary to divide one region in two to perform the splitting, which increases their number, thus making the model more complex. Hall *et al.* [8] do not perform this phase and the merged model is represented as the set of regions. Bursteinas and Long [10] claim to grow a tree but do not describe the method.

3 Methodology

To carry out the experiments, a system with five processes was developed with the architecture presented in Fig. 1.

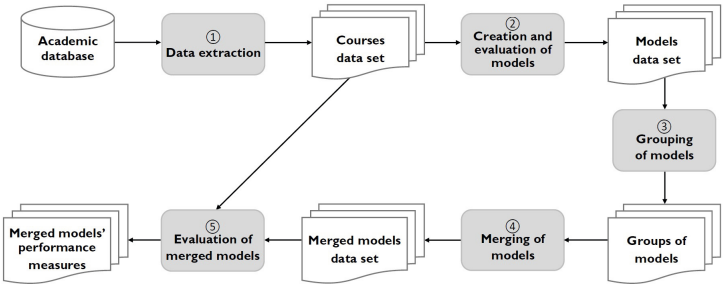

Fig. 1. System architecture

The first process creates the data sets (one for each course in the university) from the academic database. These contain enrollment data (Section 3.1). The second process creates decision tree models for each course, analyses them in order to determine the most important variables and evaluates them to assess their quality (Section 3.2). The third process groups the models according to different criteria (Section 3.3). The models in each group are then merged by the fourth process (Section 3.4). Finally the fifth process evaluates the merged models from a performance improvement point of view (Section 3.5).

3.1 Data Extraction

This process extracts data sets from the academic database of the university information system. The academic database stores a large amount of data on students, program syllabuses, courses, academic acts and assorted data related to a variety of sub-processes of the pedagogical process. The analysis done focuses on the academic year 2012/2013 with the extraction of 5779 course data sets (from 391 programmes), with the variables presented in Table 1.

3.2 Creation and Evaluation of Models

This process has two sub-processes: (1) the models for each course data set are trained and analysed in order to find out the most important variables for prediction; (2) the prediction quality of each model is evaluated.

Model Training and Analysis. The models are decision tree classifiers generated by C5.0 algorithm [11]. Decision trees have the characteristic of not requiring previous domain knowledge or heavy parameter tuning making them

Table 1. Variables for models

Variable	Remarks	Type
Age	age of student at the date of enrollment	Numerical
Sex	male, female	Nominal
Marital status	single, married, divorced, widower, . . .	Nominal
Nationality	first nationality of student	Nominal
Displaced	whether the student lived outside the Porto district	Boolean
Scholarship	whether the student has a scholarship	Boolean
Special needs	whether the student has disabilities	Boolean
Type of admission	type of application contest	Nominal
Type of student	regular, mobility, extraordinary	Nominal
Status of student	ordinary, employed, athlete, . . .	Nominal
Years of enrollment	# of academic years the student has enrolled in previously	Numerical
Delayed courses	# of courses the student should have completed	Numerical
Type of dedication	full-time, part-time	Nominal
Debt situation	whether there are fees due	Boolean
Approval	whether the student has passed or failed the course	Boolean

appropriate for both prediction, exploratory data analysis and are human interpretable. In this study, students are classified as having passed or failed a course. The C5.0 algorithm measures the importance of variable I_v by determining the percentage of examples tested in a node by that variable in relation to all examples (eq. 1).

$$I_v = \frac{\#examples\ tested\ by\ variable\ v}{\#examples} \tag{1}$$

Model Evaluation. Experimental setup uses k-fold cross-validation [12] with stratified sampling [13]. Failure is the positive class in this problem, i.e. it is the most important class, and thus, we use a suitable evaluation measure $F1$ [14].

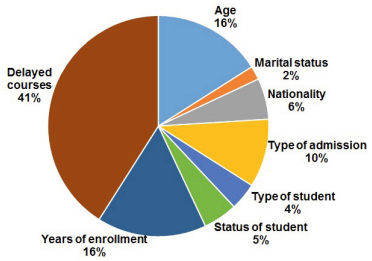

Fig. 2. Distribution of I_v in models

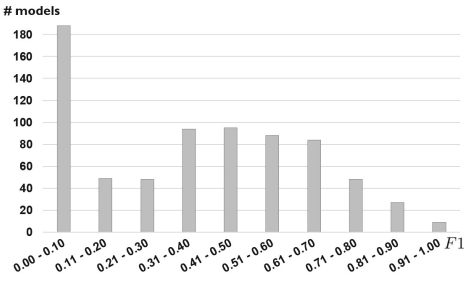

Fig. 3. Distribution of $F1$ in models

Training, analysis and evaluation of models is replicated for each course in the data set, however, models were created only for courses with a minimum of 100 students enrolled. This resulted in creating 730 models (12% of the 5779 courses). The variables used in the models are age, marital status, nationality, type of admission, type of student, status of student, years of enrollment, and delayed

courses. Fig. 2 shows the average importance I_v of each variable across all models. Delayed courses (41%) is the variable most often used, followed by age (16%) and years of enrollment (16%). The quality of the models varies significantly with only a quarter having $F1$ above 0.60, as presented in Fig. 3.

3.3 Grouping of Models

This process aims to group models according to specific criteria. We grouped models on the basis of scientific area, number of variables, variable importance, and a baseline group containing all models. For creating groups according to variable importance we used the k-means clustering algorithm [15], which created four groups (clusters) using only three of the most important variables identified in Section 3.2, namely age (I_a), years of enrollment (I_{yo}), and delayed courses (I_{dc}). Table 2 presents the four group sets and respective groups, specifying the number of models in each group.

Table 2. Groups sets of models

Group set criteria	Group	# models	%
Scientific areas	1: Architecture & Arts	47	6.44
	2: Computer Science	44	6.03
	3: Engineering	92	12.60
	4: Humanities	38	5.21
	5: Legal Sciences	48	6.58
	6: Mathematics	63	8.63
	7: Medicine	143	19.59
	8: Physical Sciences	98	13.42
	9: Social Sciences	117	16.03
	10: Sport Sciences	40	5.47
Number of variables	1: 0 variables	177	24.25
	2: 1 variable	219	30.00
	3: 2 variables	161	22.05
	4: 3 variables	92	12.60
	5: 4 variables	42	5.75
	6: 5 variables	35	4.79
	7: 6 variables	4	0.56
Importance of variables	1: I_a=17.40, I_{yo}=95.15, I_{dc}=37.56	116	15.89
	2: I_a=6.87, I_{yo}=7.75, I_{dc}=98.98	304	41.64
	3: I_a=97.91, I_{yo}= 6.83, I_{dc}=23.28	102	13.97
	4: I_a=0.06, I_{yo}= 0.00, I_{dc}= 0.00	208	28.50
None	1: Baseline	730	100.00

3.4 Merging of Models

The methodology to merge all models in a group set is done according to the experimental set-up presented in Fig. 4. For the process of merging models, each model must be represented as a set of decision regions. This can take the form of a *decision table*, in which each row is a decision region. Therefore, the first and second models are converted to decision tables and merged, yielding the model a_1, also in decision table form. Then the third model is also converted to a decision table and is merged with model a_1 yielding model a_2.

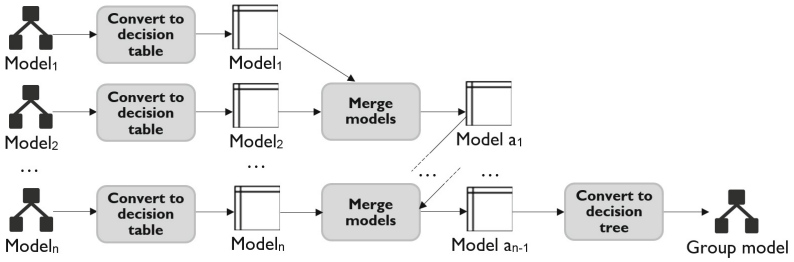

Fig. 4. Experimental set-up to merge models in a group set

The last merged model a_{n-1} is converted to the decision tree form and is evaluated against data of all courses in the group. Each one of these sub-processes and its tasks are detailed in the following sub-sections.

Conversion to Decision Table. A decision table is an alternative way of representing a decision tree. Each path from the top of the tree to the leaves defines a decision region, represented as a row in the decision table. Columns specify the class, weight and set of values of each variable. An example of conversion is presented in Fig. 5. A special case arises when the model is empty, which implies the existence of a single decision region covering the whole space.

```
AGE <= 20: y (95/9)
AGE > 20:
:...MARITAL_STATUS = divorced: n (1)
    MARITAL_STATUS = married: y (3)
    MARITAL_STATUS = single:
    :...DELAYED_COURSES > 0: n (86/28)
        DELAYED_COURSES <= 0:
        :...AGE > 23: y (5/2)
            AGE <= 23:
            :...AGE <= 21: y (16/5)
                AGE > 21: n (11/3)
```

Region #	Class	Weight	Variables		
			AGE	MARITAL_STATUS	DELAYED_COURSES
1	y	44	[0,20]		
2	n	1	[21,+∞)	divorced	
3	y	1	[21,+∞)	married	
4	n	40	[21,+∞)	single	[1,+∞)
5	y	2	[24,+∞)	single	0
6	y	7	21	single	0
7	n	5	[22,23]	single	0

Fig. 5. Example of conversion from decision tree to decision table

In the decision trees generated by the C5.0 algorithm, the leaf nodes show the number of examples used to create the split [11]. This number is used to measure the importance of the region associated with each leaf node. In a model with r decision regions, we define the *weight* of a decision region w_i as the proportion of examples in region i relative to the whole data set. It is a relative measure of the importance of a decision region in the context of a model.

Merge Models. The process of merging two models encompasses three sequential sub-processes: intersection, filtering and reduction as presented in Fig. 6.

Intersection is a sub-process to calculate the cross-product regions of two decision tables, resulting in a decision table for the merged model. The merged

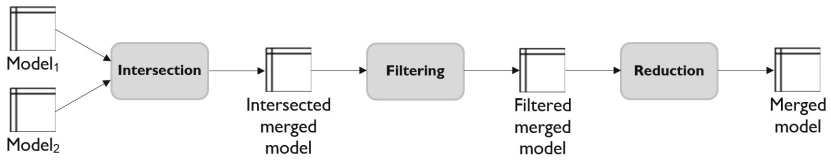

Fig. 6. Process of merging two models

model has the variables of both models. For each pair of regions, the set of values of each variable is intersected. The resulting *merged region* contains all the intersection sets relative to each variable. The intersected merged model is the set of all merged regions.

The intersection of values of each variable from both regions may have the following outcomes:

- If there are common values, then these are assigned to the variable in the merged region.
- If there are no common values for the variable being intersected in both regions, then they are considered *disjoint regions*, regardless of other variables in which the intersection set may not be empty.
- If the variable does not have a value or is not present in one of the models then the set of values in the other model is copied to the merged region (the absence of a variable in a model or its presence with no value for a specific region is considered as a neutral element, i.e., it can take any value without affecting the decision).

Each region of the intersected merged model must also have a weight so that the decision table has the same format as the original models. As weight measures the importance of a region, it is logical to assume that the most immediate choice is to consider the maximum weight of the original regions. However, for evaluation purposes, we think that it is interesting assess the impact on the model's performance by assigning the minimum weight. We also consider a third possibility which is to use the average value of the largest and smallest. To easily compare each possibility, we define a *weight attribution* parameter which can be set to be the *minimum*, *maximum* or *average* value of the original pair of regions, according to each case.

The class to assign to the merged region is straightforward if both regions have the same class, otherwise the class conflict problem arises. The strategies for its resolution are controlled by a *conflict class resolution* parameter. This can be set to assign the same class as the region with *minimum weight* or with *maximum weight*. Fig. 7 presents an example of intersection of decision tables, which illustrates the weight and class assignment to each merged regions. In this example the weight attribution parameter is set to *average* and conflict class resolution parameter to *maximum weight*.

Filtering is the sub-process to remove disjoint regions from the intersected merged model yielding the filtered merged model. Disjoint regions have to be

Model 1				Model 2					Merged model					
Region #	Class	Weight	Variables	Region #	Class	Weight	Variables		Region #	Class	Weight	Variables		Region status
			DELAYED_COURSES				AGE	DELAYED_COURSES				DELAYED_COURSES	AGE	
1	y	82	[0,1]	**1**	y	75	[0,20]		**1**	y	79	[0,1]	[0,20]	OK
1	y	82	[0,1]	**2**	n	10	[21,+∞)	[1,+∞)	**2**	y	46	1	[21,+∞)	OK
1	y	82	[0,1]	**3**	y	13	[21,25]	0	**3**	y	48	0	[21,25]	OK
1	y	82	[0,1]	**4**	n	3	[26,+∞)	0	**4**	y	43	0	[26,+∞)	OK
2	y	15	[2,5]	**1**	y	75	[0,20]		**5**	y	45	[2,5]	[0,20]	OK
2	y	15	[2,5]	**2**	n	10	[21,+∞)	[1,+∞)	**6**	y	13	[2,5]	[21,+∞)	OK
2	y	15	[2,5]	**3**	y	13	[21,25]	0	**7**	y	14	∅	[21,25]	Disjoint
2	y	15	[2,5]	**4**	n	3	[26,+∞)	0	**8**	y	9	∅	[26,+∞)	Disjoint
3	n	2	[6,+∞)	**1**	y	75	[0,20]		**9**	y	39	[6,+∞)	[0,20]	OK
3	n	2	[6,+∞)	**2**	n	10	[21,+∞)	[1,+∞)	**10**	n	6	[6,+∞)	[21,+∞)	OK
3	n	2	[6,+∞)	**3**	y	13	[21,25]	0	**11**	y	8	∅	[21,25]	Disjoint
3	n	2	[6,+∞)	**4**	n	3	[26,+∞)	0	**12**	n	3	∅	[26,+∞)	Disjoint

Fig. 7. Intersection of decision tables

Region #	Class	Weight	Variables	
			DELAYED_COURSES	AGE
1	y	25	[0,1]	[0,20]
2	y	14	1	[21,+∞)
3	y	15	0	[21,25]
4	y	13	0	[26,+∞)
5	y	14	[2,5]	[0,20]
6	y	4	[2,5]	[21,+∞)
7	y	12	[6,+∞)	[0,20]
8	n	2	[6,+∞)	[21,+∞)

Fig. 8. Filtering a merged decision table

Region #	Class	Weight	Variables		Obs.
			AGE	DELAYED_COURSES	
1	y	51	[0,20]		1;5;7
2	y	18	[21,+∞)	[1,5]	2;6
3	y	28	[21,+∞)	0	3;4
4	n	2	[21,+∞)	[6,+∞)	8

Fig. 9. Reducing a merged decision table

removed because they relate to pairs of regions of the original models that have no values in common on at least one variable present in both. Removing regions implies recalculating the weight of each region that remains to obtain a total of 100%. As weights are rounded to integers, the weight of a region less than 0.5 is rounded to zero. Therefore, a possible consequence of the filtering sub-process is that regions with zero weight may arise. Fig. 8 shows the result of filtering out the disjoint region of the merged model of Fig. 7.

The filtered merged model can be empty if all regions of the intersected merged model are disjoint. In such case, the two original models are considered *not mergeable* and the merging process is halted. It is then restarted using the last successfully merged model as model 1 and the next model from the group set as model 2.

Reduction is a sub-process to limit the number of regions in the filtered merged model, to obtain a simpler model. The regions are examined to find out which can be merged. This is possible when a set of regions have the same class and all variables have equal values except for one. In the case of nominal variables, reduction consists on the union of values of that variable from all regions. In the case of numerical variables, currently reduction is only performed if the intervals are contiguous (this procedure will be improved to allow reduction even with non-contiguous intervals).

The weight of the resulting region is the sum of the weights of the regions that are joined. After all regions have been subjected to reduction, they are again examined and those that have zero weight are removed. Another consequence of the reduction is that there may exist variables with the same value in all decision regions. The columns for these variables are removed from the table. Fig. 9 shows the result of reducing the decision table of Fig. 8. The reduction sub-process results in the last successfully merged model of the group so far.

Conversion to Decision Tree. The last merged model of the group is converted to the decision tree representation, yielding the *group model*. For this purpose, examples are generated randomly, bounded by the limits of each variable from the decision table and submitted to the C5.0 algorithm to train a model. Each decision region provides examples which corresponds to a combination of the set of values of each variable with the set of values of other variables and the assigned class to the region. The set of values of numerical variables are bounded by two limits. If the upper limit is missing ($+\infty$), then the maximum observed value from all courses data sets is used. When the lower limit is zero, the lowest observed value across all courses data sets is used (e.g., age). These values are collected as part of the final task of the data extraction process (Section 3.1).

The generation of examples can be controlled with the *examples for numerical variables* parameter. Setting to *limits* only generates two examples (one for each limit) while *samples* generates examples between the limits with a step of 5 (our initial approach used all values but was infeasible due to memory limitations). The weight of the region can also influence the number of examples generated, being controlled by a *weight examples* parameter. If active, the number of generated examples by each region is multiplied by the weight of that region. Generating some examples more frequently than others is a way to preserve the importance of a region over others with less weight in the resulting decision tree.

3.5 Merged Models Evaluation

This process evaluates a group model, using the experimental set-up presented in Fig. 10. After this process, each model has two measures of performance, one ($F1$) from the model obtained from its own data and another ($F1_g$) from the group model. Hence, $\Delta F1$ (eq. 2) allows us to measure changes in predictive power:

$$\Delta F1 = F1_g - F1 \tag{2}$$

If $\Delta F1$ is greater than zero, then there is an improvement in predictive performance by using the group model. If equal to zero, there is no improvement in predictive performance. If lower than zero, then there is loss of predictive performance relative to the original model.

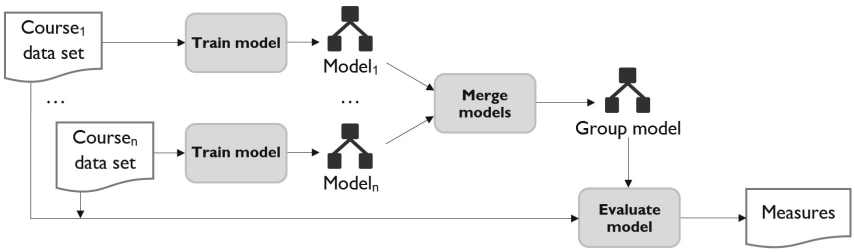

Fig. 10. Experimental set-up to evaluate a group model

The performance of a group model (eq. 3) is the average of $\Delta F1$ of all models of the group. We define the *merging score* (M) of a group (eq. 4) as the number of models that was possible to merge (m) divided by the number of pairs of models in the group ($n-1$).

$$\overline{\Delta F1} = \frac{\sum_{i=1}^{n} \Delta F1_i}{n} \quad (3) \qquad M = \frac{m}{n-1} \quad (4)$$

For group set evaluation, we normalized $\overline{\Delta F1}$ of all groups by the number of models (n_k) in each group (eq. 5). Likewise, the merging score of a group set (eq. 6) is the average merging score of all groups normalized by the number of models (g is the number of groups in a group set).

$$\overline{\Delta F1}_{gs} = \frac{\sum_{k=1}^{g} \overline{\Delta F1}_k \times n_k}{\sum_{k=1}^{g} n_k} \quad (5) \qquad \overline{M}_{gs} = \frac{\sum_{k=1}^{g} M_k \times n_k}{\sum_{k=1}^{g} n_k} \quad (6)$$

4 Results

A set of 24 experiments were run to merge models with the results presented in Table 3. Each experiment is a combination of values of the parameters *weight attribution* (wa), *conflict class resolution* (ccr), *examples for numerical variables* (efnv) and *weight examples* (we). This allows to compare the impact of each parameter on the average merging score ($\overline{\overline{M}}$), improvement in prediction ($\overline{\overline{\Delta F1}}$) and across each group set: scientific areas ($\overline{\Delta F1}_{SA}$), number of variables ($\overline{\Delta F1}_{\#v}$), importance of variables ($\overline{\Delta F1}_{I_v}$), and baseline ($\overline{\Delta F1}_B$).

Merging Score. The average merging score is 76%, which hardly changes throughout experiments. This implies that none of the parameters has a significant role in the ability to merge models. Fig. 11 shows the average merging score of all experiments across groups in all group sets. The idea behind creating groups of models is to try to bring together models that are similar the most, i.e.,

Table 3. Results of experiments

#	wa	ccr	efnv	we	$\overline{\overline{M}}$	$\overline{\Delta F1}_{SA}$	$\overline{\Delta F1}_{\#v}$	$\overline{\Delta F1}_{I_v}$	$\overline{\Delta F1}_{B}$	$\overline{\overline{\Delta F1}}$
1	min	min	limits	no	0.76	0.02	-0.04	0.03	-0.27	-0.06
2	min	min	limits	yes	0.76	0.03	-0.03	0.02	0.04	0.01
3	min	min	samples	no	0.76	0.01	-0.05	0.00	0.06	0.00
4	min	min	samples	yes	0.76	0.01	-0.03	-0.06	0.05	-0.01
5	min	max	limits	no	0.76	-0.02	-0.05	-0.18	0.00	-0.06
6	min	max	limits	yes	0.76	-0.08	-0.05	-0.22	0.00	-0.09
7	min	max	samples	no	0.76	-0.01	-0.04	-0.18	0.00	-0.06
8	min	max	samples	yes	0.76	-0.04	-0.04	-0.20	0.00	-0.05
9	max	min	limits	no	0.75	0.00	-0.06	-0.07	-0.03	-0.05
10	max	min	limits	yes	0.75	-0.02	-0.07	-0.07	0.06	-0.05
11	max	min	samples	no	0.75	-0.03	-0.04	-0.09	-0.03	-0.05
12	max	min	samples	yes	0.75	-0.02	-0.06	-0.13	-0.03	-0.06
13	max	max	limits	no	0.76	0.02	-0.07	-0.08	0.00	-0.03
14	max	max	limits	yes	0.76	0.02	-0.07	-0.07	0.00	-0.03
15	max	max	samples	no	0.76	0.02	-0.08	-0.08	0.00	-0.03
16	max	max	samples	yes	0.76	0.02	-0.07	-0.07	0.00	-0.03
17	avg	min	limits	no	0.76	0.01	-0.03	0.00	0.00	-0.01
18	avg	min	limits	yes	0.76	0.01	-0.04	0.00	0.00	-0.01
19	avg	min	samples	no	0.76	0.03	-0.06	0.00	0.00	-0.01
20	avg	min	samples	yes	0.76	0.02	-0.05	0.00	0.00	-0.01
21	avg	max	limits	no	0.76	0.01	-0.08	-0.06	0.00	-0.03
22	avg	max	limits	yes	0.76	0.02	-0.08	-0.06	0.00	-0.03
23	avg	max	samples	no	0.76	0.03	-0.11	-0.06	0.00	-0.03
24	avg	max	samples	yes	0.76	0.03	-0.08	-0.05	0.00	-0.03
				Avg.	0.76	0.01	-0.06	-0.07	-0.05	

with less likelihood of their merging resulting in disjoint regions. We observe that different group sets affect the merging score. This is particularly noticeable in grouping by the scientific areas in which group #4 (Humanities) has the highest merging score (92%) while group #7 (Medicine) has the lowest (52%). Grouping by number of variables shows that it is not possible to merge models with no variables (group #1), however, from 1 variable onward, it is always possible to merge all models into a single group model. Grouping by variable importance always allow full merging, except in the last group (probably because the models are less similar). The baseline group set has the average value of the experiments (76%). Results show that, from a merging ability perspective, merging by scientific areas is not necessarily the best way to group models while number of variables and importance of variables seem to be more suitable approaches.

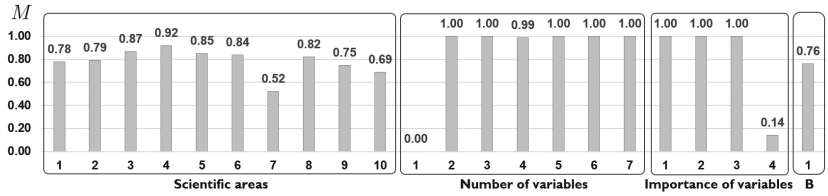

Fig. 11. Merging score distribution by groups of models

Average Improvement in Prediction. Experiments with best results in improvement in prediction are #2 and #3, although average improvement is not very significant (0.01). In all other experiments there is loss of average predictive power, with experience #6 with particularly poor results (-0.09). There is no apparent correlation between any of the four parameters and improvement in predictive performance.

Fig. 12 presents the distribution for experiment #2. Grouping by scientific areas has an average improvement of 0.03, with group #7 (Medicine) showing an improvement of 0.14 while group #3 (Engineering) has a loss of 0.21. Grouping by number of variables, has an average loss of 0.03. Grouping by variable importance has an average loss of 0.02 and it is the group set with highest variance. The baseline group set has an average improvement of 0.04 (the highest of all). Table 3 also shows the average $\overline{\Delta F1}$ across experiments for each group set. Grouping by scientific areas is the only one that has a positive global improvement in prediction with 0.01. The baseline group set has less variation and only had loss in experiment #1.

These results show that, contrary to what happened with the merging score, grouping models by scientific areas yields interesting results in terms of improvement in predictive power. This may be indicative that models obtained from courses with more similar content are more susceptible to generalize knowledge than the ones in which similarity arises from features of the models themselves (such as the number of variables or its importance).

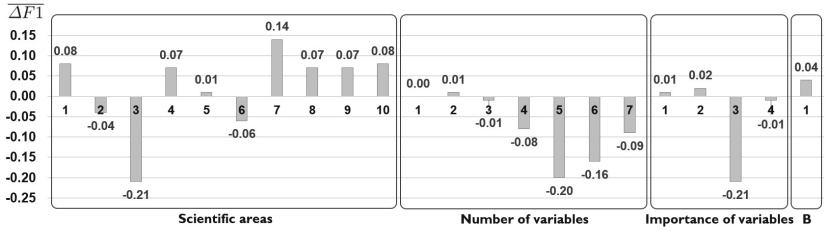

Fig. 12. Average improvement in prediction by groups of models for experiment #2

5 Conclusions and Future Work

This methodology presents an approach to merge decision trees. Its goal is to address the main problems commonly encountered while solving this problem, namely the preservation of the importance of some decision rules throughout all merging process and how to deal with the problem of class conflict of overlapping rules. The generation of the merged decision tree also presents challenges.

To study the impact of these problems, we define four parameters affecting merging and carried out experiments combining them. The case study are decision trees to predict failure of students in courses at the University of Porto. Results show that, on average, it is possible to merge 76% of models in a group. Grouping by scientific areas of courses is the best way to combine courses as the resulting model remains generic without loosing predictive quality. Tuning the four parameters improved predictions in a few cases.

Directions for future work point to improving the process with more elaborate strategies for class conflict resolution. Another important issue is the merge order of models in each group, which has yet to be studied in detail.

Acknowledgments. This work is funded by projects "NORTE-07-0124-FEDER-000059" and "NORTE-07-0124-FEDER-000057", financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

References

1. Dekker, G., Pechenizkiy, M., Vleeshouwers, J.: Predicting students drop out: a case study. In: 2nd International Educational Data Mining Conference (EDM 2009), pp. 41–50 (2009)
2. Gorbunov, K.Y., Lyubetsky, V.A.: The tree nearest on average to a given set of trees. Problems of Information Transmission 47, 274–288 (2011)
3. Kargupta, H., Park, B.: A fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments. IEEE Transactions on Knowledge and Data Engineering 16, 216–229 (2004)
4. Provost, F.J., Hennessy, D.N.: Distributed machine learning: scaling up with coarse-grained parallelism. In: Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology, vol. 2, pp. 340–347 (January 1994)
5. Provost, F., Hennessy, D.: Scaling up: Distributed machine learning with cooperation. In: Proceedings of the 13th National Conference on Artificial Intelligence, pp. 74–79 (1996)
6. Williams, G.J.: Inducing and Combining Multiple Decision Trees. PhD thesis, Australian National University (1990)
7. Andrzejak, A., Langner, F., Zabala, S.: Interpretable models from distributed data via merging of decision trees. In: 2013 IEEE Symposium on Computational Intelligence and Data Mining, CIDM (April 2013)
8. Hall, L., Chawla, N., Bowyer, K.: Combining decision trees learned in parallel. Working Notes of the KDD 1997 Workshop on Distributed Data Mining, pp. 10–15 (1998)
9. Hall, L., Chawla, N., Bowyer, K.: Decision tree learning on very large data sets. In: IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, pp. 2579–2584 (1998)
10. Bursteinas, B., Long, J.: Merging distributed classifiers. In: 5th World Multiconference on Systemics, Cybernetics and Informatics (2001)
11. Kuhn, M., Weston, S., Coulter, N., Quinlan, R.: C50: C5.0 Decision Trees and Rule-Based Models. R package version 0.1.0-16 (2014)
12. Stone, M.: Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society: Series B 36(2), 111–147 (1974)
13. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Conference on AI (IJCAI), pp. 1137–1145. Morgan Kaufmann, San Mateo (1995)
14. Chinchor, N.: MUC-4 Evaluation Metrics. In: Proceedings of the 4th Message Understanding Conference (MUC4 1992), pp. 22–29. Association for Computational Linguistics (1992)
15. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2011)