A Proposal for Simplifying Explanations from Ensembles of Symbolic Classifiers

Flavia Cristina Bernardini[1], Maria Carolina Monard[2], and Ronaldo C. Prati[3]

[1] Departamento de Ciência e Tecnologia — RCT
Pólo Universitário de Rio das Ostras — PURO
Universidade Federal Fluminense — UFF
Rio das Ostras, RJ, Brasil — fcbernardini@vm.uff.br
[2] Departamento de Ciência da Computação - SCC
Instituto de Ciências Matemáticas e Computação – ICMC
Universidade de São Paulo – USP
São Carlos, SP, Brasil — mcmonard@icmc.usp.br
[3] Centro de Matemática, Computação e Cognição – CMCC
Universidade Federal do ABC – UFABC
Santo André, SP, Brasil — ronaldo.prati@ufabc.edu.br

Abstract. Data mining applications generally use learning algorithms in order to induce knowledge. To accomplish this task, these algorithms should be able to operate with massive data sets. Several techniques, such as data sampling, can be used to scale up learning algorithms to deal with large datasets. Using data sampling, learning algorithms can be applied to small samples of the original dataset, and the individual classifiers can then be combined into an ensemble which, in numerous situation, can be more accurate than the individual classifiers. However, ensembles often lack the facility to explain their decisions. This work explores a method to offer a concise explanation of ensembles decisions whenever the ensembles are composed by a combination of symbolic classifiers. Different methods used to construct ensembles are also described.

Keywords: Symbolic Machine Learning, Symbolic Ensembles, Explanation of Symbolic Ensembles.

1 Introduction

Supervised machine learning aims to induce classifiers having good accuracy. However, in some domains such as medicine, economy and others, it is important to provide to the user the knowledge that has been used to classify new cases. In other words, the classifier must be able to explain its decisions. In these cases, symbolic machine learning has advantages over the so called "black-box" learning, such as Neural Networks and Support Vector Machine. This is due to the fact that the models induced by symbolic learning algorithms can be easily interpreted by domain experts. In general, symbolic learning algorithms can handle well datasets of medium/large size, inducing classifiers with high accuracy. However, this is not the case in data mining applications where very large

datasets are considered. A possible solution to overcome this problem is the use of ensembles. Ensembles consist of a set of classifiers whose individual predictions are somehow combined to classify new instances [8,9]. The problem with very large datasets can be tackle by splitting the dataset into small subsets and then using an ensemble to combine the classifiers induced on each subset of the dataset. Furthermore, the combination procedure beneath ensembles generally produces more accurate classifiers than single classifiers do.

An often cited problem regarding ensembles is that standard ensemble approaches behave like "black-boxes", *i.e.*, they cannot offer an explanation related to the classification of new instances, as symbolic classifiers do [2,5,6,10]. As stated earlier, there are domains where these explanations are mandatory. For instance, in some countries, when a bank does not approve a client's credit proposal, they should explain to the client why the proposal was denied. In these cases, standard ensemble approaches could not be applied. In [4], we proposed an approach for constructing ensembles of symbolic classifiers, where several ways of combining the classifiers into an ensemble were proposed and experimentally evaluated considering the error rate of the corresponding symbolic ensembles. Furthermore, this approach enables the user to analyze the knowledge (rules) used by the ensemble to classify new cases. However, the explanation shown to the user usually contains many rules which are often redundant and/or too specialized.

In this paper we extend this work by proposing a simple method to simplify the explanation given by symbolic ensembles to the user/expert domain, by providing a more concise and comprehensible explanation than simply showing all the rules used by the ensemble to classify a new case. After the ensemble classifies a new instance, our proposed approach works by identifying some redundant rules, which are not shown to the user when (s)he asks for an explanation.

This paper is organized as follows: Section 2 describes related work in ensemble approaches. Section 3 presents concepts needed for a better comprehension of this work, as well as the notation used. Section 4 describes an approach previously proposed to construct ensembles of symbolic classifiers. Section 5 presents the method proposed to simplify the explanation offered to the user by a symbolic ensemble, related to the classification given by the ensemble to a new instance. Section 6 briefly describes the error rate of several symbolic ensembles constructed using the three voting mechanisms already proposed, as well as the performance of the explanation simplification algorithm proposed in this work, considering new instances which have been previously classified by these ensembles. Finally, Section 7 concludes this work.

2 Related Work

Ensembles are increasingly gaining acceptance in the data mining community. Apart from showing a significant improvement in accuracy, this is also due to their potential for on-line classification of large databases that do not fit into memory. There are different ways in which ensembles can be generated and

the resulting output combined to classify new instances. In general, methods to construct ensembles can be divided into two sub-tasks [9]. The first one consists of generating a set of base-classifiers. The second one consists of deciding how to combine the classifications of the base-classifiers to classify new instances.

Popular approaches to generate ensembles include altering the set of instances used for training in order to construct the base-classifiers. These approaches use techniques such as bagging [5], boosting [10], wagging [2] and others. To reach improvements in accuracy, these methods generally rely on a large number of base-classifiers so that a wrong classification given by some base-classifier can be averaged out by the others. On the other hand, other papers focus on the combination of classifiers (second sub-task), inducing the base-classifiers by using different learning algorithms [11], as in stacking, which construct another classifier to combine the base-classifiers decisions [13]. However, unlike symbolic classifiers, ensembles can be considered as "black-box" classifiers, since they are not able to explain their classification decisions on new examples, like symbolic classifiers do. However, as most of the ensembles proposed in the literature tend to focus on the classification error rate, they frequently make use of black-box base-classifiers. Although these kind of base-classifiers may improve the classification error rate, they are a complicating factor in case an explanation should be provided to the user.

In what follows, after section describing definitions and notation, we first describe the approach we have proposed in previous work [4] to construct ensembles using symbolic classifiers and three voting mechanisms, which enables the ensemble to explain its decisions to the user. This approach achieved meaningful results using only a small number of base-classifiers [4]. Afterwards a method to simplify the explanation given by symbolic ensembles, which extends this work, is described.

3 Definitions and Notation

A training *dataset* T is a set of N classified instances $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ for some unknown function $y = f(\mathbf{x})$. The $\mathbf{x}_i$ values are typically vectors of the form $(x_{i1}, x_{i2}, ..., x_{im})$ whose components are discrete or real values, called *features* or *attributes*. Thus, x_{ij} denotes the value of the j-th feature X_j of $\mathbf{x}_i$. In what follows, the i subscript will be dropped when implied by the context. For classification purposes, the y values can assume any value from a discrete set of N_{Cl} classes, *i.e.* $y \in \{C_1, C_2, ..., C_{N_{Cl}}\}$. Given a set $S \subseteq T$ of training examples, a learning algorithm induces a *classifier* $\mathbf{h}$, which is a hypothesis about the true unknown function f. Given a new instance $\mathbf{x}$, $\mathbf{h}$ predicts the corresponding y values.

In this work we consider that a *symbolic classifier* is a classifier whose description language can be transformed into a set of N_R unordered or disjoint rules, *i.e.* $\mathbf{h} = \{R_1, R_2..., R_{N_R}\}$. Recall that most classification rule learning algorithms belong to one of two families, namely separate-and-conquer and divide-and-conquer algorithms. Algorithms from the first family generally use an iter-

ative greedy set-covering algorithm to search in each iteration for the best rule. When inducing an unordered set of rules, only correctly covered examples are removed from the training set. However, when inducing an ordered set of rules, correctly as well as incorrectly covered examples are removed from the training set. Thus, in the unordered case rules can be interpreted in isolation, while in the ordered set a rule has no meaning by itself, as the semantic of a fired rule must take into account all previous non fired rules. This process is repeated in the remaining examples until all examples have been covered or some stopping criterion is met. Then, a classifier is built gathering the rules to form an ordered rule list (or decision list) in case all covered examples were removed in each iteration, or to form an unordered rule set in case only examples correctly covered were removed in each iteration. On the other hand, algorithms from the second family — divide-and-conquer — construct a global classifier using a top-down strategy to consecutively refine a partial theory. Generally, the classifier is expressed as a decision tree which can be written as a set of disjoint unordered rules.

A *complex* is a disjunction of conjunctions of feature tests in the form of X_i *op* *Value*, where X_i is a feature name, *op* is an operator in the set $\{=, \neq, <, \leq, >, \geq\}$ and *Value* is a valid X_i feature value.

A *rule* R assumes the form **if** B **then** H or symbolically $B \rightarrow H$, where H stands for the head, or rule *conclusion*, and B for the body, or rule *condition*. H and B are both complexes with no features in common. In a *classification rule*, head H assumes the form class $= C_i$, where $C_i \in \{C_1, ..., C_{N_{Cl}}\}$.

The *coverage* of a rule is defined as follows: considering a rule $R = B \rightarrow H$, instances that satisfy the B part compose the *covered set* of R, called B set; in other words, these instances are *covered* by R. Instances that satisfy both B and H are *correctly covered* by R, and these instances belong to set $B \cap H$. Instances satisfying B but not H are *incorrectly covered* by the rule, and belong to set $B \cap \overline{H}$. On the other hand, instances that do not satisfy the B part are *not covered* by the rule, and belong to set $\overline{B}$.

Given two classification rules $R_i = B_i \rightarrow H_i$ and $R_j = B_j \rightarrow H_j$ with identical headers ($H_i = H_j$), it is simple to verify whether R_i is a generalization of R_j, *i.e* if R_i *subsumes* R_j, or whether R_i is a specialization of R_j, *i.e* if R_j *subsumes* R_i, by considering the set of feature tests in the bodies of these rules. Then, R_i is a generalization of R_j, or R_j is a specialization of R_i iff $A_i \Subset A_j$, where $\Subset$ indicates that A_i is a subset of A_j, *i.e.* all feature test which are in A_i are also in A_j, or tests which takes into account identical features and operators in the set $\{<, \leq, >, \geq\}$, as shown in the following example. Considering the following two classification rules:

$R_i =$ **If** $at_1 = 2$ **and** $at_2 \geq 4$ **then** *class* $= +$

and

$R_j =$ **If** $at_1 = 2$ **and** $at_2 > 4$ **and** $at_3 = 5$ **then** *class* $= +$

then

$A_i = \{at_1 = 2, at_2 \geq 4\}$
and
$A_j = \{at_1 = 2, at_2 > 4, at_3 = 5\}$.

In this case, R_i is a generalization of R_j, or R_j is a specialization of R_i, as $A_i \trianglelefteq A_j$. This is because $at_1 = 2$ is present in both rules and, as $at_2 \geq 4$ in A_i and $at_2 > 4$ in A_j, the operator "$\geq$" is a generalization of the operator "$>$". Analogously, the operator "$\leq$" is a generalization of "$<$".

Consider that $\mathbf{R}_{expl} = \{R_1, ..., R_{N_R}\}$ is the set of all individual classifiers' rules that correctly cover a new instance $\mathbf{x}$, *i.e.* fired rules from the base-classifiers that participate in the final (combined) ensemble classification. Then, the explanation mechanism should discard from $\mathbf{R}_{expl}$ all rules that are specialization of other rules. This simple idea is used in this work to simplify the explanation of symbolic ensembles, as explained in Section 5.

4 Constructing Ensembles of Symbolic Classifiers

The approach proposed in [4] for constructing ensembles of symbolic classifiers is divided into two phases. In the first phase, a set of base-classifiers is constructed in the following way: let L be the number of base-classifiers to be induced, given a training dataset S. First of all, L samples $S_1, ..., S_L$ without substitution are extracted from S. Each sample is used as an input to a symbolic learning algorithm, inducing L classifiers $\mathbf{h}_1, ..., \mathbf{h}_L$. Different symbolic learning algorithms can be used in each of the L samples. Afterwards, given a new instance (example) $\mathbf{x}$ to be classified, the individual decisions of the set of L classifiers must be combined to output its label. Figure 1 illustrates the method where $Combine(\mathbf{h}_1(\mathbf{x}), ..., \mathbf{h}_L(\mathbf{x}))$ constitutes the symbolic ensemble $\mathbf{h}^*(\mathbf{x})$.

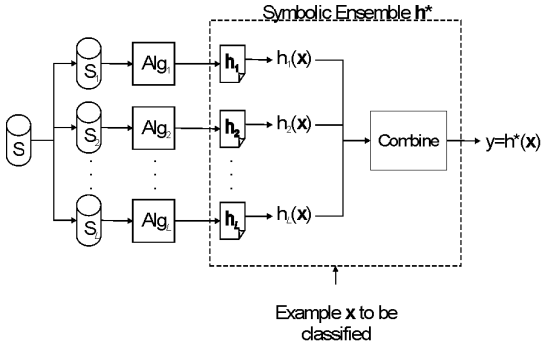

Fig. 1. A method for constructing ensembles of classifiers

The use of symbolic classifiers enables us to explore different ways to classify $\mathbf{x}$ using the L base-classifiers, as shown in [4]. In this work, each base-classifier is responsible for classifying $\mathbf{x}$, and the following three different implementations of the $Combine(\mathbf{h}_1(\mathbf{x}), ..., \mathbf{h}_L(\mathbf{x}))$ function were considered in order to construct the final ensemble $\mathbf{h}^*$. The Unweighted Voting method was based on the bagging technique for constructing ensembles of classifiers [5]; while the Weighted by Mean and Weighted by Mean and Standard Error Voting methods were inspired in the boosting technique [10].

1. **Unweighted Voting – UV**: the class label of $\mathbf{x}$ is the one that receives more votes from the L classifiers;
2. **Weighted by Mean Voting – WMV**: the $\mathbf{x}$ class label given by each classifier is weighted using the classifier's mean error rate $m\_err(\mathbf{h}_i)$, and the class label of $\mathbf{x}$ is the one having maximum total weight from the L classifiers:

$$WMV(\mathbf{x}, C_v) = \max_{C_i \in \{C_1, ..., C_{N_{Cl}}\}} \sum_{l=1}^{L} g(\mathbf{h}_l(\mathbf{x}), C_i)$$

where

$$g(\mathbf{h}_l(\mathbf{x}), C_i) = \begin{cases} \lg((1 - m\_err(\mathbf{h}_l))/m\_err(\mathbf{h}_l)) \\ \quad \text{if } \mathbf{h}_l(\mathbf{x}) = C_i, \\ 0 \text{ otherwise.} \end{cases}$$

3. **Weighted by Mean and Standard Error Voting – WMSV**: similar to the previous one, but also considering the standard error $se\_err(\mathbf{h}_i)$) of the classifier's mean error rate to estimate the corresponding weight:

$$WMSV(\mathbf{x}, C_v) = \max_{C_i \in \{C_1, ..., C_{N_{Cl}}\}} \sum_{l=1}^{L} g(\mathbf{h}_l(\mathbf{x}), C_i)$$

where

$$g(\mathbf{h}_l(\mathbf{x}), C_i) = \begin{cases} \lg((1 - m\_err(\mathbf{h}_l))/m\_err(\mathbf{h}_l)) \\ + \lg((1 - se\_err(\mathbf{h}_l))/se\_err(\mathbf{h}_l)) \\ \quad \text{if } \mathbf{h}_l(\mathbf{x}) = C_i, \\ 0 \text{ otherwise.} \end{cases}$$

Voting method 1 (UV) is a straightforward voting mechanism. Methods 2 (WMV) and 3 (WMSV) aim to improve method 1. To this end, method 2 (WMV) favors hypotheses having lower mean error rates. In similar fashion, method 3 (WMSV) favours hypotheses having lower mean error rates as well as lower standard error rates. In other words, the aim of the function $g(\mathbf{h}_l(\mathbf{x}), C_i)$, where lg is the logarithmic function, is to increment accordingly the class weight whenever the error rate and/or the standard error are less than 50%, and to decrement it otherwise. In case the error rate and/or the standard error are zero, a maximum system defined weight is considered. These methods were implemented in

a computational system called Ensemble Learning Environment (ELE) [3]. Furthermore, the method proposed in this work to improve the explanation ability of the symbolic ensemble, described next, was also implemented into the system ELE.

5 Explanation Construction

Given an ensemble of symbolic classifiers $\mathbf{h}^*$, constructed using any of the three $Combine(\mathbf{h}_1(\mathbf{x}), ..., \mathbf{h}_L(\mathbf{x}))$ functions defined in the previous section, the class of a new instance $\mathbf{x}$ is given by $\mathbf{h}^*(\mathbf{x})$. Let $\mathbf{R}_{expl} = \{R_1, ..., R_{N_R}\}$ be the set of rules that correctly cover $\mathbf{x}$, in other words, $\mathbf{R}_{expl}$ contains all fired rules which predict the same class as the one predicted by $\mathbf{h}^*(\mathbf{x})$. The rules in $\mathbf{R}_{expl}$ are an explanation of the class assigned by $\mathbf{h}^*$ to $\mathbf{x}$.

However, $\mathbf{R}_{expl}$ is not an appropriate explanation to be shown to the user, since it may contain redundant rules as well as rules that are a specialization of others rules in $\mathbf{R}_{expl}$. Thus, the idea is to construct a reduced set of rules $\mathbf{R}^*_{expl}$ from $\mathbf{R}_{expl}$, using the idea described in Section 3, where some rules which are specialization of other rules are removed. In order to accomplish this task, rules in $\mathbf{R}_{expl}$ are analyzed in pairs and, for all pair of rules in $\mathbf{R}_{expl}$ it is verified which rules are generalization of other rules, such that only the more general rules are inserted in $\mathbf{R}^*_{expl}$ and the specialized rules are discarded. At the end of this process, $\mathbf{R}^*_{expl}$ contains a simplified explanation of the class assigned to $\mathbf{x}$ by the ensemble $\mathbf{h}^*$.

The algorithm to find the final explanation $\mathbf{R}^*_{expl}$ works as follows. The body B_e of each rule $R_e \in \mathbf{R}_{expl} = \{R_1, ..., R_{N_R}\}$ consists of a conjunction of feature tests $ft_1 \wedge ... \wedge ft_b$, where each ft_i is a feature test of the type X_i *op Value*. For each $R_e \in \mathbf{R}_{expl}$, a set of its corresponding feature tests $A_e = \{ft_1, ..., ft_b\}$ is constructed, as well as the superset, called $\mathbf{A}_{expl}$, containing all corresponding A_e, *i.e.*, $\mathbf{A}_{expl} = \{A_1, ..., A_{N_R}\}$. To simplify the explanation, all pairs of subsets $(A_i, A_j), i \neq j$, are analyzed; (a) if A_i is a subset of A_j then A_i is removed from $\mathbf{A}_{expl}$; or (b) if A_j is a subset of A_i then A_j is removed from $\mathbf{A}_{expl}$.

Observe that $A_i \trianglelefteq A_j$ not only considers that $A_i \trianglelefteq A_j$ if all $ft \in A_i$ are also in A_j, but also considers tests on same features tests and operators in the set $\{<, \leq, >, \geq\}$ as explained in Section 3.

After analyzing all pairs (A_i, A_j) in $\mathbf{A}_{expl}$, $\mathbf{R}^*_{expl}$ is simply the set of rules whose bodies are given by the remaining A_i in the final $\mathbf{A}_{expl}$, and the class of these rules is given by $\mathbf{h}^*(\mathbf{x})$.

6 Experiments and Results

In [4] we presented, among other results, the results related to error rates assessed using 10-fold stratified cross validation, obtained using 3 (three) differ-

ent datasets from the UCI repository [1]: Nursery[4], Chess-Kr-Vs-Kp (Chess for short) and Splice. These datasets are often used in the literature for empirical evaluation of ensembles. Table 1 describes the characteristics of the datasets used in this study. For each dataset, it shows: number of instances (# Inst.); number of features (# Features), as well as the number of continuous and discrete features; class distribution (Class %); majority error rate; presence or absence of unknown values (Unknown Values) and number (and percentage) of duplicate or conflicting instances (Dup./Conf. Examples).

Table 1. Datasets characteristics summary

# Inst.	12958	3196	3190
# Features (cont.,disc.)	8 (0,8)	36 (0,36)	60 (0,60)
Class (Class %)	not_recom (33.34%) very_recom (2.53%) priority (32.92%) spec_prior (31.21%)	nowin (47.78%) won (52.22%)	EI (24.01%) IE (24.08%) N (51.88%)
Majority Error	66.66% in not_recom	47.78% in won	48.12% in N
Unknown Values	N	N	N
Dup./Conf. Examples	0 (0.00%)	0 (0.00%)	184 (5.77%)

The base-classifiers were induced using the unordered version of $\mathcal{CN}2$ [7] and the decision tree inducer $\mathcal{C}4.5$ [12] symbolic learning algorithms. Therefore, $\mathcal{CN}2$ induces unordered rules and $\mathcal{C}4.5$ induces disjoint rules. The experiments were conducted using the three combination methods described in Section 4: unweighted, weighted by mean and weighted by mean and standard error voting methods, referred respectively by UV, WMV and WMSV. The experiments were carried out using five different set-up scenarios described in Table 2. Different scenarios could be explored using two different learning algorithms. In this work, the same learning algorithm, $\mathcal{CN}2$ or $\mathcal{C}4.5$, were used in two scenarios with different number of base-classifiers, while both algorithms were used in the last scenario. More specifically, the learning algorithm $\mathcal{CN}2$ was used in scenarios *Scn 1* and *Scn 3* while $\mathcal{C}4.5$ was used in scenarios *Scn 2* and *Scn 5*. In scenario *Scn 4*, both algorithms $\mathcal{CN}2$ (in three samples) and $\mathcal{C}4.5$ (in two samples) were used. As sampling is carried out without substitution, we could not use a larger number of partitions.

Table 3 summarizes the error rate and the standard error (in brackets) obtained with datasets Nursery, Chess and Splice, already published in [4]. The first column identifies the experiment and the next five columns show the results

[4] The original Nursery dataset was modified to remove one of the classes that has only 2 instances. As in [4], all references to this dataset in this paper refer to this modified version.

Table 2. Experiments' description

Experiment	# of Partitions	ML algorithms
Scn 1	3	$\mathcal{CN}2$- $\mathcal{CN}2$- $\mathcal{CN}2$
Scn 2	3	$\mathcal{C}4.5$- $\mathcal{C}4.5$- $\mathcal{C}4.5$
Scn 3	5	$\mathcal{CN}2$- $\mathcal{CN}2$- $\mathcal{CN}2$- $\mathcal{CN}2$- $\mathcal{CN}2$
Scn 4	5	$\mathcal{CN}2$- $\mathcal{CN}2$- $\mathcal{CN}2$- $\mathcal{C}4.5$- $\mathcal{C}4.5$
Scn 5	5	$\mathcal{C}4.5$- $\mathcal{C}4.5$- $\mathcal{C}4.5$- $\mathcal{C}4.5$- $\mathcal{C}4.5$

obtained in each scenario. The first five rows, labeled as S_1, S_2, S_3, S_4 and S_5, present the results related to each base-classifier, while the other rows present the results related to the ensembles' construction methods. Results in **bold** indicate that the ensemble is better than each one of the base-classifiers with a 95% confidence level according to a paired t-test. To facilitate the visualization, identical results for all three voting methods are shown only once.

Table 3. Ensembles' mean error rate

Nursery dataset					
Sample	*Scn 1*	*Scn 2*	*Scn 3*	*Scn 4*	*Scn 5*
S_1	5.60 (0.13)	6.16 (0.23)	7.64 (0.24)	7.75 (0.19)	7.92 (0.18)
S_2	5.66 (0.25)	6.47 (0.21)	7.24 (0.21)	7.86 (0.10)	7.77 (0.24)
S_3	5.43 (0.18)	5.97 (0.09)	7.93 (0.26)	7.47 (0.27)	7.80 (0.17)
S_4	-	-	7.83 (0.22)	7.50 (0.17)	7.73 (0.24)
S_5	-	-	7.82 (0.16)	7.94 (0.29)	7.46 (0.24)
UV	**3.86 (0.13)**		**4.51 (0.16)**	**4.42 (0.15)**	
WMV	**4.13 (0.13)**	**4.81 (0.18)**	**5.06 (0.18)**	**4.81 (0.11)**	**6.41 (0.14)**
WMSV	**4.18 (0.14)**		**4.95 (0.19)**	**4.88 (0.13)**	
Chess dataset					
Sample	*Scn 1*	*Scn 2*	*Scn 3*	*Scn 4*	*Scn 5*
S_1	2.72 (0.35)	1.69 (0.29)	3.82 (0.73)	3.63 (0.72)	3.04 (0.35)
S_2	2.47 (0.32)	1.25 (0.19)	3.57 (0.44)	2.75 (0.36)	3.00 (0.46)
S_3	2.72 (0.49)	1.75 (0.28)	2.88 (0.27)	3.00 (0.35)	2.53 (0.35)
S_4	-	-	2.94 (0.47)	2.22 (0.28)	3.13 (0.51)
S_5	-	-	3.04 (0.44)	2.50 (0.27)	3.04 (0.48)
UV,WMV,WMSV	2.50 (0.33)	0.91 (0.16)	2.25 (0.33)	**1.60 (0.25)**	2.32 (0.35)
Splice dataset					
Sample	*Scn 1*	*Scn 2*	*Scn 3*	*Scn 4*	*Scn 5*
S_1	18.50 (1.85)	9.03 (0.38)	15.33 (0.97)	14.61 (0.73)	11.25 (0.70)
S_2	15.52 (1.04)	9.37 (0.47)	14.80 (0.92)	16.02 (1.20)	13.01 (0.67)
S_3	15.92 (1.28)	9.00 (0.48)	15.30 (0.64)	19.75 (1.73)	11.41 (0.40)
S_4	-	-	15.45 (1.45)	11.72 (0.74)	12.13 (0.55)
S_5	-	-	16.77 (0.93)	11.32 (0.52)	13.26 (0.71)
UV	**11.54 (0.49)**	**7.55 (0.39)**	**9.72 (0.45)**	**7.30 (0.70)**	**8.68 (0.42)**
WMV	**11.13 (0.39)**	**7.34 (0.27)**	**9.84 (0.41)**	**7.08 (0.66)**	**8.53 (0.53)**
WMSV	**11.19 (0.37)**	**7.59 (0.38)**	**9.59 (0.37)**	**7.15 (0.62)**	**8.53 (0.52)**

As can be observed in Table 3, using Nursery and Splice datasets, better precision results were obtained using the ensemble methods in all scenarios, while using chess dataset better results were obtained only in one scenario — *Scn 4* — with 95% confidence level. Further analises about these results can be found in [4].

To evaluate the performance of the explanation simplification algorithm proposed in this work, we measured the reduction rate in the number of rules between the initial set $\mathbf{R}_{expl}$ and the final explanation set $\mathbf{R}^*_{expl}$. Given an instance $\mathbf{x}$[5], the reduction rate $RR(\mathbf{x})$ on the number of rules between the initial explanation set $\mathbf{R}_{expl}(\mathbf{x})$ and the final simplified explanation set $\mathbf{R}^*_{expl}(\mathbf{x})$, can be defined by Equation 1.

$$RR(\mathbf{x}) = \frac{|\mathbf{R}_{expl}(\mathbf{x})| - |\mathbf{R}^*_{expl}(\mathbf{x})|}{|\mathbf{R}_{expl}(\mathbf{x})|}. \quad (1)$$

To obtain the mean and the standard error of the reduction rate of each constructed ensemble, we averaged the reduction rate RR of all instances in the datasets Nursery, Chess and Splice. Table 4 shows, for each dataset, the reduction rate obtained by applying the simplification process. For each dataset, it shows the mean number of rules in $\mathbf{R}_{expl}$, $\mathbf{R}^*_{expl}$, the reduction rate RR and the corresponding standard errors for the respective datasets. To facilitate visualization, identical results for all three voting methods are shown only once.

Table 4. Reduction rate of the explanation set

Nursery Dataset

	Comb. Meth.	*Scn 1*	*Scn 2*	*Scn 3*	*Scn 4*	*Scn 5*
$\lvert\mathbf{R}_{expl}\rvert$	UV	4.04 (2.20)	2.92 (0.26)	6.52 (3.22)	5.77 (2.06)	4.78 (0.55)
	WMV			6.51 (3.24)	5.77 (2.08)	
	WMSV			6.51 (3.25)	5.77 (2.07)	
$\lvert\mathbf{R}^*_{expl}\rvert$	UV	1.70 (1.11)	1.17 (0.38)	2.01 (1.41)	1.80 (1.16)	1.16 (0.40)
	WMV	1.69 (1.12)		2.00 (1.42)		
	WMSV	1.69 (1.12)		2.00 (1.42)		
$RR\%$	UV	58.19 (16.19)	59.43 (14.22)	68.88 (17.46)	68.88 (16.25)	75.02 (10.61)
	WMV	58.18 (16.22)		68.76 (17.75)	68.79 (16.49)	
	WMSV	58.18 (16.21)		68.77 (17.39)	68.92 (16.15)	

Chess Dataset

	Comb. Meth.	*Scn 1*	*Scn 2*	*Scn 3*	*Scn 4*	*Scn 5*
$\lvert\mathbf{R}_{expl}\rvert$	UV, WMV e WMSV	4.22 (2.19)	2.99 (0.09)	7.18 (3.50)	6.14 (1.97)	4.92 (0.36)
$\lvert\mathbf{R}^*_{expl}\rvert$	UV, WMV e WMSV	1.91 (1.16)	1.05 (0.22)	2.48 (1.68)	2.83 (1.51)	1.06 (0.29)
$RR\%$	UV, WMV e WMSV	52.46 (19.93)	64.89 (7.43)	63.52 (20.95)	53.73 (21.23)	77.98 (9.83)

Splice Dataset

	Comb. Meth.	*Scn 1*	*Scn 2*	*Scn 3*	*Scn 4*	*Scn 5*
$\lvert\mathbf{R}_{expl}\rvert$	UV	4.69 (2.99)	2.89 (0.31)	7.07 (3.54)	6.09 (2.54)	4.69 (0.64)
	WMV			7.07 (3.54)	6.10 (2.51)	
	WMSV			7.06 (3.55)	6.10 (2.51)	
$\lvert\mathbf{R}^*_{expl}\rvert$	UV	3.84 (2.34)	1.52 (0.61)	5.53 (2.77)	4.63 (2.03)	1.83 (0.88)
	WMV			5.54 (2.76)	4.64 (2.01)	
	WMSV			5.53 (2.77)	4.64 (2.01)	
$RR\%$	UV	14.07 (14.57)	46.79 (21.89)	17.93 (18.94)	22.65 (15.40)	59.39 (22.21)
	WMV			17.91 (18.94)	22.67 (15.39)	
	WMSV			17.93 (18.94)	22.67 (15.39)	

Analyzing the results, it can be observed that when $\mathcal{CN}2$ is used as base-classifier (*Scn 1* and *Scn 3*), the average number of rules in $|\mathbf{R}_{expl}|$ is larger than in the other scenarios. A similar phenomena occurs in scenario *Scn 4*, in which

[5] Recall that the classification of a new instance $\mathbf{x}$ is given by $\mathbf{h}^*(\mathbf{x})$.

$\mathcal{CN}2$ was used to induce 3 (three) out of 5 (five) base-classifiers. Furthermore, *Scn 3* has the highest number of rules $|\mathbf{R}_{expl}|$ before simplification. A possible reason is that, unlike $\mathcal{C}4.5$, which induces disjoint rule sets, $\mathcal{CN}2$ may induce overlapping rules, thus incrementing the number of rules that covers an instance. Another aspect to be observed is that, for each dataset, the average number of rules $|\mathbf{R}_{expl}|$ in each of the five scenarios is almost the same for the three voting methods; the same occurs with the average number of rules after simplification $|\mathbf{R}^*_{expl}|$.

For all datasets, the maximum reduction rate was achieved in *Scn 5*: 75.02% for Nursery dataset; 77.98% for Chess dataset; and 59.42% for Splice dataset. Considering the high reduction rate in the final set $\mathbf{R}^*_{expl}$, we can conclude that most rules in $\mathbf{R}_{expl}$ are specializations of other rules in $\mathbf{R}_{expl}$. On the other hand, the minimum reduction rate occurs in scenario *Scn 1*: 58.18% for Nursery dataset; 52.46% for Chess dataset; and 14.07% for Splice dataset.

Table 5 illustrates an example of application of the algorithm described in Section 5 using Nursery dataset. Features in Nursery dataset are parents, has_nurs, form, children, housing, finance, social, health; and the class feature is nursery. The instance considered from the dataset is $\mathbf{x}$ = (usual, proper, complete, 1, less_conv, convenient, nonprob, priority), which is (correctly) classified as priority by an ensemble $\mathbf{h}^*$ constructed using Scenario *Scn 1* and UV combination method — $\mathbf{h}^*(\mathbf{x})$ = priority. In this table, the bodies of the rules that cover $\mathbf{x}$ from $\mathbf{h}^*$ are shown, as well as the final simplified explanation given by the algorithm — bodies of the remaining rules. The first line in Table 5 shows the bodies of the two rules from base-classifier $\mathbf{h}_1$, the second line shows the the bodies of the 3 (three) rules from base-classifier $\mathbf{h}_2$, and the third line shows the the bodies of the two rules from base-classifier $\mathbf{h}_3$, which correctly cover the instance $\mathbf{x}$. The last line in Table 5 shows the final explanation, composed by the bodies of the more general rules from the original set of rules $\mathbf{R}_{expl}$. Observe that the original set of rules $\mathbf{R}_{expl}$ contains seven rules, while the simplified explanation set $\mathbf{R}^*_{expl}$ only contains two rules: rules 1a and 2a, as rules 1b and 1c are equal to 1a, and 2b, 3b and 2c are specializations of 2a.

Table 5. Rules from hypotheses $\mathbf{h}_1$, $\mathbf{h}_2$ and $\mathbf{h}_3$ of ensemble $\mathbf{h}^*$ that cover instance $\mathbf{x}$, and the final simplified explanation $\mathbf{R}^*_{expl}$.

$\mathbf{h}\_1$	1a – parents = usual AND has_nurs = proper AND health = priority 2a – has_nurs = proper AND children = 1 AND health = priority
$\mathbf{h}\_2$	1b – parents = usual AND has_nurs = proper AND health = priority (*equals to* 1a) 2b – has_nurs = proper AND children = 1 AND housing = less_conv AND health = priority (*a specialization of* 2a) 3b – has_nurs = proper AND form = complete AND children = 1 AND health = priority (*a specialization of* 2a)
$\mathbf{h}\_3$	1c – parents = usual AND has_nurs = proper AND health = priority (*equals to* 1a) 2c – has_nurs = proper AND form = complete AND children = 1 AND health = priority (*a specialization of* 2a)
$\mathbf{R}^*_{expl}$	1a – parents = usual AND has_nurs = proper AND health = priority 2a – has_nurs = proper AND children = 1 AND health = priority

7 Conclusions and Future Work

In this work, we proposed a method to simplify classification explanations given by symbolic ensembles, which are constructed using combination methods of symbolic base-classifiers as proposed in [4]. Based on the symbolic ensembles constructed using these combination methods, we conducted some experiments to analyze the viability of our proposal to simplify the ensemble's explanation. Results show that a substantial reduction of the number of rules needed to explain the ensembles decisions was achieved, showing the feasibility of our proposal. Future work includes using more datasets to test the explanation method, as well as analyzing the syntactic complexity of the resulting rules in explanation set.

Acknowledgments: This research was supported by the Brazilian Research Councils CNPq and FAPESP. We also would like to thank the anonymous referees for their important comments.

References

1. A. Asuncion and D.J. Newman. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`. Irvine, CA: University of California, School of Information and Computer Science.
2. E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36(1/2):105–139, 1999.
3. F. C. Bernardini and M. C. Monard. ELE — Ensemble Learning Environment to construct symbolic classifiers: Implementation description (in portuguese). Technical Report 243, ICMC/USP, 2004.
4. F. C. Bernardini, M. C. Monard, and R. C. Prati. Constructing ensembles of symbolic classifiers. *Int. J. Hybrid Intelligent Systems*, 3(3):159–167, 2006.
5. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
6. L. Breiman. Arcing classifiers. *The Annals Of Statistics*, 26(3):801–849, 1998.
7. P. Clark and R. Boswell. Rule induction with $\mathcal{CN}2$: Some recent improvements. In Y. Kodratoff, editor, *Proc. of the 5th European Working Session on Learning (EWSL 91)*, pages 151–163, 1991.
8. T. G. Dietterich. Machine learning research: Four current directions. *AI Magazine*, 18:97–136, 1997. `http://www.cs.orst.edu/~tgd/`.
9. T. G. Dietterich. Ensemble methods in machine learning. In *First International Workshop on Multiple Classifier Systems. LNCS*, volume 1857, pages 1–15, New York, 2000.
10. Y. Freund and R.E.. Schapire. A decision-theoretic generalization of on-line learninng and an application to boosting. *J. of Computer and System Sciences*, 55(1):119–139, 1997.
11. K. T. Leung and D. S. Parker. Empirical comparisons of various voting methods in bagging. In *KDD '03: Proc. 9th ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining*, pages 595–600. ACM Press, 2003.
12. J. R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers, 1988.
13. L. Todorovski and S. Dzeroski. Combining classifiers with meta decision trees. *Machine Learning*, 50(3):223–249, 2003.