Expert Systems with Applications 39 (2012) 6827–6837

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

# Bagging schemes on the presence of class noise in classification

Joaquín Abellán [*], Andrés R. Masegosa

*Department of Computer Science and Artificial Intelligence, University of Granada, Spain*

ARTICLE INFO

*Keywords:*
Imprecise probabilities
Imprecise Dirichlet model
Information based uncertainty measures
Ensemble decision trees
Classification noise

ABSTRACT

In this paper, we study one application of Bagging credal decision tree, i.e. decision trees built using imprecise probabilities and uncertainty measures, on data sets with class noise (data sets with wrong assignations of the class label). For this aim, previously we also extend a original method that build credal decision trees to one which works with continuous features and missing data. Through an experimental study, we prove that Bagging credal decision trees outperforms more complex Bagging approaches on data sets with class noise. Finally, using a bias–variance error decomposition analysis, we also justify the performance of the method of Bagging credal decision trees, showing that it achieves a stronger reduction of the variance error component.

## 1. Introduction

In the general area of data mining, the supervised classification problem is normally defined as follows: we have a data set of observations, called the *training set*, and we wish to obtain a set of rules in order to assign a value of the variable to be predicted (discrete or discretized) to each new observation. To verify the quality of this set of rules, a different set of observations is used, this set is called the *test set*. The variable to be predicted (classified) is called *class variable* and the rest of variables in the data set are called *predictive attributes* or *features*. There exist important applications of classification in fields such as medicine, bioinformatics, physics, pattern recognition, economics, etc., and are used for disease diagnosis, meteorological forecasts, insurance, text classification, to name but a few.

Within a probabilistic approach, supervised classification problem is faced as an inference problem. The probability distribution of the class variable given the predictive attributes is estimated from a training data set and the quality of this estimation is then evaluated in an independent test data set. In order to estimate or learn this probability distribution, many different approaches can be employed.

The combination or ensemble of classifiers can improve the accuracy of these in supervised classification, specially for the high classification accuracy performance they offer as well as the robustness to different issues which appear in real applications such as class imbalanced problems or training data sets with a very low size. Among the different approaches to combine classification models, ensembles of decision trees are the most accepted and studied, although this approach is not just restricted to learning decision trees, it has been also applied to most other machine learning methods.

Decision trees (or classification trees) are a special family of classifiers with a simple structure and very easy to interpret. But the important aspect of decision trees which make them very suitable to be employed in ensembles of classifiers is their inherent instability. This property causes that different training datasubsets from a given problem domain will produce very different trees. This characteristic was essential to consider them as suitable classifiers in ensemble schemes such as Bagging (Breiman, 1996), Boosting (Freund & Schapire, 1996) or Random Forest (Breiman, 2001).

Classification noise is named to those situations which appear when data sets have incorrect class labels in their training and/or test data sets. There are many relevant situations in which this problem can arise due to deficiencies in the data learning and/or test capture process (wrong disease diagnosis method, human errors in the class label assignation, etc.) The performance of classifiers on data sets with classification noise is a very important issue for machine learning methods.

Many studies have been concerned with the problems related to the performance of ensembles of decision trees in noisy data domains (Dietterich, 2000; Freund & Schapire, 1996; Melville et al., 2004). These studies showed as Boosting strongly deteriorates its performance while Bagging ensembles are the most robust and outperforming ensembles in these situations. Noisy training data usually increases the variance in the predictions of the classifiers, therefore, Bagging ensembles based on variance-reducing methods work very well (Breiman, 1996).

In this study, we show as the employment of Bagging ensembles of a special type of decision trees, called credal sets, which

[*] Corresponding author.
*E-mail addresses:* jabellan@decsai.ugr.es (J. Abellán), andrew@decsai.ugr.es (A.R. Masegosa).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.01.013

are based on imprecise probabilities (via the Imprecise Dirichlet model Walley, 1996) and information based uncertainty measures (via the maximum of entropy function, Klir, 2006), can be a successful tool in classification problems with a high level of noise in the class variable.

This paper is organized as follows: in Section 2, we present previous knowledge necessary about decision trees and methods to ensemble decision trees; in Section 3, we focus on our method of Bagging credal decision trees; in Section 4, we present the results of experiments conducted to compare the performance of our method for Bagging credal decision trees with other Bagging scheme which uses the popular C4.5 method, on data sets with different levels of classification noise; and finally, Section 6 is devoted to the conclusions.

## 2. Decision trees

Decision trees (also known as classification trees or hierarchical classifiers) started to play an important role in machine learning since the publication of Quinlan's ID3 (Quinlan, 1986). Subsequently, Quinlan also presented the algorithm C4.5 (Quinlan, 1993), which is an advanced version of ID3. Since then, C4.5 has been considered as a standard model in supervised classification. They have also been widely applied as a data analysis tool to very different fields, such as astronomy, biology, medicine, etc.

Decision trees are models based on a recursive partition method, the aim of which is to divide the data set using a single variable at each level. This variable is selected with a given criterion. They ideally come to define set of cases in which all the cases belong to the same class.

Their knowledge representation has a simple tree structure. It can be interpreted as a compact rule set in which each node of the tree is labelled with an attribute variable that produces a ramification for each one of its values. The leaf nodes are labelled with a class label, as can be seen in Fig. 1.

The process for inferring a decision tree is mainly determined by the followings points:

- The criteria used for selecting the attribute to be placed in a node and ramified.
- The criteria for stopping the ramification of the tree.
- The method for assigning a class label or a probability distribution at the leaf nodes.
- The post-pruning process used to simplify the tree structure.

Many different approaches for inferring decision trees, which depend upon the aforementioned factors, have been published. Quinlan's ID3 (Quinlan, 1986) and C4.5 (Quinlan, 1993) along with the CART approach of (Breiman, Friedman, Olshen, & Stone, 1984) stand out among all of these.

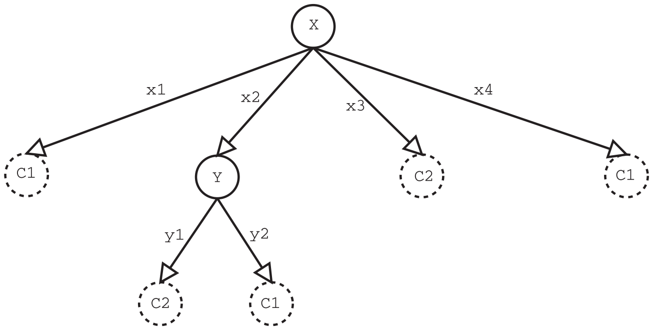


**Fig. 1.** Decision tree.

### 2.1. C4.5 tree inducer

In this subsection we will give a brief explanation of the most important aspects of the this very-well known tree inducer. There are at least eight different releases of its concrete implementation, aiming to improve the efficiency, the handling of numeric attributes, missing values... and many of them provided different options and alternatives. We just highlight the main ideas behind all these approaches that were introduced in Quinlan (1993):

**Split criterion**: Information Gain (Quinlan, 1986) was firstly employed to select the split attribute at each branching node. But this measure was strongly affected but the number of states of the split attribute: attributes with a higher number of states were usually preferred. Quinlan's introduced for this new tree inducer the *Information Gain Ratio* (IGR) criterion which penalizes variables with many states. This score normalizes the Information Gain of an attribute $X$ by its own entropy.

**Handling numeric attributes**: This tree inducer handles numeric attributes employing a very simple approach. Within this method, only binary split attributes are considered and each possible split point is evaluated and it is finally selected the one which induced a partition of the samples with the highest split score (i.e. the *Information Gain Ratio*).

**Dealing with missing values**: It is assumed that missing values are randomly distributed (*Missing at Random Hypothesis*). In order to compute the scores, the instances are split into pieces. The initial weight of an instance is equal to the unit, but when it goes going down a branch receives a weight equal to the proportion of instances that belongs to this branch (weights sum to 1). Information gain based scores can work with this fractional instances using sum of weights instead of sum of counts. When making predictions, C4.5 marginalize the missing variable merging the predictions of all the possible branches that are consistent with the instance (there are several branches because it has a missing value) using their previously computed weights.

**Post-pruning process**: Although there are many different proposals to carry out a post-pruning process of a decision tree, the technique employed by C4.5 is called *Pessimistic Error Pruning*. This method computes an upper bound of the estimated error rate of a given subtree employing a continuity correction of the Binomial distribution. When the upper bound of a subtree hanging from a given node is greater than the upper bound of the errors produced by the estimations of this node supposing it acts as a leaf, then this subtree is pruned.

### 2.2. Ensembles of decision trees

In many fields, such as medicine or finances, a second opinion, sometimes even more, is often sought before a decision is taken. Thus, we finally weight the individual opinions and combine them to take a final decision that should initially be more robust and trusted. This process of consulting "several experts" before making a decision has been also employed by the computational intelligence community. This approach is known by different names such as multiple classifier systems, committee of classifiers, mixture of experts or ensemble of classifiers, and has shown to produce much better results in comparison to single classifiers.

In this field, ensembles of decision trees appear to present the best trade-off among performance, simplicity and theoretic bases. The basic idea consists of generating a set of different decision trees and combining them with a majority vote criteria. That is to say, when an unlabelled unclassified instance arises, each single decision tree makes a prediction and the instance is assigned to the class value with the highest number of votes. In this way, a diversity issue appears as a critical point when an ensemble is built (Breiman, 1996). If all decision trees are quite similar, the ensemble performance will not be much better than a single decision tree. However, if the ensemble is made up of a broad set of different decisions and exhibits good individual performance, the ensemble will become more robust, with a better prediction capacity.

There are many different approaches to this problem but Bagging (Breiman, 1996), Random Forests (Breiman, 2001) and AdaBoost (Freund & Schapire, 1996) stand out as the best known and most competitive.

### Bagging

Breiman's Bagging (bootstrap aggregating) (Breiman, 1996) is one of the first cases of an ensemble of decision trees. It is also the most intuitive and simple and performs very well. Diversity in Bagging is obtained by using bootstrapped replicas of the original training set: different training datasets are randomly drawn with replacement. And, subsequently, a single decision tree is built with each training data replica with the use of the standard approach (Breiman et al., 1984). Thus, each tree can be defined by a different set of variables, nodes and leaves. Finally, their predictions are combined by a majority vote. In Algorithm 1 a pseudocode description of this method is depicted.

**Algorithm 1.** Bagging Algorithm

```
Input:
    -Training data D with correct labels $c_1, \ldots, c_k$ representing
  the $k_c$ classes.
    -A Decision Tree learning algorithm LearnDecisionTree,
    -Integer M specifying number of iterations.
Do $m = 1, \ldots, M$
    1. Take a bootstrapped replica $D_m$ by randomly drawing
  from D.
    2. Call LearnDecisionTree with $D_m$ and receive a single
  decision tree $DT_m$.
    3. Add $DT_m$ to the ensemble, E.
End
Test: Simple Majority Voting Given an unlabeled instance x
    1. Evaluate the ensemble $E = DT_1, \ldots, DT_M$ on x.
    2. Let $v_{m,i} = 1$ if $DT_m$ predicts class $c_i$, $v_{m,i} = 0$ otherwise ($v_{m,i}$
  be the vote given to class $c_i$ by classifier $DT_m$).
    3. Obtain total vote received by each class

  $V_i = \sum_{m=1}^{M} v_{m,i}, i = 1, \ldots, k_c$

    4. Choose the class that receives the highest total vote as the
  final classification.
```

Bagging approach can be employed with different decision tree inducers, although there is none considered as an standard. There are studies were Bagging is employed with ID3, CART or C4.5 tree inducers and sometimes these inducers employed some post-pruning method. The very cited work of Diettrich (Dietterich, 2000) employs the C4.5 tree inducer (with and without post-pruning) to point out Bagging as an outperforming ensemble approach in datasets with classification noise, but there was any definitive suggestion about the employment of pruning.

## 3. Bagging credal decision trees

A new method which uses a split criterion based on uncertainty measures and imprecise probabilities (Imprecise Info-Gain criterion) to build simple decision trees was firstly presented in Abellán & Moral's method (Abellán & Moral, 2003) and in a more complex procedure in Abellán and Moral (2005). In a similar way to ID3, this decision tree is only defined for discrete variables, it does not works with missing values, and it does not carry out a posterior pruning process.

In a recent work (Abellán & Masegosa, 2009), these decision trees were introduced in a Bagging scheme and compared against similar ensembles which were built with several classic information split criteria based on frequentist approaches: Info-Gain (Quinlan, 1986), Info-Gain-Ratio (Quinlan, 1993) and Gini Index (Breiman et al., 1984); and a similar preprocessing step. The conclusions depicted in this study pointed out as the Bagging ensembles of single decision trees built with the Imprecise Info-Gain criteria outperformed these other classic split criteria on data sets with classification noise. These promising results encouraged the extension of the ensembles we presented in that paper to a broader class of data sets with continuous attributes and missing values as well as the introduction of a post-pruning method. Moreover, we aim to compare this new method with some of the state-of-the-art ensembles of classification trees in data sets with classification noise.

In the following subsections, we detail our method to build credal decision trees and its extension to be able to handle numeric attributes and deal with missing values. Moreover, we highlight the main differences respect to the current implementation of C4.5 release 8 with the aim of pointing out as our approach does not employ many of the free parameters introduced in this famous tree inducer.

### 3.1. Imprecise Information Gain

This is the split criterion employed to build credal decision trees. It is based on the application of uncertainty measures on convex sets of probability distributions. More specifically, probability intervals are extracted from the data set for each case of the class variable using Walley's Imprecise Dirichlet model (IDM) (Walley, 1996), which represents a specific kind of convex sets of probability distributions (see Abellán, 2006), and on these the entropy maximum is estimated. This is a total uncertainty measure which is well known for this type of set (see Abellán, Klir, & Moral, 2006 and Abellán & Masegosa, 2008).

The IDM depends on a hyperparameter $s$ and it estimates that (in a given data set) the probabilities for each value of the class variable are within the interval:

$$p(c_j) \in \left[\frac{n_{c_j}}{N+s}, \frac{n_{c_j}+s}{N+s}\right],$$

with $n_{c_j}$ as the frequency of the set of values ($C = c_j$), of the class variable, in the data set. The value of parameter $s$ determines the speed with which the upper and lower probability values converge when the sample size increases. Higher values of $s$ give a more cautious inference. Walley (1996) does not give a definitive recommendation for the value of this parameter but he suggests values between $s = 1$ and $s = 2$. In Bernard (2005), we can find reasons in favor of values greater than 1 for $s$.

Let $K(C)$ and $K(C|(X_i = x_t^i))$ be the following closed and convex sets of probability distributions, also called credal sets, $q$ on $\Omega_C$ (the set of possible values or states of $C$):

$$K(C) = \left\{q|\ q(c_j) \in \left[\frac{n_{c_j}}{N+s}, \frac{n_{c_j}+s}{N+s}\right]\right\},$$

$$K(C|(X_i = x_t^i)) = \left\{q|q(c_j) \in \left[\frac{n_{\{c_j,x_t^i\}}}{N+s}, \frac{n_{\{c_j,x_t^i\}}+s}{N+s}\right]\right\},$$

with $n_{\{c_j,x_t^i\}}$ as the frequency of the set of values $\{C = c_j, X_i = x_t^i\}$ in the data set, with $X_i$ a predictive attribute. We can define the Imprecise Info-Gain for each variable $X_i$ as:

$$\mathbf{IIG}(C|X_i) = S(K(C)) - \sum_t p(x_t^i) S(K(C|(X_i = x_t^i))),$$

where $S()$ is the maximum entropy function of a credal set (see Abellán, 2011; Abellán et al., 2006).

For the previously defined intervals and for a value of $s$ between 1 and 2, it is very easy to obtain the maximum entropy using procedures of Abellán and Moral (2003) and Abellán and Moral (2006) or the specific one for the IDM of Abellán (2006), which obtains its lower computational cost for $s = 1$.

We must remark that this new information measure can give us negative values (it is the base of the ensemble method exposed in Abellán & Masegosa (2010)). This is not a characteristic of classics scores, as the Information Gain or the Information Gain Ratio. If the data set is split, there is always a reduction of the entropy of the class variable with this scores but no with our new score.

### 3.2. Building credal decision trees

Each node $\aleph$, in a credal decision tree, produces a partition $\mathscr{D}$ of the data set (for the root node it is considered all the data set). Also, each node $\aleph$ has associated a list $\Gamma$ of labels of features (features that are not in the path from the root node to $\aleph$). The original procedure of Abellán and Moral (2003) for building credal trees, which was presented to work on discrete variables and without treatment of missing values, can be simplified in the algorithm of Fig. 2.

Each `Exit` state of the above procedure corresponds to a leaf node. Here, the most probable value of the class variable, associated with the corresponding partition, is selected (to more details, see Abellán & Moral, 2003).

### 3.3. Decision tree inducer

In this subsection we present the extension of the decision tree inducer presented in Abellán and Moral (2003). This extension aims to handle numeric attributes and deal with missing values. It is based on a direct adaptation of the C4.5 standard approaches to these situations. The main differences arise with strong simplification in the procedures with respect to the last implementation of C4.5 release 8.

**Split criterion**: The attribute with the maximum Imprecise Info-Gain is selected as split attribute at each branching node. *This simple criteria contrast with the sophisticated conditions of C4.5 release 8: it is selected the attribute with the highest Info-Gain Ratio score and whose Info-Gain score is higher than the average Info-Gain scores of the valid split attributes. These valid split attributes are those which are either numeric or whose number of values is smaller than the thirty percent of the number of instances which are in this branch.*

**Stop criterion**: The branching of the decision tree is stopped when there is no split attribute with a positive IIG score or there are 2 or less instances in a leaf.
*In C4.5 release 8 is also established a minimum number of instances per leaf which is usually set to 2. But in addition to this, using the aforementioned condition in "Split Criteria" of valid split attributes, the branching of a decision tree is also stopped when there is not any valid split attribute.*

**Handling numeric attributes**: Numeric attributes are handled using the same approach detailed in Section 2.1. Each possible split point is evaluated and the one which induces a binary partition with the highest Imprecise Info-Gain is selected.
*C4.5 release 8 employs this approach but it establishes an additional requirement related to the minimum number of instances required for each partition. This minimum is set using the following heuristic: the ten per cent of the ratio between the number of instances which fall in this branch node and the number of values of the class variable.*
*It also corrects the Information Gain of the optimal partition subtracting to this final value the logarithm of the number of evaluated split points divided by the total number of instances in this branching node.*

**Dealing with missing values**: We employ the same approach previously detailed in Section 2.1. The Imprecise Info-Gain can be also computed to those cases where there are fractional counts in the class variable.

**Post-pruning process**: Because the aim of this work is to propose a simple approach that exploits the robustness and performance of the Imprecise Info-Gain criterion, we introduce one the most simple pruning techniques: Reduced Error Pruning (Quinlan, 1987). It is a bottom-up method that compares the error in a subtraining data set of one node with the error of its hanging subtree. If the error of the parent node is lower, then the subtree is pruned. In this implementation, the training data set was divided in 3 folds, two of them were employed to build the tree and the other one to estimate the errors.

## 4. Experimental analysis

The experimental analysis is carried out in two main blocks. In Section 4.1, we analyze how Bagging ensembles of credal decision

```
Procedure BuildTree($\aleph$,$\Gamma$)

1. If $\Gamma = \emptyset$, then Exit.
2. Let $\mathcal{D}$ be the partition associated with node $\aleph$
3. Compute the value
        $\delta = \max_{X_j \in \Gamma} \{IIG^{\mathcal{D}}(C|X_j)\}$
4. If $\delta$ is lower than or equal to 0 then
   5. Exit
6. Else
   7. Let $X_l$ be the variable for which the maximum $\delta$ is attained
8. Remove $X_l$ from $\Gamma$
9. Assign $X_l$ to node $\aleph$
10. For each possible value $x_l$ of $X_l$
   11. Add a node $\aleph_l$
   12. Make $\aleph_l$ a child of $\aleph$
   13. Call BuilTree($\aleph_l$,$\Gamma$)
```

**Fig. 2.** Procedure to build a credal decision tree.

trees are affected by the introduction of classification noise in terms of classification accuracy and, also, we analyze how the size of the single trees varies with the different rates of noise. In Section 4.2, we employ a bias–variance error decomposition analysis to see why Bagging credal decision trees outperforms the standard approach in data sets with classification noise.

### *4.1. Performance evaluation in terms of classification accuracy experimental set-up*

In our experimentation, we have used a wide and different set of 25 known data sets, obtained from the *UCI repository of machine learning databases* which can be directly downloaded from ftp://ftp.ics.uci.edu/machine-learning-databases. A brief description of these can be found in Table 1, where column "N" is the number of instances in the data sets, column "Attrib" is the number of predictive attributes, "Num" is the number of numerical variables, column "Nom" is the number of nominal variables, column "k" is the number of cases or states of the class variable (always a nominal variable) and column "Range" is the range of states of the nominal variables of each data set.

In the literature, ensembles of Bagging decision trees have been implemented using many different tree inducers such as CART (Breiman et al., 1984), ID3 (Quinlan, 1986) or C4.5 (Quinlan, 1993). We take as reference the work of Diettrich (Dietterich, 2000) where Bagging was evaluated with the last release of C4.5 (cited as Bagging-C4.5R8 or B-C4.5). More precisely, we took the implementation of this tree inducer included in the machine learning platform *Weka* (Witten & Frank, 2005).

Our Bagging ensembles of credal decision trees (cited as Bagging-CDT or B-CDT) was implemented using data structures of *Weka* and *Elvira* software (Elvira, 2002) (other software platform for the evaluation of probabilistic graphical models). The parameter of the IDM was set to $s = 1$ (see Section 3.1).

Both Bagging ensembles were built with 100 decision trees. Although the number of trees can strongly affect the performance of the ensembles, this is a reasonable number of trees for the low-medium size of the data sets employed in this evaluation (see Table 1) and it has been used in other related works as in Freund and Schapire (1996).

**Table 1**
Data set description.

| Data sets | N | Attrib | Num | Nom | k | Range |
|---|---|---|---|---|---|---|
| Anneal | 898 | 38 | 6 | 32 | 6 | 2–10 |
| Audiology | 226 | 69 | 0 | 69 | 24 | 2–6 |
| Autos | 205 | 25 | 15 | 10 | 7 | 2–22 |
| Breast-cancer | 286 | 9 | 0 | 9 | 2 | 2–13 |
| CMC | 1473 | 9 | 2 | 7 | 3 | 2–4 |
| Colic | 368 | 22 | 7 | 15 | 2 | 2–6 |
| Credit-German | 1000 | 20 | 7 | 13 | 2 | 2–11 |
| Diabetes-Pima | 768 | 8 | 8 | 0 | 2 | – |
| Glass-2 | 163 | 9 | 9 | 0 | 2 | – |
| Hepatitis | 155 | 19 | 4 | 15 | 2 | 2 |
| Hypothyroid | 3772 | 29 | 7 | 22 | 4 | 2–4 |
| Ionosfere | 351 | 35 | 35 | 0 | 2 | – |
| kr-vs-kp | 3196 | 36 | 0 | 36 | 2 | 2–3 |
| Labor | 57 | 16 | 8 | 8 | 2 | 2–3 |
| Lymph | 146 | 18 | 3 | 15 | 4 | 2–8 |
| Mushroom | 8123 | 22 | 0 | 22 | 2 | 2–12 |
| Segment | 2310 | 19 | 19 | 0 | 7 | – |
| Sick | 3772 | 29 | 7 | 22 | 2 | 2 |
| Solar-flare1 | 323 | 12 | 0 | 12 | 2 | 2–6 |
| Sonar | 208 | 60 | 60 | 0 | 2 | – |
| Soybean | 683 | 35 | 0 | 35 | 19 | 2–7 |
| Sponge | 76 | 44 | 0 | 44 | 3 | 2–9 |
| Vote | 435 | 16 | 0 | 16 | 2 | 2 |
| Vowel | 990 | 11 | 10 | 1 | 11 | 2 |
| Zoo | 101 | 16 | 1 | 15 | 7 | 2 |

Using *Weka's* filters, we added the following percentages of random noise to the class variable: 0%, 5%, 10%, 20% and 30%, only in the training data set. The procedure to introduce noise was the following: a given percentage of instances of the training data set was randomly selected and, then, their current class values were randomly changed to other possible values. The instances belonging to the test data set were left unmodified. To estimate the classification accuracy of each classifier ensemble in each data set, we repeated 10 times a k-10 folds cross validation procedure and the average values were reported.

To compare both ensembles, we have used different statistical tests with the aim of having robust comparisons which, in other case, might be biased if only one statistical test is employed, because all of them are based on different assumptions (see Demsar, 2006; Witten & Frank, 2005 for a complete explanation and further references to these statistical tests).

**Corrected Paired T-test**: a corrected version of the Paired T-test implemented in *Weka*. It is used to avoid some problems of the original test with cross validation schemes. This test checks whether one classifier is better or worse than another on average, across all training and test datasubsets obtained from a given data set. We use this test on the training and test datasubsets obtained from a 10 times k-10 folds cross validation procedure on a original data set. The levels of significance used for this test is 0.05.

**Sign test (counts of wins, losses and ties)**: a binomial test that counts the number $w$ of data sets on which an algorithm is the overall winner.

**Wilcoxon signed-ranks test**: a non-parametric test which ranks the differences in performance of two classifiers of each data set, ignoring the sings, and compares the ranks for the positive and the negative differences.

**Friedman test**: a non-parametric test which ranks the algorithms for each data set separately, the best performing algorithm getting the rank of 1, the second best rank 2,... The null hypothesis is that all the algorithms are equivalent. When the null-hypothesis is rejected, we can compare all the algorithms to each other using the average raking employing the **Nemenyi test**.

Except for the *Corrected Paired T-test*, we detail in bold face the name of the ensemble that has a statistically significant better classification accuracy (when the p-value is lower or equal than 0.05) and we write "=" when there is no statistical significant differences between both ensembles (p-values higher than 0.05).

#### Results of the accuracy

We present the results of the accuracy of each method, without and with a pruning procedure, on each data set and for each level of noise, in Tables 2 and 3 respectively.

#### Performance evaluation without tree post-pruning

In this subsection, we compare our approach, B-CDT, with respect to B-C4.5 in terms of classification accuracy, both without post-pruning methods. We also give the average number of nodes of the trees in the different ensembles.

In Table 4, we depict the average classification accuracy and the average size of the different trees for both ensembles and for the different noise levels. In Fig. 3, these values are graphically represented in dashed lines for Bagging-C4.5R8 ensembles and in continuous lines for Bagging-CDT ensembles.

As can be seen, Bagging-C4.5R8 has a better average performance when no noise is added. And when random noise is intro-

**Table 2**
Classification accuracy of B-C4.5/B-CDT without pruning for the different noise levels.

| Data sets | 0% | 5% | 10% | 20% | 30% |
|---|---|---|---|---|---|
| Anneal | 98.9/98.89 | 98.83/98.78 | 98.05/98.5 | 95.34/97.42 | 89.44/93.97 |
| Audiology | 81.83/80.41 | 81.32/80.36 | 80.84/79.28 | 76.25/75.57 | 73.37/71.51 |
| Autos | 85.45/80.27 | 83.54/78.56 | 80.44/75.79 | 73.34/69.8 | 64.32/62.54 |
| Breast-cancer | 70.43/70.35 | 69.03/70.63 | 67.17/69.87 | 63.4/66.2 | 59.83/61.24 |
| CMC | 52.19/53.21 | 51.17/52.5 | 50.12/51.82 | 48.38/50.14 | 45.41/47.51 |
| Horse-colic | 85.51/84.91 | 85.21/84.15 | 84.55/83.71 | 81.46/80.73 | 76.04/75.16 |
| German-credit | 73.01/74.64 | 72.81/73.96 | 72.67/73.43 | 69.91/71.38 | 65.07/67.19 |
| Pima-diabetes | 76.14/75.8 | 75.88/74.88 | 75.59/74.48 | 74.62/72.6 | 70.8/67.5 |
| Glass2 | 82.22/83 | 80.85/82.15 | 78.45/79.51 | 74.66/76.62 | 68.47/68.51 |
| Hepatitis | 81.76/80.99 | 81.12/81.76 | 80.63/81.53 | 79.35/79.95 | 73.24/75.51 |
| Hypothyroid | 99.62/99.59 | 99.53/99.55 | 99.3/99.48 | 98.34/99.37 | 95.9/98.82 |
| Ionosphere | 92.57/91.23 | 92.25/91.54 | 91.8/90.58 | 87.84/86.7 | 79.86/78.38 |
| kr-vs-kp | 99.46/9.4 | 99.08/99.16 | 98.02/98.72 | 92.68/95.63 | 82.68/86.36 |
| Labor | 86.43/86.53 | 83.67/84.27 | 84.2/84.77 | 79.47/79.53 | 76.33/76.73 |
| Lymphography | 79.96/76.24 | 79.63/77.02 | 79.58/77.02 | 75.99/76 | 73.02/73.14 |
| Mushroom | 100/100 | 99.67/99.99 | 98.82/99.93 | 94.15/98.67 | 84.24/89.06 |
| Segment | 97.75/97.45 | 97.59/97.38 | 96.75/97.08 | 94.29/95.83 | 90.5/93.15 |
| Sick | 98.97/98.97 | 98.68/98.66 | 98.08/98.47 | 96.14/97.87 | 90.34/94.64 |
| Solar-flare | 97/97.31 | 96.29/97.22 | 94.96/97.07 | 90.17/94.93 | 81.55/86.29 |
| Sonar | 80.07/80.78 | 79.43/80.35 | 77.45/79.47 | 74.77/76.27 | 69.52/71.75 |
| Soybean | 92.28/90.47 | 91.95/90.5 | 91.22/90.25 | 88.07/87.7 | 83.45/81.65 |
| Sponge | 93.91/92.63 | 92.75/92.57 | 91.39/92.68 | 87.89/90.57 | 77.45/84.05 |
| Vote | 96.78/96.34 | 96.04/95.79 | 95.22/95.35 | 92.59/93.93 | 86.25/88.87 |
| Vowel | 92.37/91.14 | 91.6/90.61 | 91.41/90.37 | 88.62/88.17 | 83.77/83.91 |
| Zoo | 92.8/92.4 | 93.01/92.5 | 93.66/93.37 | 93.5/93.27 | 89.71/90.71 |

**Table 3**
Classification accuracy of B-C4.5/B-CDT with pruning for the different noise levels.

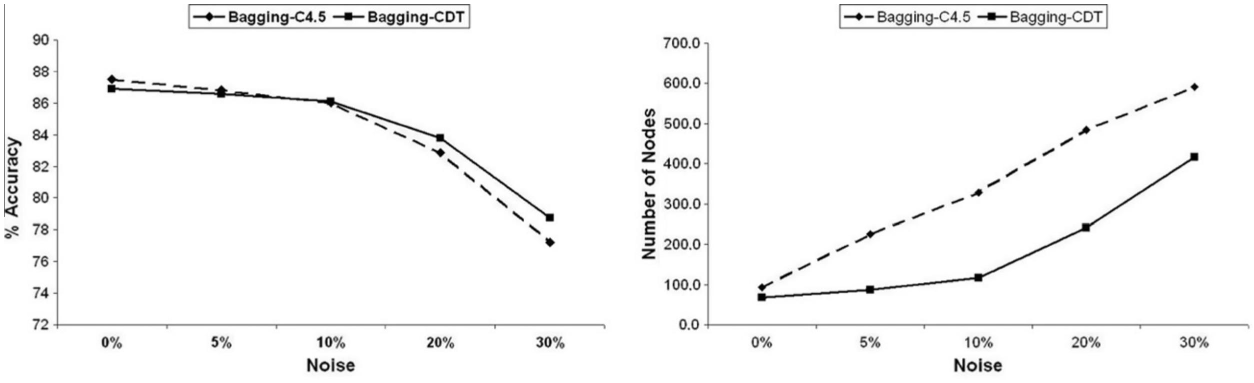
| Data sets | 0% | 5% | 10% | 20% | 30% |
|---|---|---|---|---|---|
| Anneal | 98.79/98.59 | 98.74/98.46 | 98.64/98.36 | 98.04/98.1 | 95.99/97.54 |
| Audiology | 80.75/74.35 | 80.89/75.41 | 81.01/75.68 | 78.37/72.84 | 77.21/69.01 |
| Autos | 84.39/72.65 | 82.71/70.5 | 79.99/67.43 | 73.88/63.51 | 64.91/57.87 |
| Breast-cancer | 73.09/72.35 | 72.25/71.91 | 72.04/71.44 | 70.95/69.94 | 64.31/64.62 |
| CMC | 53.12/56.02 | 52.45/55.07 | 51.56/54.75 | 50.39/53.21 | 47.75/51.32 |
| Horse-colic | 85.21/85.21 | 85.56/85.02 | 85.07/84.64 | 83.96/83.44 | 80.43/78.12 |
| German-credit | 74.73/75.26 | 74.36/74.88 | 73.92/74.66 | 71.9/73.85 | 67.27/70.65 |
| Pima-diabetes | 76.17/75.92 | 75.83/76.34 | 75.53/75.84 | 74.76/75.3 | 71.09/71.53 |
| Glass2 | 81.97/80.44 | 80.98/79.83 | 78.2/79.79 | 75.22/77.62 | 69.01/72.54 |
| Hepatitis | 81.37/81.57 | 81.45/83.03 | 82.06/82.64 | 80.63/81.38 | 75.18/80.33 |
| Hypothyroid | 99.61/99.55 | 99.56/99.53 | 99.5/99.47 | 99.29/99.36 | 97.43/99.15 |
| Ionosphere | 92.54/90.77 | 92.28/91.32 | 91.71/91.35 | 88.01/90.41 | 80.06/84.37 |
| kr-vs-kp | 99.44/98.92 | 99.29/98.92 | 99.17/98.79 | 97.5/97.99 | 88.91/95.14 |
| Labor | 84.63/84.03 | 83.07/84.3 | 81.67/83.43 | 79.63/82.33 | 77.17/80.37 |
| Lymphography | 79.69/77.51 | 78.15/77.78 | 78.63/78.1 | 77.49/77.44 | 75.82/76.53 |
| Mushroom | 100/100 | 100/100 | 99.99/99.98 | 99.84/99.9 | 96.31/98.3 |
| Segment | 97.64/96.74 | 97.58/96.54 | 97.14/96.49 | 95.81/96.28 | 92.33/95.99 |
| Sick | 98.85/98.54 | 98.6/98.46 | 98.43/98.45 | 97.29/98.29 | 92.47/97.25 |
| Solar-flare | 97.84/97.84 | 97.84/97.84 | 97.81/97.81 | 97.66/97.66 | 95.76/96.01 |
| Sonar | 80.4/77.57 | 79.53/76.95 | 77.6/76.99 | 74.86/76.22 | 69.47/73.22 |
| Soybean | 93.1/88.81 | 93.15/88.67 | 92.72/88.45 | 91.93/85.51 | 90.82/81.61 |
| Sponge | 92.63/92.5 | 92.75/92.5 | 92.34/92.5 | 91.79/92.5 | 89.16/91.95 |
| Vote | 96.69/95.52 | 96.16/95.49 | 95.91/95.56 | 95.17/95.49 | 91.54/94 |
| Vowel | 92.14/87.54 | 91.47/86.38 | 91.38/85.83 | 88.76/84.87 | 84/83.11 |
| Zoo | 92.5/92.61 | 92.71/92.7 | 93.77/93.77 | 93.99/91.39 | 91.31/90.53 |

**Table 4**
Average results of B-C4.5 and B-CDT without post-pruning methods (full details per data set can be found in Table 2).

| | Classification accuracy | | | | | Tree size | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0% | 5% | 10% | 20% | 30% | 0% | 5% | 10% | 20% | 30% |
| Bagging-C45 | 87.5 | 86.8 | 86.0 | 82.8 | 77.2 | 93.7 | 225.9 | 328.7 | 484.4 | 591.0 |
| Bagging-CDT | 86.9 | 86.6 | 86.1 | 83.8 | 78.7 | 68.0 | 86.8 | 116.5 | 240.8 | 415.9 |

duced in the training data sets the performance of both ensembles degenerates. However, Bagging-CDT is more robust to the presence of noise and since the 20% of noise level it gets a better average classification accuracy.

Moreover, as can be seen Fig. 3(b), the size of the trees of Bagging-C4.5R8 lineally grows with the different noise levels in opposite to Bagging-CDT where the size of the trees is much lower and almost remains stable until the 10% of noise level.



(a) Classification Accuracy

(b) Tree Size

**Fig. 3.** Average results of B-C4.5 and B-CDT without post-pruning methods.

**Table 5**
Tests results of B-C4.5 and B-CDT without post-pruning methods (see Section 4.1 for details).

| | | | | |
|---|---|---|---|---|
| **Corrected paired-T test:** | | | | |
| **Notation**: w-d-l, number of data sets where B-CDT respectively wins, draws and loses respect to B-C4.5. | | | | |
| 0% | 5% | 10% | 20% | 30% |
| 0-24-1 | 1-23-1 | 3-22-0 | 7-18-0 | 8-16-1 |
| **Sign test:** | | | | |
| **Notation**: $w$, number of data sets where an ensemble is the overall winner; $p$, the significance level. | | | | |
| 0% | 5% | 10% | 20% | 30% |
| $w = 17$, $p < 0.1$ | $w = 14$, $p > 0.1$ | $w = 16$, $p > 0.1$ | $w = 17$, $p < 0.1$ | $w = 19$, $p < 0.05$ |
| **B-C4.5** | – | – | **B-CDT** | **B-CDT** |
| **Wilcoxon signed-ranks test:** | | | | |
| **Notation:** $z$, value of the normal two tailed distribution; $p$, the significance level. | | | | |
| 0% | 5% | 10% | 20% | 30% |
| $z = -2.01$, $p < 0.05$ | $z = -0.47$, $p < 0.1$ | $z = -0.66$, $p > 0.1$ | $z = -2.27$, $p < 0.05$ | $z = -2.70$, $p < 0.01$ |
| **B-C4.5** | – | – | **B-CDT** | **B-CDT** |
| **Friedman test:** | | | | |
| **Notation:** $F$, value of the F-distribution; $p$, the significance level; Rank = (B-CDT, B-C4.5) the average rank. | | | | |
| 0% | 5% | 10% | 20% | 30% |
| 0Rank = (1.7, 1.3) | Rank = (1.56, 1.44) | Rank = (1.36, 1.64) | Rank = (1.32, 1.68) | Rank = (1.24, 1.76) |
| $F = 4.57$, $p < 0.05$ | $F = 0.35$, $p > 0.1$ | $F = 2.04$, $p > 0.1$ | $F = 3.57$, $p < 0.1$ | $F = 8.89$, $p < 0.01$ |
| **B-C4.5** | – | – | **B-CDT** | **B-CDT** |

In Table 5, we carry out an exhaustive comparison of the performance of both ensembles using the set of statistical tests detailed in Section 4.1. As can be seen, when no noise is introduced, the performance of Bagging-C4.5R8 is statistically better than Bagging-CDT, however when the different noise levels are introduced there is a shift in the results. For a low noise level, 5% and 10%, the advantage of Bagging-C4.5R8 disappears and there are no statistical differences between both ensembles. When the noise level is higher, 20% and 30%, Bagging-CDT outperforms Bagging-C4.5R8.

In Table 2, we can see the comparative results about the accuracy of both methods on each data set for each level of noise. We can observe that with 0% of noise the Bagging-C4.5R8 method wins in 17 of the data sets, ties in 2, and loses in 6 with respect to the Bagging-CDT method. This situation changes when more level of noise is added to finish in the contrary situation when 30% of noise is added: Bagging-C4.5R8 wins in only 6 of the data sets and loses in 19 data sets with respect to the Bagging-CDT method.

## Performance evaluation with tree post-pruning

In this subsection, we analyze the performance of both ensembles where the decision trees now apply a post-pruning method (see Sections 2 and 3.3).

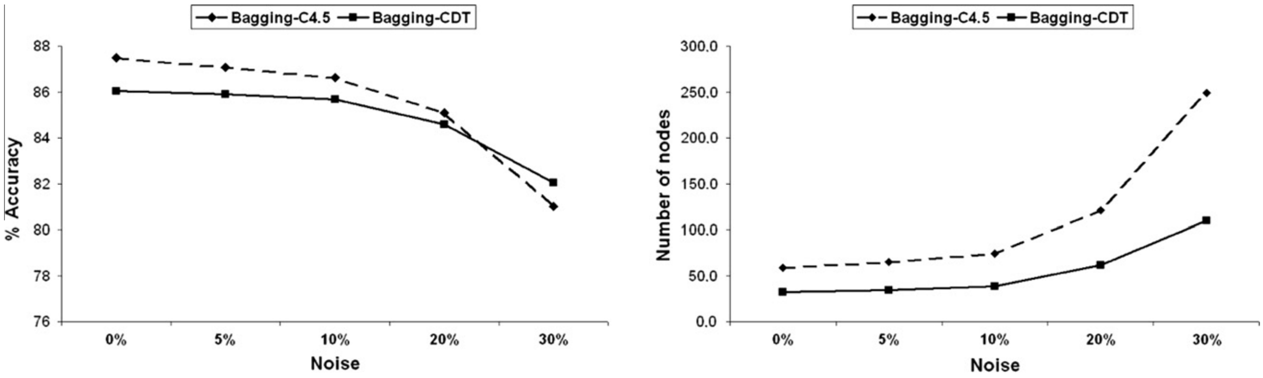
In Table 6, we show the average classification accuracy and the average size of the trees for both ensembles. In Fig. 4 we also graphically show these values.

In Dietterich (2000), there were no conclusions about the suitability to introduce post-pruning methods for decision trees in Bagging ensembles. Our findings are similar, when no noise was introduced there were not statistical significant differences between Bagging-C4.5R8 ensembles with and without post-pruning methods (Wilconxon Signed-Ranks Test, $z = -1.10$, $p > 0.1$). We found the same conclusion for Bagging-CDT (Wilconxon Signed-Ranks Test, $z = -1.48$, $p > 0.1$). However when the noise rate is higher, the introduction of post-pruning methods is worthy and there is statistically significant differences for Bagging-C4.5R8 since 10% of noise level (Wilconxon Signed-Ranks Test, $z = -2.65$, $p < 0.01$). For Bagging-CDT, the introduction of post-pruning methods starts to be statistically significant at 20% of noise level (Wilconxon Signed-Ranks Test, $z = -1.87$, $p < 0.1$) and it is clearly positive at 30% of noise level (Wilconxon Signed-Ranks Test, $z = -1.87$, $p < 0.01$). In this way, Bagging-CDT does not need post-pruning methods with low noise levels.

When we compare Bagging-C4.5R8 with Bagging-CDT the conclusions are mainly the same than in the previous Section 4.1. When no noise is added, Bagging-C4.5R8 performs better than Bagging-CDT but when the noise level increases there is a shift in the

**Table 6**
Average results of B-C4.5 and B-CDT with post-pruning methods (full details per data set can be found in Table 3).

| | Classification accuracy | | | | | Tree size | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0% | 5% | 10% | 20% | 30% | 0% | 5% | 10% | 20% | 30% |
| Bagging-C45 | 87.5 | 87.1 | 86.6 | 85.1 | 81.0 | 58.9 | 65.1 | 74.4 | 120.8 | 249.0 |
| Bagging-CDT | 86.0 | 85.9 | 85.7 | 84.6 | 82.0 | 32.4 | 34.6 | 38.6 | 61.8 | 109.7 |



(a) Classification Accuracy

(b) Tree Size

**Fig. 4.** Average results of B-C4.5 and B-CDT with post-pruning methods.

**Table 7**
Test results of B-C4.5 and B-CDT with post-pruning methods (see Section 4.1 for details).

| | | | | |
|---|---|---|---|---|
| **Corrected paired-T test:** | | | | |
| **Notation**: w-d-l, number of data sets where B-CDT respectively wins, draws and loses respect to B-C4.5. | | | | |
| 0% | 5% | 10% | 20% | 30% |
| 1-17-7 | 1-18-6 | 1-18-6 | 2-19-4 | 9-13-3 |
| **Sign test:** | | | | |
| **Notation**: *w*, number of data sets where an ensemble is the overall winner; *p*, the significance level. | | | | |
| 0% | 5% | 10% | 20% | 30% |
| $w = 18$, $p < 0.05$ | $w = 18$, $p < 0.05$ | $w = 16$, $p > 0.1$ | $w = 17$, $p < 0.1$ | $w = 19$, $p < 0.05$ |
| **B-C4.5** | **B-C4.5** | – | **B-CDT** | **B-CDT** |
| **Wilcoxon signed-ranks test:** | | | | |
| **Notation:** *z*, value of the normal two tailed distribution; *p*, the significance level. | | | | |
| 0% | 5% | 10% | 20% | 30% |
| $z = -3.04$, $p < 0.01$ | $z = -2.12$, $p < 0.01$ | $z = -1.39$, $p > 0.1$ | $z = -0.65$, $p > 0.1$ | $z = -1.87$,$p < 0.1$ |
| **B-C4.5** | **B-C4.5** | – | – | **B-CDT** |
| **Friedman test:** | | | | |
| **Notation:** *F*, value of the F-distribution; *p*, the significance level; Rank = (B-CDT, B-C4.5) the average rank. | | | | |
| 0% | 5% | 10% | 20% | 30% |
| Rank = (1.76, 1.24) | Rank = (1.74, 1.26) | Rank = (1.66, 1.34) | Rank = (1.32, 1.68) | Rank = (1.24, 1.76) |
| $F = 8.89$, $p < 0.01$ | $F = 7.18$, $p < 0.01$ | $F = 2.73$, $p > 0.1$ | $F = 3.07$, $p < 0.1$ | $F = 8.89$, $p < 0.01$ |
| **B-C4.5** | **B-C4.5** | – | **B-CDT** | **B-CDT** |

comparison and for 20% and 30% of noise levels, Bagging-CDT outperforms Bagging-C4.5R8.

As can be seen in Fig. 4(b), the size of the trees in both ensembles also grows with the noise level. However, the introduction of a post-pruning method helps to maintain a lower size for the trees at least with a noise level lower than 10%. When the noise level is higher or equal than 20%, the size of both ensembles quickly grows but for Bagging-CDT this increment is slower, as happened with the unpruned version.

In Table 7, we also carried out the comparison of both ensembles using our wide set of statistical tests. The conclusions are quite similar, Bagging-C4.5R8 outperforms Bagging-CDT with low noise levels (0% and 5%) and when the noise level increases there is a shift in the performances and Bagging-CDT statistically significantly outperforms Bagging-C4.5R8.

We can observe in Table 3, similar situation that the one about the methods without pruning. In this table, we can see the results of the accuracy of both methods with pruning on each data set for each level of noise. We can observe that with 0% of noise the Bagging-C4.5R8 method wins in 18 of the data sets, ties in 3, and loses in 4 data sets with respect to Bagging-CDT method. Again, this situation changes when more level of noise is added to finish in the contrary situation when 30% of noise is added: Bagging-C4.5R8 method wins in only 6 of the data sets and loses in 19 data sets with respect to the Bagging-CDT method.

### 4.2. Bias–variance analysis experimental set-up

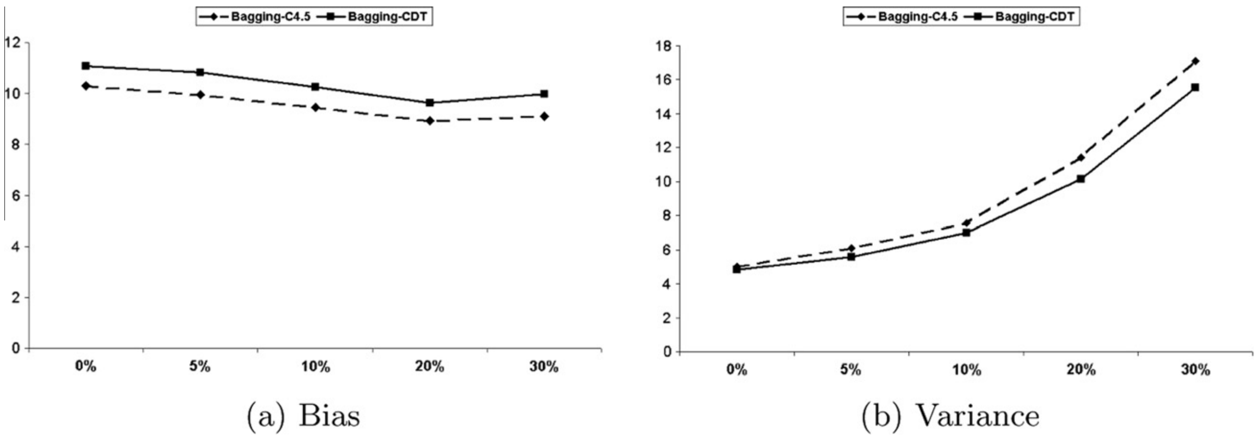
In this section, we attempt to analyze why Bagging-CDT performs better than Bagging-C4.5R8 in situations where there are

**Table 8**
Average bias–variance of B-C4.5 and B-CDT without post-pruning methods.

| | *Bias*$^2$ | | | | | Variance | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0% | 5% | 10% | 20% | 30% | 0% | 5% | 10% | 20% | 30% |
| Bagging-C45 | 10.3 | 9.9 | 9.5 | 8.9 | 9.1 | 5.0 | 6.1 | 7.6 | 11.4 | 17.1 |
| Bagging-CDT | 11.1 | 10.8 | 10.3 | 9.6 | 10.0 | 4.8 | 5.6 | 7.0 | 10.1 | 15.5 |

**Table 9**
Average bias–variance of B-C4.5 and B-CDT with post-pruning methods.

| | *Bias*$^2$ | | | | | Variance | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0% | 5% | 10% | 20% | 30% | 0% | 5% | 10% | 20% | 30% |
| Bagging-C45 | 10.7 | 10.4 | 10.1 | 9.4 | 9.3 | 4.5 | 5.3 | 6.2 | 9.0 | 13.9 |
| Bagging-CDT | 12.7 | 12.6 | 12.2 | 11.7 | 11.6 | 4.2 | 4.7 | 5.4 | 7.4 | 11.0 |



(a) Bias (b) Variance

**Fig. 5.** Bias–variance of B-C4.5 and B-CDT without post-pruning methods.

high noise levels. To do that, we carry out a bias–variance decomposition of the classification error using the decomposition proposed by Kohavi and Wolpert (1996). In this decomposition, the expected error rate of a classifier under different training sets can be decomposed in the following terms:

$$E_D[Loss(Classifier)] = \sum_x P(x)\left(Bias_x^2 + Variance_x + noise\right)$$

where the previous three factors are interpreted as follows:

**Bias**: It represents the systematic component of the error resulting from the incapacity of the predictor to model the underlying distribution. It is computed as the square difference between the target distribution's average output and the classifier's average output across the different training data sets.
**Variance**: It represents the component of the error that stems from the particularities of the training sample (i.e. a measure of overfitting) and can be decreased by the increasing in the size of the data set. So, if the classifier becomes more sensitive to changes in the training data set, the variance increases.
**Noise**: This component measures the underlying variance of the target probability distribution generating the data.

All these components are added to the error and, in consequence, a bias–variance trade-off therefore takes place (Kohavi & Wolpert, 1996): when we attempt to reduce bias by creating more complex models that fit better the underlying distribution of the data, we take the risk of increasing the variance component due to overfitting of the learning data.

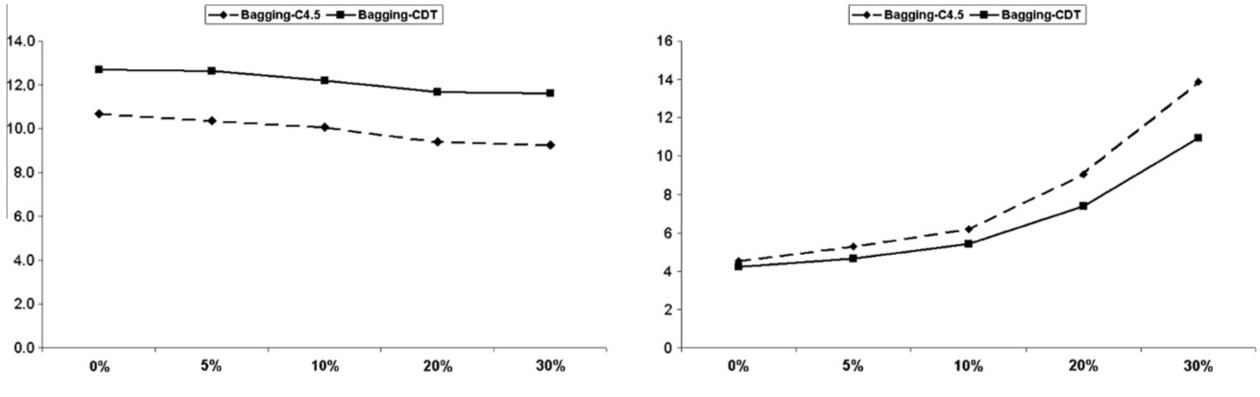
This decomposition was implemented using Weka's utilities and employing the methodology detailed in Webb and Conilione (2005). The data sets, the preprocessing steps and the evaluation methodology were equal to the previously detailed in Section 4.1.

## Experimental analysis

The average percentage values of the decomposition of the classification error in bias and variance components for both ensembles without pruning with post-pruning methods are detailed in Tables 8 and 9, respectively. They are also graphically depicted in Figs. 5 and 6.

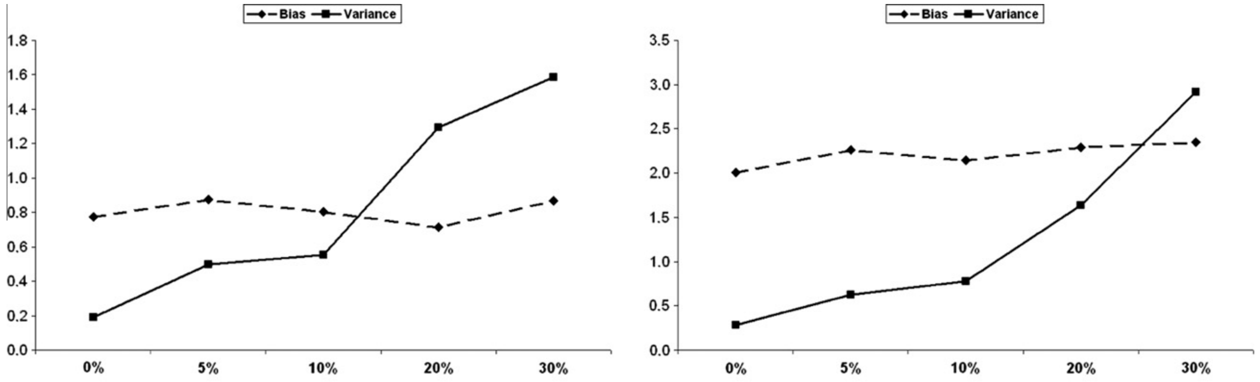
The followings are the main facts that can be found in this first analysis:

- As expected, the introduction of post-pruning methods increases the bias component while reduces the variance, specially in datasets with high noise levels, in both ensembles. That is to say, post-pruning reduces the overfitting of the models, specially when the overfitting risk is higher as happens with high noise levels.
- Bagging-CDT consistently has a higher bias and a lower variance than Bagging-C4.5R8 across different noise rate levels, without and with post-prunning methods as can be seen in Figs. 5 and 6.
- The addition of classification noise produces a strong increment in the variance component of both ensemble and it is the primary responsibility of the degradation of the performance of the classifiers.



(a) Bias (b) Variance

**Fig. 6.** Bias–variance of B-C4.5 and B-CDT with post-pruning methods.



(a) Unpruned Trees (b) Pruned Trees

**Fig. 7.** Absolute bias–variance difference between B-C4.5 and B-CDT.

**Table 10**
Average bias–variance of B-C4.5 and B-CDT with post-pruning methods, using 100, 200, 300 and 500 trees in the Bagging scheme on data sets with high level of noise.

| | *Bias*$^2$ (20% noise) | | | | Variance (20% noise) | | | |
|---|---|---|---|---|---|---|---|---|
| | 100 | 200 | 300 | 500 | 100 | 200 | 300 | 500 |
| Bagging-C45 | 9.4 | 9.5 | 9.4 | 9.4 | 9.0 | 8.9 | 8.9 | 8.9 |
| Bagging-CDT | 11.7 | 11.8 | 11.8 | 11.8 | 7.4 | 7.2 | 7.2 | 7.2 |
| | *Bias*$^2$ (30% noise) | | | | Variance (30% noise) | | | |
| | 100 | 200 | 300 | 500 | 100 | 200 | 300 | 500 |
| Bagging-C45 | 9.3 | 9.3 | 9.2 | 9.3 | 13.9 | 13.6 | 13.7 | 13.6 |
| Bagging-CDT | 11.6 | 11.7 | 11.8 | 11.6 | 11.0 | 10.7 | 10.5 | 10.6 |

The previous facts can be explained by the different methods which employ Bagging-CDT and Bagging-C4.5R8 in order to approximate the underlying distribution of the data and make the classification predictions. The main difference is that Bagging-CDT are based on the Imprecise Info-Gain score that measures the maximum entropy of a credal set, in opposite to the classic Information Gain that measure the expected entropy. So, Imprecise Info-Gain is a more conservative measure than classic Info-Gain.

Let us see how the different behaviors of these two strategies can be measured by means of this bias–variance error decomposition. In Fig. 7, we show the absolute difference between the bias (in dashed line) and the variance (in continuous line) for both Bagging ensembles and when there is no post-pruning methods (Fig. 7(a)) and when there is post-pruning methods (Fig. 7(b)). To understand the meaning of these figures, we must firstly consider that Bagging-CDT systematically has a lower variance and higher bias than Bagging-C4.5R8 in all the cases (see Figs. 5 and 6).

As can be seen, when no noise is added, the more conservative strategy of Bagging-CDT obtains a lower variance (there is an approximate difference of 0.2 in favor of B-CDT) but a much higher bias component (there is an approximate difference of 0.8), what results in a worse classification accuracy (the improvement in the variance in not compensated by the loses in the bias).

However, when the datasets contains higher noise levels, this conservative strategy of Bagging-CDT is favored and the bad bias values it obtains are compensated by stronger improvements in the variance component. As can be seen, the difference in the bias (dashed lines) between both ensembles remains stable across the different noise levels while the difference in the variance (continuous lines) strongly increases with higher noise levels.

As conclusion we have that, in datasets with high noise levels, the variance of Bagging-CDT does not increases so quickly than in the case of Bagging-C4.5R8. This also corresponds to the generation of smaller decision trees as can be seen in Fig. 3(b) and Fig. 4(b). In consequence, this is the reason why Bagging-CDT outperforms Bagging-C4.5R8 in data sets with high classification noise rates, because it has a lower variance in this situations.

In order to strengthen the previous conclusions another experiment was accomplished to discard that the increasing of the variance error for the Bagging-C4.5R8 method on data sets with high levels of classification noise could be reduced taking an upper

**Table 11**
Average bias–variance of B-C4.5 and B-CDT without post-pruning methods, using 100, 200, 300 and 500 trees in the Bagging scheme on data sets with high level of noise.

| | *Bias*$^2$ (20% noise) | | | | Variance (20% noise) | | | |
|---|---|---|---|---|---|---|---|---|
| | 100 | 200 | 300 | 500 | 100 | 200 | 300 | 500 |
| Bagging-C45 | 8.9 | 9.1 | 9.1 | 9.0 | 11.4 | 11.3 | 11.2 | 11.1 |
| Bagging-CDT | 9.6 | 9.7 | 9.7 | 9.8 | 10.1 | 10.0 | 9.9 | 9.9 |
| | *Bias*$^2$ (30% noise) | | | | Variance (30% noise) | | | |
| | 100 | 200 | 300 | 500 | 100 | 200 | 300 | 500 |
| Bagging-C45 | 9.1 | 9.1 | 9.0 | 9.1 | 17.1 | 16.9 | 16.9 | 16.8 |
| Bagging-CDT | 10.0 | 9.9 | 9.8 | 9.9 | 15.5 | 15.5 | 15.5 | 15.5 |

number of trees into the Bagging scheme. To analyze this possibility, we repeated the experiments with 100, 200, 300 and 500 decision trees into the Bagging schemes on the data sets with 20% and 30% of classification noise. The results of the values of the bias and variance can be seen in Tables 10 and 11.

As we can see, it exists a little bit decreasing in the variance values in both methods when we increase the number of trees, but the differences of the errors remain constant in favor of our method. Hence, we can say that the number of trees used is not an important parameter for the comparison of the methods.

## 5. Conclusions

In this paper, we have extended a method to build decision trees based on imprecise probabilities and uncertainty measures (called credal decision trees), to one which works with continuous features and missing data. As an application of this type of decision trees, we have use them into a Bagging scheme on data sets with class noise. In our an experimental study, we have proved that our method of Bagging credal decision trees can reduce the percentage of error in classification when it is applied on data sets with medium–high level of classification noise.

As reference, we have used a similar scheme using decision trees built with the known C4.5 procedure, which has a number of fix parameters to improve the accuracy. In the literature, we can see that Bagging classification methods using C4.5 procedure can obtain excellent results when it is applied on data sets with classification noise.

In a bias–variance study, we have prove that the Bagging credal decision trees method maintains the variance error component lower when the noise level is increased and, in consequence, obtains a better performance. In addition to this, our approach is more simple and requires a less number of parameters.

## Acknowledgments

This work has been supported by the Spanish “Consejería de Economía, Innovación y Ciencia de la Junta de Andalucía”, “Ministerio de Educación y Ciencia” and by the Spanish research programme Consolider Ingenio 2010 under Projects TIC-06016, TIN2010–20900-C04–01 and MIPRCV (CSD2007-00018), respectively.

A shorted version of this paper has been presented in FOIKS’10.

## References

Abellán, J. (2011). Combining nonspecificity measures in Dempster–Shafer theory of evidence. *International Journal of General Systems, 40*(6), 611–622.

Abellán, J. (2006). Uncertainty measures on probability intervals from Imprecise Dirichlet model. *International Journal of General Systems, 35*(5), 509–528.

Abellán, J., & Moral, S. (2003). Maximum entropy for credal sets, International Journal of Uncertainty. *Fuzziness and Knowledge-Based Systems, 11*(5), 587–597.

Abellán, J., & Moral, S. (2003). Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems, 18*(12), 1215–1225.

Abellán, J., & Moral, S. (2006). An algorithm that computes the upper entropy for order-2 capacities, International Journal of Uncertainty. *Fuzziness and Knowledge-Based Systems, 14*(2), 141–154.

Abellán, J., & Moral, S. (2005). Upper entropy of credal sets. Applications to credal classification. *International Journal of Approximate Reasoning, 39*(2-3), 235–255.

Abellán, J., Klir, G. J., & Moral, S. (2006). Disaggregated total uncertainty measure for credal sets. *International Journal of General Systems, 35*(1), 29–44.

Abellán, J., & Masegosa, A. R. (2009). An Experimental Study about Simple Decision Trees for Bagging Ensemble on Datasets with Classification Noise. In C. Sossai & G. Chemello (Eds.). *ECSQARU of Lecture Notes in Computer Science* (Vol. 5590, pp. 446–456). Springer.

Abellán, J., & Masegosa, A. R. (2008). Requirements for total uncertainty measures in Dempster-Shafer theory of evidence. *International Journal of General Systems, 37*(6), 733–747.

Abellán, J., & Masegosa, A. R. (2010). An ensemble method using credal decision trees. *European Journal of Operational Research, 205*(1), 218–226.

Bernard, J. M. (2005). An introduction to the imprecise Dirichlet model for multinomial data. *International Journal of Approximate Reasoning, 39*, 123–150.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Belmont: Wadsworth Statistics, Probability Series.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Demsar, J. (2006). Statistical comparison of classifiers over multiple datasets. *Journal of Machine Learning Research, 7*, 1–30.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning, 40*(2), 139–157.

Elvira: An Environment for Creating and Using Probabilistic Graphical Models. In *Proceedings of the First European Workshop on Probabilistic Graphical Models (PGM’02), Cuenca (Spain)* (pp. 1–11) 2002.

Freund, Y. Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning, San Francisco* (pp. 148–156).

Klir, G. J. (2006). *Uncertainty and Information: Foundations of Generalized Information Theory*. Hoboken, NJ: John Wiley.

Kohavi, R. Wolpert, D. (1996). Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth International Conference of Machine Learning* (pp. 275–283).

Prem Melville, Nishit Shah, L. M. R. J. M. (2004). Experiments on ensembles with missing and noisy data. In *Proceedings of 5th International Workshop on Multiple Classifier Systems, Springer Verlag* (pp. 293–302).

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Quinlan, J. (1987). Simplifying decision trees. *International Journal of Machine Learning Studies, 27*, 221–234.

Quinlan, J. R. (1993). Programs for Machine Learning. Morgan Kaufmann series in Machine Learning.

Walley, P. (1996). Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B, 58*, 3–57.

Webb, G. Conilione, P. (2005). Estimating bias and variance from data, Technical Report.

Witten, I. H., & Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques. *2nd ed*. San Francisco: Morgan Kaufmann.