Expert Systems with Applications 41 (2014) 3825–3830

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring

Joaquín Abellán *, Carlos J. Mantas

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

ARTICLE INFO

Keywords:
Bankruptcy prediction
Credit scoring
Ensembles of classifiers
Decision trees
Imprecise Dirichlet model

ABSTRACT

Previous studies about ensembles of classifiers for bankruptcy prediction and credit scoring have been presented. In these studies, different ensemble schemes for complex classifiers were applied, and the best results were obtained using the Random Subspace method. The Bagging scheme was one of the ensemble methods used in the comparison. However, it was not correctly used. It is very important to use this ensemble scheme on weak and unstable classifiers for producing diversity in the combination. In order to improve the comparison, Bagging scheme on several decision trees models is applied to bankruptcy prediction and credit scoring. Decision trees encourage diversity for the combination of classifiers. Finally, an experimental study shows that Bagging scheme on decision trees present the best results for bankruptcy prediction and credit scoring.

1. Introduction

In supervised classification tasks, the combination or ensemble of classifiers represents an interesting way of merging information that can provide a better accuracy than each individual method. The high classification accuracy performance of these combined methods makes them very suitable for real world applications, such as bankruptcy prediction and credit scoring.

In the paper of Nanni and Lumini (2009) it is presented an interesting analysis about previous papers on bankruptcy prediction and credit scoring. The importance of this type of real application is well exposed: (a) The credit scoring models permit to discriminate between good credit group and bad credit group; (b) Developing a reliable credit scoring system offers several benefits, including cost reduction of credit analysis, delivery of faster decisions, guaranteed credit collection, and risk mitigation.

In this paper (Nanni & Lumini, 2009), the authors analyzed some well established financial decision-making methods based on machine learning to solve the financial decision-making problems mentioned above. In that work, the individual methods providing better performance were based on Artificial Neural Networks (ANNs). They also presented a thorough study about several techniques to create ensembles of classifiers based on some complex classifiers, including ANNs. All of them were applied to data sets related to the problem of bankruptcy prediction and credit scoring, with the aim of outperforming previous works, like in Tsai and Wu (2008).

It is important to highlight that some schemes to create classifier ensembles do not have to be based on very complex and accurate individual classifiers. For example, Bagging scheme (Breiman, 1996) is a well known procedure for creating ensembles of classifiers that performs best when applied to weak and unstable classifiers. However, this fact was forgotten in the previous studies about bankruptcy prediction and credit scoring.

Decision trees (DTs) represent a family of simple classifiers that can be built in very little time and have a simple structure which can be interpreted easily. An important aspect of DTs, which make them very suitable for ensembles of classifiers, is their instability: Different training sets from a given problem domain will produce very different models. Hence, DTs encourage diversity for the combination of classifiers (Breiman, 1996) and provide an excellent model for the Bagging ensemble scheme.

In Abellán and Masegosa (2012), it is shown that using Bagging ensembles on a special type of decision trees, called credal decision trees (CDTs) (Abellán & Moral, 2003b), provides an interesting tool for the classification task. CDTs are based on imprecise probabilities (more specifically, on the Imprecise Dirichlet Model (IDM); see Walley, 1996) and information/uncertainty measures (in particular, on the maximum of entropy function; see Klir, 2006, Abellán, 2011). An important characteristic of the CDT procedure is that the split criterion used to build a DT has a different treatment of the imprecision than the one used for the classic split criteria.

Hence, the main purpose of this paper is to complete previous works, especially the one presented in Nanni and Lumini (2009)

* Corresponding author. Tel.: +34 958 242376.
E-mail addresses: jabellan@decsai.ugr.es (J. Abellán), cmantas@decsai.ugr.es (C.J. Mantas).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.12.003

about the use of Bagging ensembles on DTs. We show that the use of CDTs in a Bagging scheme outperforms previous results for data sets related to bankruptcy prediction and credit scoring. We have used the same setting employed in Nanni and Lumini (2009): Same data sets, same type of experimentation and same measure to compare results. Moreover, we have used known statistical tests to support our results.

In order to compare the performance of the mentioned procedures in a logical way, we have used the best ensemble method described in previous works (the Random Subspace ensemble procedure (Ho, 1998)) and the Bagging scheme on DTs. In addition, the trees were built with the most successful classification method based on DTs: Quinlan's C4.5 algorithm (Quinlan, 1993); and the mentioned CDT procedure (Abellán & Moral, 2003b).

This paper is organized as follows: In Section 2, we present the necessary background about DTs, CDTs, ensemble methods and previous works made on data sets concerning bankruptcy prediction and credit scoring; in Section 3, we describe and comment on the results of the experiments carried out; and finally, Section 4 presents the conclusions.

2. Background

2.1. Decision trees

The classification task is focused on elements that are described by one or more characteristics, known as *attribute variables* (also called *predictive attributes* or *features*), and by a single *class variable*, with the aim to predict the class value of a new element by considering its attribute values.

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 *Iterative Dichotomiser 3*, known as ID3 (Quinlan, 1986). Subsequently, Quinlan also presented the *Classifier 4.5*, known as C4.5 (Quinlan, 1993), which is an advanced version of the 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 partitioning method that divides the data set using a single variable at each level. This variable is selected by means of a given criterion. Ideally, these models define sets of cases in which all the cases in a set 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 labeled with an attribute variable that produces a different branch for each variable value (i.e., a partition of the data set). Leaf nodes are labeled with a class label.

The process for inferring a decision tree is mainly determined by the following points: (i) The criteria used to select the attribute that should be placed in a node and branched; (ii) The criteria for stopping the tree branching process; (iii) The method for assigning a class label or a probability distribution to the leaf nodes; and (iv) The posterior 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) stand out among the most popular ones.

Decision trees are built using a data set referred to as the training data set. A different set, called the test data set, is used to check the model. When we get a new sample or instance of the test data set, we can make a decision or prediction about the state of its class variable, following the path in the tree from the root to a leaf node using the sample values and the tree structure.

2.2. Credal decision trees

2.2.1. Mathematical foundations

The split criterion employed to build CDTs (Abellán & Moral, 2003b) is based on the application of uncertainty measures on convex sets of probability distributions (credal sets). 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 set of probability distributions (see Abellán, 2006a).

The IDM depends on a given hyperparameter s which does not depend on the sample space (Walley, 1996). The IDM estimates that the probabilities for each value of the class variable C are within an interval defined by:

$$p(c_j) \in \left[\frac{n_{c_j}}{N+s}, \frac{n_{c_j}+s}{N+s}\right], \quad j = 1, \ldots, k;$$

with n_{c_j} as the frequency of the set of values $(C = c_j)$ in the data set, N the sample size and k the number of cases in class variable C. One important thing is that intervals are wider if the sample size is smaller. So this method produces more precise intervals as N increases.

Walley (1996) does not give a definitive recommendation for the value of the parameter s, but he suggests two candidates: $s = 1$ or $s = 2$. In our case, we will use a value $s = 1$. The reason for this is the low computational cost of the inference with credal sets for $s = 1$, as it will be shown in the following paragraph.

The entropy of this set of probability intervals will be estimated as the maximum of the entropy of all probability distributions $(q(c_1), \ldots, q(c_k))$ verifying that, for any c_i, $q(c_i)$ belongs to the estimated interval for $p(c_i)$. For $s = 1$ this entropy is very simple to compute. First, we have to determine $A = \{c_j : n_{c_j} = \min_i\{n_{c_i}\}\}$. If l is the number of elements of A, then the distribution with maximum entropy is p^*, where $p^*(c_i) = \frac{n_{c_i}}{N+s}$ if $c_i \notin A$ and $p^*(c_i) = \frac{n_{c_i}+s/l}{N+s}$ if $c_i \in A$.

This upper entropy function, denoted as S^*, is a total uncertainty measure which is well known for this type of set (see Abellán & Moral, 2003a; Abellán, Klir, & Moral, 2006; Abellán & Masegosa, 2008).

As the intervals are wider with smaller sample sizes, there will be a tendency to get greater maximum entropy values with smaller sample sizes. This property will be important to differentiate the action of the CDTs from the behavior of other kinds of DTs.

For example, if we assume $k = 2$, then the information obtained from a node A with $n_{c_1} = 400$ and $n_{c_2} = 100$ will be more reliable than that provided by a node B with $n_{c_1} = 4$ and $n_{c_2} = 1$, since the first node has a larger sample size. This fact is taken into account by the maximum entropy function (the first node will have the lowest uncertainty); in contrast, using classic entropy both nodes will have the same uncertainty.

Using S to denote the classic entropy function on a probability distribution p, and K^A, K^B to denote the sets of probabilities via the IDM (with $s = 1$) associated to nodes A and B mentioned above, we have that:

$$S\left(\frac{400}{500}, \frac{100}{500}\right) = S\left(\frac{4}{5}, \frac{1}{5}\right) = 0.5004,$$

$$S^*(K^A) = 0.5025; \quad S^*(K^B) = 0.6365$$

Hence

$$S\left(\frac{400}{500}, \frac{100}{500}\right) = S\left(\frac{4}{5}, \frac{1}{5}\right) \cong S^*(K^A) < S^*(K^B),$$

This shows that the treatment of imprecision is clearly different with the new split criterion based on imprecise probabilities of

the CDT procedure compared to the classic procedure based on the entropy function. This is the reason that different DTs can be obtained using classic entropy (as in the ID3 procedure) and upper entropy via the IDM (as in the CDT procedure).

Not surprisingly, for the root node and higher level nodes (the ones close to the root node) the classic criterion based on entropy is similar to the one used by the CDT procedure. The difference is more evident for lower level nodes. It makes sense, since probabilities based on high frequencies are more reliable than those based on low frequencies, as shown in the above example.

2.2.2. Method for building credal decision trees

The procedure for building CDTs is very similar to the method used in the well-known Quinlan's ID3 algorithm (Quinlan, 1986). All we need is to replace the criterion used in ID3, the *Info-Gain* split criterion, with the *Imprecise Info-Gain* (IIG) split criterion. The IIG criterion can be defined as follows: In a classification problem, let C be the class variable, $\{Z_1,\ldots,Z_n\}$ the set of features, and Z a feature; then

$$IIG^{\mathscr{D}}(C,Z) = S^*(K^{\mathscr{D}}(C)) - \sum_i P(Z=z_i)S^*(K^{\mathscr{D}}(C|Z=z_i)),$$

where $K^{\mathscr{D}}(C)$ and $K^{\mathscr{D}}(C|Z=z_i)$ are the credal sets obtained via the IDM for variables C and $(C|Z=z_i)$ respectively, for a partition $\mathscr{D}$ of the data set (see Abellán & Moral, 2003b); and S^* is the maximum entropy function (see Abellán, 2006a).

The IIG criterion differs from the classical criteria. It is based on the principle of maximum uncertainty (see Klir, 2006), broadly used in classical information theory, where it is known as the maximum entropy principle (Jaynes, 1982). The use of the maximum entropy function in the procedure for building decision trees is justified in Abellán and Moral (2005). It is important to note that for a feature Z and a partition $\mathscr{D}$, $IIG^{\mathscr{D}}(C,Z)$ can be negative. This situation does not happen with classical split criteria such as the Info-Gain criterion used in ID3.[1] This characteristic enables the IIG criterion to reveal features that worsen the information on the class variable.

Each node *No* in a decision tree produces a partition of the data set (for the root node, $\mathscr{D}$ is considered to be the entire data set). Furthermore, each node *No* has an associated list $\mathscr{L}$ of feature labels (that are not in the path from the root node to *No*). The procedure for building credal trees is explained through the algorithm in Fig. 1.

Considering this algorithm, when an *Exit* situation is attained, i.e. when there are no more possible features to introduce in a node, or when the uncertainty is not reduced (steps 1 and 4 of the algorithm), a leaf node is produced.

In a leaf node, the most probable state or value of the class variable for the partition associated with that leaf node is inserted. To avoid obtaining unclassified instances, if we do not have one single most probable class value, we can select the one obtained in its parent node (see Abellán & Masegosa, 2010).

In the original procedure for building a CDT there is no pruning process. In a subsequent extension, a posterior pruning process similar to that used in Quinlan's C4.5 algorithm (Quinlan, 1993), has been implemented and incorporated to the CDT method (see Abellán & Masegosa, 2012). This extension has been used for the experiments carried out in this work.

CDTs can be somewhat smaller than the trees produced with similar procedures by replacing the IIG criterion with the classic ones (see Abellán & Masegosa, 2009b). This generally reduces the overfitting of the model (see Abellán & Moral, 2005), and could impair its use in Bagging schemes (see Provost & Domingos, 2003).

[1] The Info-Gain criterion is actually a particular case of the IIG criterion using the parameter $s=0$.

Procedure `BuildCredalTree`$(No,\mathcal{L})$

1. If $\mathcal{L}=\emptyset$, then `Exit`.
2. Let $\mathcal{D}$ be the partition associated with node No
3. Compute the value
   $\alpha = \max_{Z_j\in\mathcal{L}}\{IIG^{\mathcal{D}}(C,Z_j)\}$
4. If $\alpha \leq 0$ then `Exit`
5. Else
   6. Let Z_l be the variable for which the maximum α is attained
7. Remove Z_l from $\mathcal{L}$
8. Assign Z_l to node No
9. For each possible value z_l of Z_l
   10. Add a node No_l
   11. Make No_l a child of No
   12. Call `BuilCredalTree`$(No_l,\mathcal{L})$

Fig. 1. Procedure for building CDTs.

Credal trees and IIG criterion have been successfully used in other data mining procedures and tools: As part of a procedure for selecting variables (Abellán & Masegosa, 2009a); and on data sets with classification noise (Abellán & Masegosa, 2009b). Also, an extended version of the IIG criterion has been used to define a semi-nave Bayes classifier (Abellán, 2006b).

2.3. Ensemble of classifiers

In many areas of science, such as Medicine or Finances, a second opinion (sometimes even more) is often sought before a decision is taken. Eventually, the individual opinions are weighted and combined to take a final decision that, in principle, should be more robust and reliable. This process of consulting "several experts" before making a decision has also been 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 proven to produce much better results than individual classifiers.

The general idea normally used in the procedures for combining classifiers is based on the generation of a set of different classifiers combined with a majority vote criterion. When a new unclassified instance arises, each single classifier 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 classifiers are quite similar, the ensemble performance will not be much better than that of an individual classifier. However, if the ensemble is made up of a broad set of different classifiers with a good individual performance, the ensemble will become more robust and will have a better prediction capacity.

There are many different approaches to this problem. For this work, we have used the Random Subspace ensemble procedure (Ho, 1998) that is the best ensemble method according to (Nanni & Lumini, 2009), and the Bagging scheme, which is well suited for combining DTs (Breiman, 1996). These schemes can be briefly described as follows:

- *Random Subspace* (Ho, 1998): This method uses several classifiers built on randomly chosen subspaces of the original input space, and combine them into a final decision rule via a simple majority vote procedure. Each single classifier uses only a subset of all features available in the data set for training and testing. These features are chosen uniformly at random from the full set of features. The standard number of features used to obtain good results is the half of the total number of features (as empirical studies have suggested).

- *Bagging* (Breiman, 1996): It stands for *Bootstrap Aggregating* ensemble method. It is an intuitive and simple method that offers and excellent performance. Diversity in Bagging is obtained by using bootstrapped replicas of the original training set: Different training data sets are randomly drawn with replacement. Subsequently, a classifier is built for each instance of training data using the standard approach (Breiman, Friedman, Olshen, & Stone, 1984). Finally, their predictions are combined by a majority vote.

The previous ensemble methods are used in Nanni and Lumini (2009) with complex and accurate individual classifiers as the basis. However, as it has been mentioned, using a weak and unstable classifier as the basis of a Bagging scheme is essential (Breiman, 1996). A DT procedure normally builds very different models when applied to different training data sets. This fact makes DTs well suited as the basis of a Bagging scheme.

Besides, the DT building procedure has an inherent variable selection method based on uncertainty measures. This positive property is partially lost when the Random Subspace ensemble scheme is used with DT classifiers as the basis. In order to illustrate this fact, we have considered interesting to add Random Subspace schemes on DTs for the experiments presented in this paper.

There are many different approaches to the implementation of ensemble schemes on DTs. Dietterich (2000) employed a Bagging scheme on DTs built using the C4.5 algorithm (with and without posterior pruning) to show that Bagging as stands out as an outperforming ensemble approach for data sets with classification noise. But there was no definitive suggestion about the employment of pruning. In Abellán and Masegosa (2012) it is shown that the performance of that procedure can be improved using CDTs, i.e. DTs built via the IDM and the maximum entropy function. The models compared in that study perform better when posterior pruning procedures are applied. Hence, this last option will be chosen to carry out the experiments described in this paper.

2.4. Previous works on ensemble schemes for data sets related to bankruptcy prediction and credit scoring

As a reference, we use the paper of Nanni and Lumini (2009) for the following reasons: It presents an interesting analysis about previous papers on ensemble schemes for data sets related to bankruptcy prediction and credit scoring; and, via an experimental study, earlier works presented in the literature for the mentioned type of data sets (like Tsai & Wu, 2008) are improved. In these works, it is shown that systems based on machine learning are better than the traditional ones, based on statistics.

According to Nanni and Lumini (2009), the previous works about bankruptcy prediction and credit scoring have the following issues: (i) generally, only one data set is used to validate the model; (ii) generally, only the accuracy is used to check the performance of the model. The following recipe was proposed to solve these drawbacks (see Nanni & Lumini (2009) for more details and references to schemes and methods):

- Use the following successful schemes for creating an ensemble of classifiers: Bagging, Random Subspace, Class Switching, and Rotation Forest.
- Use the following successful and complex classifiers as the basis for each ensemble scheme: Levenberg–Marquardt neural net, Multi-layer perceptron neural net, Radial Basis function Support Vector Machine and 5-nearest neighbor method.
- Use the following data sets: Japanese credit, Australian credit and German credit.
- Apply 50 times a 70–30% training-test procedure for classification: A data set is split in 70% of data for training and 30% for testing; then the data set is randomly reordered and the procedure is repeated. This procedure is used on 50 data sets for training and their corresponding 50 data sets for testing, and the average results are reported.
- Compare the results using primarily the *Area Under the Receiver Operating Characteristic curve* (AUC) measure.[2]

Table 1
Description of the data sets.

Data sets	*N*	Feat	*k*
Japanese credit	690	15	2
Australian credit	690	14	2
German credit	1000	24	2

Table 2
AUC average values on the data sets for each method.

Data set	RS-LNMC	B-CDT	B-C4.5	RS-CDT	RS-C4.5
Japanese credit	0.9285	0.9357 ○	0.9326 ○	0.9315 ○	0.9313 ○
Australian credit	0.9338	0.9364 ○	0.9301 •	0.9296 •	0.9285 •
German credit	0.7847	0.7860 ○	0.7854 ○	0.7792 •	0.7826 •

○, • Improvement or degradation.

The conclusion presented in Nanni and Lumini (2009) is that the best ensemble scheme is a Random Subspace (RS) of Levenberg–Marquardt neural nets (LNMC). This scheme performed better than the Bagging scheme applied to the same base classifier. This is not the only successful application of Artificial Neural Network to real world problems. Others can be found in Muttil and Chau (2006), Taormina, Chau, and Sethi (2012) and Wu, Chau, and Li (2009).

It should be noted that the Bagging scheme is an excellent way to combine classifiers, but as it was mentioned in previous sections, it must be used on unstable and weak classifiers (e. g., DTs). This characteristic has not been taken into account in Nanni and Lumini (2009), where complex and stable classifiers were used as the basis. Hence, that work presents a poor performance of the Bagging scheme.

In the next section, we present the use of a Bagging scheme of DTs applied to data sets related to bankruptcy prediction and credit scoring. These models are compared with the best models provided by Nanni and Lumini (2009). In this manner, the veracity of the previous paragraph is confirmed.

3. Experimental analysis

In this experimental study, our main goal is to compare the best procedure presented in Nanni and Lumini (2009) with Bagging ensemble of DTs. These trees will be built using the CDT method exposed in Section 2.2 and using the well known C4.5 algorithm (Quinlan, 1993). Both methods will be used with the same pruning approach (see Abellán & Masegosa, 2012).

We have used the same setting exposed in Section 2.3. The data sets were provided by the authors of paper (Nanni & Lumini, 2009). A brief description of these data sets can be found in Table 1, where column "*N*" is the number of instances in the data sets, column "Feat" is the number of features,[3] and column "*k*" is the number

[2] Other measures are used in that paper, but AUC is exposed as the most important and reliable measure for such types of experimentation and data sets. AUC is a measure of classification performance that takes into account the True Positive rate against the False Positive rate. In the general machine learning literature it is acknowledged as one of the best measures for comparing classifiers in two-class problems (see Beck & Schultz, 1986; Fawcett, 2004).

[3] There was a misprint in Nanni and Lumini (2009) regarding the number of Features in the *German credit* data set; the correct value is 24.

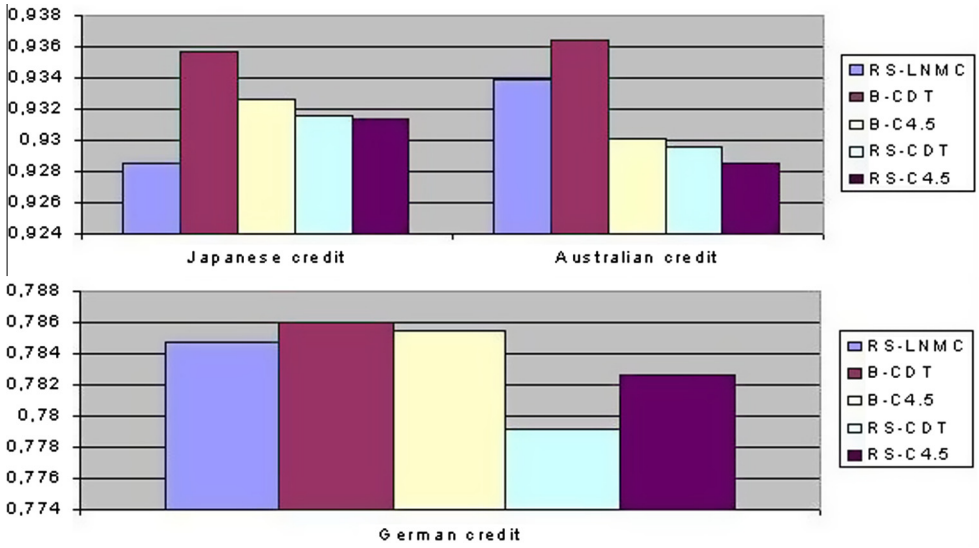

Fig. 2. AUC measures of the methods for each data set.

of cases or states of the class variable (always a nominal variable) of each data set.

For the experimentation we used the *Weka* software (Witten & Frank, 2005). Our Bagging ensembles of CDTs were implemented using Weka data structures. The IDM parameter was set to $s = 1$, for which the procedure used to obtain the IIG value via the maximum entropy function attains its lower computational cost (see Section 2.2).

All experiments were carried out with the setting presented in Nanni and Lumini (2009) (described in the previous section). Each iteration used 50 DTs.

To complete the experimental comparison of the mentioned procedures, we considered that adding Random Subspace schemes (the best ensemble method according to previous works) on DTs could be of interest. These trees are also built using the CDT and C4.5 procedures. Hence, the methods that we used for our experimental study were the following:

(1) Random Subspace on Levenberg–Marquardt neural nets (the best one according to Nanni & Lumini, 2009), denoted by RS-LMNC.
(2) Bagging decision trees built using the CDT procedure, denoted by B-CDT.
(3) Bagging decision trees built using the C4.5 procedure, denoted by B-C4.5.
(4) Random Subspace on decision trees built using the CDT procedure, denoted by RS-CDT.
(5) Random Subspace on decision trees built using the C4.5 procedure, denoted by RS-C4.5.

We have used the AUC measure (the main one in Nanni & Lumini, 2009) as the measure to compare results. From this measure, the following known statistical tests with a level of significance of $\alpha = 0.1$ have been used to compare all the procedures (see Demsar, 2006 for a complete explanation and further references to these statistical tests):

Friedman test: A non-parametric test that ranks the algorithms for each data set separately, the best performing algorithm getting the rank of 1, the second best rank 2, etc. The null hypothesis is that all the algorithms are equivalent. If the null-hypothesis is rejected, we can compare the best procedure with the others by using the post hoc **Bonferroni–Dunn test.**

In Table 2, and graphically in Fig. 2, we can see the results of the methods on each data set via the AUC measure. It also presents the improvement or degradation of the results for each method compared to the results presented by the RS-LNMC method, which was used as a reference.

The main result observed is that the winner method in the study of Nanni and Lumini (2009) is beaten by B-CDT in the same setting for all the data sets. Even the B-C4.5 method is better than RS-LNMC for two data sets (and worse for one).

The methods based on a RS scheme on DTs are worse compared to the RS-LNMC method: They are worse for two data sets and better for one of them. Also, it must be remarked that the B-CDT method got the best results for each data set when compared to every other method.

It can be observed that the Random Subspace scheme on the LNMC procedure is better than when it is applied on DTs. Besides, the Bagging ensemble is better than the Random Subspace one when both are applied on DTs; i.e. B-CDT is better than RS-CDT, and B-C4.5 is better than RS-C4.5 for each data set. These results are consistent according to the comments in previous sections, that is, the Random Subspace ensemble scheme partially loses the benefit of the variable selection method associated to the DT build process.

From the previous paragraph and for reasons of clarity, only the three best methods (RS-LNMC, B-CDT and B-C4.5) were compared in this experimental study via a set of tests. We carried out Friedman's test and the ranks obtained for each method are presented in Table 3.

As expected, the best rank was obtained by the B-CDT method, followed by B-C4.5 and RS-LNMC. The $p - value$ of Friedman's test is 0.09697. Hence, the null hypothesis is rejected for a level of significance of $\alpha = 0.1$.

In the post hoc Bonferroni–Dunn test, using the best method (B-CDT) as a reference, we found a significant difference, for

Table 3
Friedman's rank for each method.

Method	Rank
B-CDT	1
B-C4.5	2.333
RS-LNMC	2.666

$\alpha = 0.1$, in favor of B-CDT when compared to the RS-LNMC method.[4] No more significant differences are found by the Bonferroni–Dunn test.

Summarizing the results obtained, the Bagging scheme on DTs outperforms the best method presented in Nanni and Lumini (2009). This difference is statistically significant when the Bagging scheme is applied on credal decision trees.

4. Conclusions

This paper complements the work presented in Nanni and Lumini (2009) about ensembles of classifiers on data sets for bankruptcy prediction and credit. The Bagging scheme of decision trees has been correctly implemented and incorporated in the previous comparisons.

This paper presents the following: (i) A new procedure to build decision trees, called *Credal Decision Trees*, treats the imprecision in a different way than classic procedures; (ii) The Bagging scheme on this type of decision trees outperforms the results obtained using the best method previously presented for data sets related to bankruptcy prediction and credit scoring. This affirmation is supported by a set of statistical tests.

The new machine learning model described in the present work, *Bagging scheme of credal decision trees*, can be applied to data sets of real world applications other than bankruptcy prediction and credit scoring. Also new mathematical models, procedures and split criteria, as the ones of Abellán, Baker, Coolen, Crossman, and Masegosa (2013), Abellán (2013a) and Abellán (2013b) can be applied in a similar way.

Acknowledgments

This work has been supported by the Spanish "Consejería de Economía, Innovación y Ciencia de la Junta de Andalucía" under Project TIC-06016.

We are very grateful to the authors of Nanni and Lumini (2009), for providing us with the data sets they used in their work.

References

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

Abellán, J. (2006b). Application of uncertainty measures on credal sets on the Naive Bayes classifier. *International Journal of General Systems, 35*, 675–686.

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

Abellán, J. (2013a). Ensembles of decision trees based on imprecise probabilities and uncertainty measures. *Information Fusion, 14*(4), 423–430.

Abellán, J. (2013b). An application of non-parametric predictive inference on multi-class classification high-level-noise problems. *Expert Systems with Applications, 40*, 4585–4592.

Abellán, J., Baker, R. M., Coolen, F. P. A., Crossman, R., & Masegosa, A. (2013). Classification with decision trees from a nonparametric predictive inference perspective. *Computational Statistics and Data Analysis, 71*(14), 789–802.

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. (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. (2009a). A filter-wrapper method to select variables for the Naive Bayes classifier based on credal decision trees. *Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 17*(6), 833–854.

Abellán, J., & Masegosa, A. R. (2009b). An experimental study about simple decision trees for bagging ensemble on data sets with classification noise. In C. Sossai & G. Chemello (Eds.), *ECSQARU. LNCS* (Vol. 5590, pp. 446–456). Springer.

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

Abellán, J., & Masegosa, A. (2012). Bagging schemes on the presence of noise in classification. *Expert Systems with Applications, 39*(8), 6827–6837.

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

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

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

Beck, J. R., & Schultz, E. K. (1986). The use of ROC curves in test performance evaluation. *Archives of Pathology and Laboratory Medicine, 110*, 13–20.

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

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. *Wadsworth statistics, probability series*. Belmont.

Demsar, J. (2006). Statistical comparison of classifiers over multiple data sets. *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.

Fawcett, T. (2004). ROC graphs: Notes and practical considerations for researchers. Technical Report, Palo Alto, USA: HP Laboratories.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(8), 832–844.

Jaynes, E. T. (1982). On the rationale of maximum-entropy methods. *Proceeding of the IEEE, 70*(9), 939–952.

Klir, G. J. (2006). *Uncertainty and information, foundations of generalized information theory*. Hoboken, NJ.: John Wiley.

Muttil, N., & Chau, K. W. (2006). Neural network and genetic programming for modelling coastal algal blooms. *International Journal of Environment and Pollution, 28*(3–4), 223–238.

Nanni, L., & Lumini, A. (2009). An experimental comparison of ensemble of classifiers for bankruptcy prediction credit scoring. *Expert Systems with Applications, 36*(2 PART 2), 3028–3033.

Provost, F., & Domingos, P. (2003). Tree induction for probability-based ranking. *Machine Learning, 52*(3), 199–215.

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

Quinlan, J. R. (1993). Programs for machine learning. Morgan Kaufmann Series in Machine Learning. San Mateo, CA.

Taormina, R., Chau, K. W., & Sethi, R. (2012). Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. *Engineering Applications of Artificial Intelligence, 25*(8), 1670–1676.

Tsai, C. F., & Wu, J. W. (2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 34*(4), 2639–2649.

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

Witten, I. H., & Frank, E. (2005). *Data mining, practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Wu, C. L., Chau, K. W., & Li, Y. S. (2009). Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. *Water Resources Research, 45*, W08432. http://dx.doi.org/10.1029/2007WR006737.

[4] The critical difference is 1.60033 (see Demsar, 2006).