Bagging of Credal Decision Trees for Imprecise Classification

S. Moral-García[1], Carlos J. Mantas[1], Javier G. Castellano[1], María D. Benítez[2] and Joaquín Abellán[1]

[1] Department of Computer Science and Artificial Intelligence
University of Granada, Granada, Spain
{seramoral,cmantas,fjgc,jabellan}@decsai.ugr.es
[2] BANKIA
Central Office, Granada, Spain
mbenitez@bankia.com

Abstract. The Credal Decision Trees (CDT) have been adapted for Imprecise Classification (ICDT). However, no ensembles of imprecise classifiers have been proposed so far. The reason might be that it is not a trivial question to combine the predictions made by multiple imprecise classifier. In fact, if the combination method used is not appropriate, the ensemble method could even worse the performance of one single classifier. On the other hand, the Bagging scheme has shown to provide satisfactory results in precise classification, specially when it is used with CDTs, which are known to be very weak and unstable classifiers. For these reasons, in this research, it is proposed a new Bagging scheme with ICDTs. It is presented a new technique for combining predictions made by imprecise classifiers that tries to maximize the precision of the bagging classifier. If the procedure for such a combination is too conservative it is easy to obtain few information and worse the results of a single classifier. Our proposal considers only the states with the minimum level of non-dominance. An exhaustive experimentation carried out in this work has shown that the Bagging of ICDTs, with our proposed combination technique, performs clearly better than a single ICDT.

Keywords: Imprecise Classification, Credal Decision Trees, ensembles, Bagging, combination technique.

1 Introduction

Within Machine Learning field, supervised classification (Hand, 1997) is a crucial task that tries to predict, for an instance described via a set of *attributes* or *features*, the value of a variable under study, also known as *class variable*. A training set is used in order to learn the model that allows to make the predictions about the class. For a new instance that is wanted to be classified, its associated class value is predicted using the learned model. This prediction tends to consist of a single state. Nevertheless, in many cases, the information available does not

allow to point clearly to one class value. In these situations, it is more informative that the classifier returns a set of states of the class variable. Classifiers of this type are known as *imprecise classifiers*, which give us *imprecise predictions*.

When an imprecise classifier is utilized, a set of states of the class variable might be obtained. These states are those which there is no evidence to reject them, i.e, for each predicted class value, there is no another "better" state according to a certain criterion utilized. Usually, the set of predicted class values is known as the set of *non-dominated states* and the criterion employed to obtain it is called *dominance criterion*. Logically, the performance measure of an imprecise classifier must take into account if the real class value is among the non-dominated states and how precise is the set of predicted states, being the precision measured by its cardinality.

Models based on imprecise probabilities are more appropriate to be applied in the building process of an imprecise classifier than those based on the classical probability theory. Many mathematical theories developed in the literature are associated with the term *imprecise probabilities*. A detailed description of them can be found in (Klir, 2005). Examples of these models are *lower* and *upper probabilities, belief functions, Choquet capacities, probability intervals* etc.

There are few methods for imprecise classification in the literature. The first of them was the *Naive Credal Classifier* (NCC) (Zaffalon, 2002; Corani & Zaffalon, 2008). It combines the Naive assumption (all the attributes are independent given the class variable) and the Imprecise Dirichlet Model (IDM) to produce an imprecise classification.

The Credal Decision Tree algorithm (CDT) (Abellán & Moral, 2003), which is based on Decision Trees (DT) and uses the IDM and general uncertainty measures on closed and convex sets of probability distributions (also called credal sets) in the building process, has been adapted for Imprecise Classification in (Abellán & Masegosa, 2012b). The model is called Imprecise Credal Decision Tree (ICDT). Basically, ICDT builds the tree in a similar way that CDTs for precise classification and, in the leaf nodes, it utilizes a dominance criterion in order to predict the set of non-dominated states associated with that leaf node. In (Abellán & Masegosa, 2012b; Corani, Abellán, Masegosa, Moral, & Zaffalon, 2014), it is shown that the ICDT method is much more informative method than NCC.

On the other hand, the Bagging scheme (Breiman, 1996) is an ensemble method that has been shown to have good performance in precise classification. Essentially, for an instance, it combines the predictions made by distinct models built with different training sets in order to give a final prediction about the value of the class variable of the instance, in such a way that the diversity is increased. The Bagging method performs better when it is used with weak and unstable classifiers. Examples about this point can be found in (Abellán & Mantas, 2014; Abellán & Masegosa, 2009; Abellán & Castellano, 2017; Marqués, García, & Sánchez, 2012).

Furthermore, an important issue of DTs is that few variations in the training set can produce considerable differences in the model, which is known as *diversity*

(Tsymbal, Pechenizkiy, & Cunningham, 2005). Thus, the DTs are very suitable to be applied in Bagging schemes. More specifically, the Bagging method has been shown to provide good results with CDTs as base classifiers. Examples of this fact can be found in (Abellán & Mantas, 2014; Abellán & Castellano, 2017; Abellán & Masegosa, 2010, 2012a).

None of the Imprecise Classification algorithms proposed so far make ensemble of classifiers. The reason may be the following: Remark that the predictions made by imprecise classifiers consist of a set of states, so that it is not a trivial question to combine the predictions made by imprecise classifiers and no technique has been proposed so far for this purpose. If the predictions are not combined in a suitable way, it is really probable that the performance of the ensemble is not better than the obtained by a single classifier, because an excessive reduction of the information or the uncertainty can be produced.

Summarizing, the Bagging schemes has been shown to have a good performance in precise classification, specially when they are used with CDTs as base classifiers. In addition, there is no a procedure to combine the predictions made by several imprecise classifiers and, consequently, there are no ensemble methods for Imprecise Classification proposed so far. For these reasons, a new Bagging scheme using CDTs for imprecise classification is proposed in this research. The combination method that we suggest tries that the new Bagging method is as more precise as possible, though it implies a higher risk of making erroneous predictions. When a large set of imprecise informations are combined there is a risk of loss of information. We will see that this does not happen with our proposed combination technique.

An exhaustive experimental study is carried out in this work in which we try to compare the performance of the Bagging of CDTs for imprecise classification versus the ICDT algorithm. Recall that in (Abellán & Masegosa, 2012b) ICDT algorithm is shown to be more informative that NCC. In addition, there is no other ensemble method for Imprecise Classification in the literature. Both methods are applied to 34 different datasets, which have in common that all of them have at least 3 states of the class variables, in such a way that this study on imprecise classification makes more sense. Two known evaluation measures for imprecise classification are used in order to compare the performance of both algorithms. This experimentation shows that, similarly to what happens in precise classification, the Bagging of CDTs for Imprecise Classification obtains better results than the ICDT algorithm.

The rest of this paper is organised as follows: In Section 2, the necessary previous knowledge is exposed: The Bagging scheme, the probability intervals from the IDM, the dominance criteria on probability intervals from the IDM and the adaptation of Credal Decision Trees for Imprecise Classification. The Bagging scheme with Credal Decision Trees for imprecise classification is explained in Section 3. In Section 4 the evaluation metrics for imprecise classification sed in this work are exposed. The experimental study carried out in this work is detailed in Section 5. Finally, Section 6 is devoted to concluding remarks and future work.

2 Background

2.1 The Bagging scheme

Let us suppose that a training set of N instances is disposed. Let us denote C the class variable and $\{c_1, \cdots, c_k\}$ its possible states. The Bagging method (bootstrap aggregating) (Breiman, 1996) builds a set of m classifiers. For each one of them, it is obtained a bootstrapped replica of the original training set: N instances are randomly selected with replacement. Then, a model is learned using this selected bootstrapped replica. In order to build the classifiers, a basic learning algorithm is used. This method is often a DT, but many other classification algorithms can be also used in the Bagging scheme.

When a new instance is wanted to be classified, for each state of the class variable, it is counted the number of classifiers that predict that class value for the instance, which is called the number of votes. The state that has the highest number of votes is the one predicted for the instance.

The Bagging scheme is summarized in Figure 1.

Procedure **Bagging**(training set of N instances $\mathcal{D}$, learning algorithm $\mathcal{C}$, number of classifiers m)

1. From $i = 1$ to m
   2. Select with replacement a sample of N instances, $\mathcal{D}_i$, from $\mathcal{D}$.
   3. Build a classifier $\mathcal{C}_i$ using the algorithm $\mathcal{C}$ and $\mathcal{D}_i$ as training set

For classifying a new instance $\mathbf{x}$:

1. From $j = 1$ to k
   2. Let v_j be the number of classifiers that predict the class value c_j for $\mathbf{x}$ (the number of votes)
3. Let consider $l = \arg\max_{j=1,\cdots,k} v_j$
4. Return c_l

Fig. 1. Pseudo-code of Bagging scheme

The idea of the Bagging method is to obtain diversity by building several models with different training sets. This diversity is reached specially when the base classifier is weak and unstable, i.e, when it is sensitive to few variations in the training set. This fact happens with DTs (Tsymbal et al., 2005). For this reason, models based on DTs are specially suitable to be applied in the Bagging scheme.

2.2 Probability intervals from IDM

Let $\mathcal{D}$ be a dataset with N instances. Let us suppose that X is an attribute and $\{x_1, \cdots, x_t\}$ are its possible values.

The *Imprecise Dirichlet Model* (IDM) (Walley, 1996) is a particular case of the Probability intervals theory (De Campos, Huete, & Moral, 1994). The IDM estimates that the probability that the variable X takes its possible value x_i, $1 \leq i \leq t$ is within the interval:

$$I_i = \left\{ \left[\frac{n(x_i)}{N+s}, \frac{n(x_i)+s}{N+s} \right] \right\}, \forall i = 1, \cdots, t \tag{1}$$

where $n(x_i)$ is the number of instances in the dataset that verify $X = x_i$, $\forall i = 1, 2, \cdots, t$ and $s > 0$ a given hyperparameter of the model.

In (Abellán, 2006) it is proved that this set of probability intervals is reachable and that the IDM intervals can be also expressed by a belief function. This set of intervals gives rise to the following credal set (Abellán, 2006):

$$K^{\mathcal{D}}(X) = \left\{ p \mid \sum_{i=1}^{t} p(x_i) = 1, \frac{n(x_i)}{N+s} \leq p(x_i) \leq \frac{n(x_i)+s}{N+s} \quad \forall i = 1, \cdots, t \right\} \tag{2}$$

An important question is the selection of the s hyperparameter. It is easily observable that the intervals are wider if the s value is higher. The s hyperparameter determines how quickly the lower and upper probabilities converge as there is more data available. In (Walley, 1996) two values are proposed: $s = 1$ and $s = 2$, and it is recommended the value $s = 1$.

2.3 Dominance criteria on probability intervals from the IDM:

In Imprecise Classification, a *dominance criterion* is employed in order to determinate which class values are not defeated under probability terms by the rest. For this purpose, when probability intervals are used, as in this work, the bounds of the intervals can be utilized.

Let us suppose that c_i and c_j are two possible values of the class variable C. With the available information, the two following *dominance criteria* are very utilized:

1. Let suppose that $[l_i, u_i]$ and $[l_j, u_j]$ are, respectively, the probability intervals about the class values c_i and c_j. It is said that there is *stochastic dominance* of c_j on c_i iff $l_j \geq u_i$.
2. If we know that the probability of each state of C can be expressed via a non-empty credal set $\mathbf{P}$. We say that there is *credal dominance* of c_j on c_i iff $p(C = c_j) \geq p(C = c_i)$ for each probability distribution p that belongs to $\mathbf{P}$.

It is known that the credal dominance is more significant criterion that stochastic dominance (Zaffalon, 2002). Nevertheless, it is often more complicated to check it than stochastic dominance. According to the results proved in (Abellán, 2012), under the IDM, both dominance criteria are equivalent. Thus, with the IDM, if it is verified that there is stochastic dominance of one state on another, then it is known that the credal dominance of the first state on the second one is also satisfied. Therefore, with IDM, it is only required to consider the extreme values of the intervals in order to check the cases of credal dominance among the possible values of the class variable.

2.4 Credal Decision Tree for Imprecise Classification

The Credal Decision Tree algorithm, proposed in (Abellán & Moral, 2003), was adapted for Imprecise Classification in (Abellán & Masegosa, 2012b). It was called Imprecise Credal Decision Tree (ICDT).

As in classical DTs methods, in ICDT each node is associated with a feature and each branch corresponds to a possible value of that feature. When there are no more features to be entered in a node, or when entering a feature in that node does not give more information of the class variable according to a measure, a leaf or terminal node is reached. This terminal node indicates the expected value of the class variable. When a new instance is wanted to be classified, it is followed a path from the root to a terminal node using its attribute values and the tree structure. The set of predicted states of the class variable for the instance is the set of class values associated with that leaf node.

The most important point of the building process of a DT is the criterion used in order to select the feature to split in each node, which is known as the split criterion. In ICDT, the split criterion is the same as the one used in CDT [1]. Let $\mathcal{D}$ be a partition of the training set in a certain node. Let us suppose that C is the class variable and $\{c_1, \ldots, c_k\}$ are its possible values. Let X be a variable whose possible values are $\{x_1, \ldots, x_t\}$. Using the same notation than in Section 2.2, let us denote $K^{\mathcal{D}}(C)$ to the credal set associated with the class variable in the partition $\mathcal{D}$:

$$K^{\mathcal{D}}(C) = \left\{ p \mid \sum_{j=1}^{k} p(c_j) = 1, \frac{n(c_j)}{N+s} \leq p(c_j) \leq \frac{n(c_j)+s}{N+s} \quad \forall j = 1, 2, \cdots, k \right\} \tag{3}$$

The split criterion used in ICDT considers the maximum of the Shannon Entropy (Shannon, 1948) in this credal set:

$$H^*(K^{\mathcal{D}}(C)) = \max\left\{ H(p) \mid p \in K^{\mathcal{D}}(C) \right\}, \tag{4}$$

where H is the Shannon entropy (Shannon, 1948), defined as follows:

[1] The building process is the same for both algorithms.

$$H(p) = -\sum_{i=1}^{k} p(c_k) \log p(c_k) \tag{5}$$

The maximum of entropy in credal sets is a good uncertainty measure that verifies a couple of good properties (Klir, 2005).

The algorithm that obtains the probability distribution that maximizes H^* is detailed in (Mantas & Abellán, 2014).

Thus, the split criterion employed in the ICDT algorithm is the Imprecise Informatio Gain (IIG) (Abellán & Moral, 2003), defined by the following formula:

$$IIG(C, X) = H^*(K^{\mathcal{D}}(C)) - \sum_{i=1}^{t} P^{\mathcal{D}}(X = x_i) H^*(K^{\mathcal{D}}(C \mid X = x_i)) \tag{6}$$

being $H^*(K^{\mathcal{D}}(C \mid X = x_i))$ the credal set associated with C in the partition of $\mathcal{D}$ for which $X = x_i$, $\forall i = 1, \ldots, t$ and $P^{\mathcal{D}}(X = x_i)$ the probability distribution that reaches the maximum of entropy in $H^*(K^{\mathcal{D}}(X))$. It is estimated in the same way that the distribution that obtains the maximum of entropy in (4).

The main difference between the CDT algorithm and its adaptation for Imprecise Classification, ICDT, is the criterion used to classify an example once the instance has reached a leaf node following a path from the root node using the tree structure. Whereas CDT simply assigns to the instance the most probable state of the class variable in the corresponding terminal node (the most frequent in that leaf), ICDT assigns IDM intervals to each possible value of class variable according to that leaf node. Then, it applies a dominance criterion to that IDM intervals associated with that terminal node in order to select the set of non-dominated states of the instance.

The process to classify a new instance in ICDT algorithm is summarized in Figure 2.

As we have said previously, for the concrete case of IDM, stochastic and credal dominance are equivalent and the first criterion is far easier to check than the second one. Therefore, in this research we use the stochastic dominance.

3 Bagging of Credal Decision Trees for Imprecise Classification

In this research it is proposed a Bagging scheme for Imprecise Classification using the ICDT algorithm as base classifier. We call it Bagging of Imprecise Credal Decision Trees (Bagging-ICDT).

In order to build the base classifiers, the idea is similar to Bagging scheme for precise classification. For each one of them a boostraped replica of the original training set is selected. Then, using the boostraped sample and our base classification algorithm, ICDT, an Imprecise Classification model is learned.

The key point of the proposed Bagging scheme for Imprecise Classification is how to combine the predictions made for each one of the base classifiers.

Procedure **Classify_Instance_ICDT**(Built Tree $\mathcal{T}$, new instance $\mathbf{x}$)

1. Apply $\mathbf{x}$ in $\mathcal{T}$ to reach a leaf node
2. Consider the probability intervals in this terminal node for $\mathbf{x}$

$$\left\{\left[\frac{n'(c_i)}{N'+s}, \frac{n'(c_i)+s}{N'+s}\right], i = 1, \cdots, k\right\},$$

being $n'(c_i)$ the number of instances which verify that $C = c_i$ in the partition associated with that terminal node and N' the size of that partition (Clearly, $N' = \sum_{i=1}^{k} n'(c_i)$).
3. Apply a dominance criterion to the above intervals in order to get a set of non-dominated states for $\mathbf{x}$: $\{c_{i_1}, c_{i_2}, \cdots, c_{i_r}\}$, with $r \leq k$.

Fig. 2. Classification of a new instance in ICDT algorithm

Remark that in precise classification this is as simple as taking the majority vote. However, for Imprecise Classification it is not a trivial question, since in these cases the base classifiers do not return an unique value of the class variable, but they predict a set of non-dominated states.

In fact, there are multiple ways to combine the predictions, since one state might be predicted as dominated by some classifiers and as non dominated by others. The crucial issue is to determinate, taking into account the number of classifiers that predict that one state is dominated, the threshold to decide if the state is dominated or not for the final combination. Actually, this consists of a trade-off between *risk* and *information*. Here, the term *risk* is used to denote the probability of not including the real class value between the set of not-dominated states. The term *information* indicates how precise is the prediction, i.e, how many states are predicted as non-dominated. Logically, more information implies more risk. We consider that our proposal is closer to the risk, because it consider the states with the minimum level of non-dominance.

If all the states which have been predicted as non-dominated by at least one classifier are finally predicted as non-dominated, the probability of making an erroneous prediction is minimum. However, in these situations, the set of predicted states would be composed by almost all the possible values of the class variable, so that the predictions are hardly informative and the Bagging classifier would not be very useful. For this reason, our strategy consists of the opposite extreme: we want that the Bagging scheme be as more informative as possible, even thought this implies more risk of erroneous prediction.

Therefore, in our proposed algorithm, when a new instance is wanted to be classified, for each possible value of the class variable, it is counted the number of classifiers which predict that state as dominated, which we call the number of *votes against*. The states which has the minimum number of *votes against* are those which are finally predicted as non-dominated by our Bagging scheme.

The Bagging-ICDT algorithm is summarized in Figure 3, where C is the class variable, being $\{c_1, c_2, \cdots, c_k\}$ its set of possible values.

Procedure **Bagging-ICDT**(training set of N instances $\mathcal{D}$, number of classifiers m)

1. From $i = 1$ to m
   2. Select with replacement a sample of N instances, $\mathcal{D}_i$, from $\mathcal{D}$.
   3. Build a classifier $\mathcal{C}_i$ using the ICDT algorithm and $\mathcal{D}_i$ as training set

For classifying a new instance $\mathbf{x}$

1. From $j = 1$ to k
   2. Let va_j be the number of classifiers that predict the state c_j as dominated
3. Let consider $min\_against = \min_{j=1,\cdots,k} va_j$
4. The final predicted non-dominated set is $\{c_r \mid va_r = min\_against, 1 \leq r \leq k\}$

Fig. 3. Bagging scheme with ICDT

In summary, with our proposed method, Bagging-ICDT, it is tried to increase the diversity considering distinct ICDTs built with different training set. To classify new instances, the predictions made by the base classifiers are combined in such a way that the Bagging is as more informative as possible.

4 Evaluation metrics for Imprecise Classification

As we have said previously, an evaluation metric for Imprecise Classification should take into account two issues. The first of them is the accuracy of the imprecise classifier, i.e, the average number of times in which if the real class value of one instance is between the set of non-dominated states for that example. The second point is the precision of the imprecise classifier, which is measured via the average number of non-dominated states.

The following evaluation metrics only focus on one of the points described above:

- **Determinacy:** It measures the proportion of instances classified precisely, i.e, the proportion of instances for which the classifier returns only one state.
- **Single Accuracy:** It is the accuracy between the instances for which there is only one non-dominated state.
- **Set Accuracy:** It indicates, between the instances for which the classifier predicts more than one class value, the proportion of them for which the real value of the class value is one of the non-dominated states.

– **Indeterminacy Size:** It is the average number of non-dominated states.

Obviously, none of these metrics is useful in order to evaluate the whole performance of an imprecise classifier.

The *Discounted Accuracy measure* (DACC), proposed in (Corani & Zaffalon, 2009), is a metric which tries to provide a global evaluation of an imprecise classifier. It is defined as follows:

$$DACC = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} \frac{(correct)_i}{|U_i|} \tag{7}$$

where N_{Test} is the cardinality of the test set; U_i is the set of non-dominated states for the i-th instance; $|U_i|$ its cardinality; $(correct)_i$ is equal to 1 if the real class value belongs to U_i and 0 otherwise.

As we can see, this evaluation measure does not penalize the errors in a strict sense, since it does not add any value when there is an error. It is an accuracy metric. DACC does not sum any value for the incorrect predictions and, for the right ones, the added value is penalized by the number of predicted class values.

Clearly, a higher value of DACC implies a better performance. The highest value of DACC is equal to 1, and it is reached when all the predictions are correct and precise, i.e, when for all the instances there is only one non-dominated state and it coincides with its real class value. If the classifier always returns as non-dominated states all the possible values class values, the value of DACC is $\frac{1}{k}$. In our opinion this value should be lower, since in these situations the classifier is not informative.

A new evaluation measure for Imprecise Classification, MIC, was proposed in (Abellán & Masegosa, 2012b). That metric supposes that the errors have different degrees of importance in some situations. For the sake of simplicity, in this research it is supposed that the importance of the errors is the same for all of them.

If the prediction for an instance is right, MIC adds a value which depends on $\frac{|U_i|}{k}$. When an instance is incorrectly classified, since it is supposed the same degree of importance for all the errors in this work, MIC adds a constant value, dependent on k. More specifically, MIC is defined as follows:

$$MIC = \frac{1}{N_{Test}} \left(\sum_{i:Success} \log \frac{|U_i|}{k} + \frac{1}{k-1} \sum_{i:Error} \log k \right) \tag{8}$$

It is obvious that the higher is the value of MIC, the better is the performance. We can observe that the optimal value of MIC, which is reached when all the predictions are precise and correct, is equal to $-\log \frac{1}{k} = \log k$. Moreover, when it is verified that $|U_i| = k$, $\forall i = 1, \cdots, N_{Test}$, i.e, when the imprecise classifier always predicts all possible class values as non-dominated states for an instance, the value of MIC is equal to 0. It is intuitively more correct, since in these cases the classifier is not informative.

5 Experimentation

In this Section we present the experimental study in which it is shown that the Bagging of Imprecise Credal Decision Trees (Bagging-ICDT), with our proposed combination technique, performs clearly better than the Credal Decision Trees for Imprecise Classification (ICDT). [2] When a lot of imprecise informations are combined there is a risk of loss of information. [3] We will see that this does not happen with our proposed combination technique.

5.1 Experimental settings

For this experimentation, we have used the implementation given in Weka software (Witten & Frank, 2005) for ICDT. The structures given in Weka for the Bagging scheme have been also utilized in order to add all the necessary methods for implementing Bagging-ICDT. For Bagging-ICDT we have used 100 trees. It is an appropriate number of classifiers for Bagging (Breiman, 1996). The rest of the parameters used for both algorithms have been those given by the default in Weka.

ICDT and Bagging-ICDT have been applied to a set of 34 known datasets. They can be obtained from the *UCI Machine Learning repository* (Lichman, 2013). These datasets are different in terms of size of the set, number of continuous and discrete features, number of values per attribute, number of class values, etc. The datasets have been selected in such a way that the class variable has at least three possible values, as in (Abellán & Masegosa, 2012b). The reason is that with only two class values or all that states of the class variable are obtained or just one. Table 1 illustrates the most important characteristics of each dataset.

Consistently with (Abellán & Masegosa, 2012b), we have preprocessed the datasets as follows: Missing values have been replaced by mean values for continuous variables and by modal values for discrete attributes. Then, continuous variables have been discretized via Fayyad and Irani's discretization procedure (Fayyad & Irani, 1993). For each preprocessed dataset a 10 times 10-fold cross-validation procedure has been repeated.

In order to evaluate the performance of the algorithms considered in this experimentation, ICDT and Bagging-ICDT, we have considered the two evaluation metrics for Imprecise Classification employed in (Abellán & Masegosa, 2012b): MIC and DACC. Both of these measures have been detailed in Section 4.

For statistical comparisons between both algorithms, consistently with the recommendations given in (Demšar, 2006), the two following tests with a level of significance of $\alpha = 0.05$ have been used:

[2] We do not use the NCC in this experimentation because, as we have said in the Introduction Section, in previous works, it has been shown to be a less informative method than ICDT.

[3] If the union of non-dominated states in each classifier were considered, it would be produced a loss of information which would give rise to poor results, which we have not included in this work.

Table 1. dataset description. Column "N" is the number of instances in the datasets, column "Feat" is the number of features or attribute variables, column "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 dataset.

dataset	N	Feat	Num	Nom	k	Range
anneal	898	38	6	32	6	2-10
arrhythmia	452	279	206	73	16	2
audiology	226	69	0	69	24	2-6
autos	205	25	15	10	7	2-22
balance-scale	625	4	4	0	3	-
bridges-version1	107	11	3	8	6	2-54
bridges-version2	107	11	0	11	6	2-54
car	1728	6	0	6	4	3-4
cmc	1473	9	2	7	3	2-4
dermatology	366	34	1	33	6	2-4
ecoli	366	7	7	0	7	-
flags	194	30	2	28	8	2-13
hypothyroid	3772	30	7	23	4	2-4
iris	150	4	4	0	3	-
letter	20000	16	16	0	26	-
lymphography	146	18	3	15	4	2-8
mfeat-pixel	2000	240	0	240	10	4-6
nursery	12960	8	0	8	4	2-4
optdigits	5620	64	64	0	10	-
page-blocks	5473	10	10	0	5	-
pendigits	10992	16	16	0	10	-
postop-patient-data	90	9	0	9	3	2-4
primary-tumor	339	17	0	17	21	2-3
segment	2310	19	16	0	7	-
soybean	683	35	0	35	19	2-7
spectrometer	531	101	100	1	48	4
splice	3190	60	0	60	3	4-6
sponge	76	44	0	44	3	2-9
tae	151	5	3	2	3	2
vehicle	946	18	18	0	4	-
vowel	990	11	10	1	11	2
waveform	5000	40	40	0	3	-
wine	178	13	13	0	3	-
zoo	101	16	1	16	7	2

- **Corrected Paired t-test**: This test is used to compare two algorithms in a single dataset. It consists of a corrected version of the Paired t-test implemented in Weka. Essentially, this test verifies if one algorithm is better than the other one on average, across all training and test sets extracted from a 10 times 10-fold cross-validation procedure on a original dataset.
- **Wilcoxon test** (Wilcoxon, 1945): We use this test to compare two algorithms in multiple datasets. It ranks the differences betweeen the performance of two algorithms for each dataset, without taking into account signs. Then, it compares the ranks for the positive and negative differences.

5.2 Results and discussion

Tables 2 and 3 show, respectively, the results obtained by each algorithm for each dataset in DACC and MIC measures. For each dataset, the best result is marked in bold. Furthermore, for each dataset, it is illustrated which algorithm is better according to Corrected paired t-test (in case that the differences are significative).

A summary of the results for both DACC and MIC evaluation metrics can be seen in Table 4. In concrete, for MIC and DACC, it shows the average value, the result of Wilcoxon test and the number of datasets where ICDT performs significantly better than Bagging-ICDT according to Corrected Paired t-test and vice-versa.

As it can be easily observed, for both DACC and MIC, in almost all datasets Bagging-ICDT outperforms ICDT. In fact, for DACC, ICDT only performs better than Bagging-ICDT in one dataset and both algorithms obtain the same result is two datasets. In the rest of the datasets the performance is better for ICDT according to this measure. Similarly, for MIC, Bagging-ICDT obtains a better result than ICDT for all datasets except four. In three of them ICDT outperforms Bagging-ICDT and in the other one both algorithms obtain exactly the same result. As can be seen in Table 4, the average values of DACC and MIC are much higher for Bagging-ICDT and, according to Wilcoxon test, Bagging-ICDT outperforms ICDT significantly in both of these metrics. In addition, according to corrected paired test, the number of datasets where Bagging-ICDT obtains significantly better results than ICDT is 16 for DACC and 14 for MIC. Nevertheless, for none of two metrics ICDT performs significantly better than Bagging-ICDT in any dataset.

Therefore, it can be concluded that Bagging-ICDT performs clearly better than ICDT, being the differences really considerable.

For a deeper analysis, we show in Table 5 the average results of Determinacy, Single Accuracy, Set Accuracy and Indeterminacy size obtained by ICDT and Bagging-ICDT. As in previous tables, the best results are marked in bold.

According to these results, the percentage of instances for which a single class state is returned is higher for Bagging-ICDT (Determinacy). Between these instances determinantly classified, the accuracy is similar for both algorithms (Single Accuracy). Besides, the indeterminacy size is considerably lower for Bagging-ICDT than for ICDT. It implies that the predictions of Bagging-ICDT are more

precise than the ones made by ICDT, although with Bagging-ICDT there are more erroneous predictions, due to the results obtained in Set Accuracy. Hence, it can be said that Bagging-ICDT classifies the instances in a much more precise way than ICDT, even though with Bagging-ICDT the error rate is a little bit higher.

In summary, Bagging-ICDT is a far more precise classifier than ICDT, although the second algorithm is a little bit better with the Set Accuracy measure. The results obtained in the Imprecise Classification metrics proposed so far in the literature, DACC and MIC, allow to conclude that the Bagging-ICDT clearly outperforms ICDT. Therefore, it can be said that our proposed technique of combining predictions given by several imprecise classifiers, which tries to maximize *information* although it implies more *risk*, as explained in Section 3, is quite appropriate in the sense that it improves the results obtained with a single ICDT.

6 Conclusions and Future Work

It this research it has been proposed the first ensemble method for Imprecise Classification. We have taken into account that the Bagging scheme has been shown to provide pretty good results for precise classification, specially when it is used with Credal Decision Trees (CDT), which are known to be diverse and unstable classifiers. Hence, the proposed ensemble scheme for Imprecise Classification consists of a Bagging algorithm using the adaptation of CDTs for Imprecise Classification (ICDT) as base classifier.

In order to combine the predictions made by multiple classifiers, it has been proposed a new technique which tries that the Bagging imprecise classifier is as more precise as possible. Our new technique considers only the states with the lowest possible level of non-dominance, i.e. it is not too conservative. To reduce that number of states could produce an unnecessary excessive risk.

An exhaustive experimentation has been carried out in this work, comparing the ICDT algorithm with our Bagging scheme for ICDTs (Bagging-ICDT) with the proposed combination technique. This experimental analysis has shown that the Bagging-ICDT with our proposed combination technique, which tries to minimize the loss of information assuming risk, clearly outperforms the ICDT algorithm. As it was expected, even though the error rate is a little bit higher for Bagging-ICDT than for ICDT, the first algorithm is much more precise than the second one.

As future work, other ensemble schemes that have been shown to have good performance in precise classification, can be also used in Imprecise Classification, such as Boosting (Freund & Schapire, 1996) or Random Forest (Breiman, 2001). As said previously, in precise classification, the Decision Trees are very suitable to be applied in ensemble schemes. Thus, the ICDT algorithm would be very appropriate to be employed as base classifier in other future ensembles for Imprecise Classification.

On the other hand, the proposed combination technique has been shown to have satisfactory behaviour. Nevertheless, it is not clear that this combination method is optimal. Therefore, for future research, it would be interesting to study other techniques for combining the predictions made for Imprecise Classifiers.

Acknowledgments

This work has been supported by the Spanish "Ministerio de Economía y Competitividad" and by "Fondo Europeo de Desarrollo Regional" (FEDER) under Project TEC2015-69496-R.

References

Abellán, J. (2006). Uncertainty measures on probability intervals from the imprecise dirichlet model. *International Journal of General Systems*, *35*(5), 509-528. doi: 10.1080/03081070600687643

Abellán, J. (2012). Equivalence relations among dominance concepts on probability intervals and general credal sets. *International Journal of General Systems*, *41*(2), 109-122. doi: 10.1080/03081079.2011.607449

Abellán, J., & Castellano, J. G. (2017). A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications*, *73*, 1 - 10. doi: 10.1016/j.eswa.2016.12.020

Abellán, J., & Mantas, C. J. (2014). Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, *41*(8), 3825 - 3830. doi: 10.1016/j.eswa.2013.12.003

Abellán, J., & Masegosa, A. (2009). An experimental study about simple decision trees for bagging ensemble on datasets with classification noise. In *Symbolic and quantitative approaches to reasoning with uncertainty* (Vol. 5590, p. 446-456). Springer. doi: 10.1007/978-3-642-02906-6_39

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. R. (2012a). Bagging schemes on the presence of class noise in classification. *Expert Systems with Applications*, *39*(8), 6827 - 6837. doi: 10.1016/j.eswa.2012.01.013

Abellán, J., & Masegosa, A. R. (2012b). Imprecise classification with credal decision trees. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *20*(05), 763-787. doi: 10.1142/S0218488512500353

Abellán, J., & Moral, S. (2003). Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, *18*(12), 1215–1225. doi: 10.1002/int.10143

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. doi: 10.1023/A:1018054314350

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. doi: 10.1023/A:1010933404324

Corani, G., Abellán, J., Masegosa, A., Moral, S., & Zaffalon, M. (2014). Classification. In *Introduction to imprecise probabilities* (p. 230-257). John Wiley & Sons, Ltd. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118763117.ch10 doi: 10.1002/9781118763117.ch10

Corani, G., & Zaffalon, M. (2008). Learning reliable classifiers from small or incomplete data sets: the naive credal classifier 2. *JOURNAL OF MACHINE LEARNING RESEARCH*, *9*, 581–621.

Corani, G., & Zaffalon, M. (2009). Lazy naive credal classifier. In *Proceedings of the 1st acm sigkdd workshop on knowledge discovery from uncertain data* (pp. 30–37). New York, NY, USA: ACM. doi: 10.1145/1610555.1610560

De Campos, L. M., Huete, J. F., & Moral, S. (1994). Probability intervals: a tool for uncertain reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *02*(02), 167-196. doi: 10.1142/S0218488594000146

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Fayyad, U., & Irani, K. (1993). Multi-valued interval discretization of continuous-valued attributes for classification learning. In *Proceeding of the 13th international joint conference on artificial inteligence* (p. 1022-1027). Morgan Kaufmann.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In L. Saitta (Ed.), *Proceedings of the thirteenth international conference on machine learning (icml 1996)* (p. 148-156). Morgan Kaufmann.

Hand, D. J. (1997). *Construction and assessment of classification rules*. John Wiley and Sons, New York.

Klir, G. J. (2005). *Uncertainty and information: Foundations of generalized information theory*. John Wiley And Sons, Inc. doi: 10.1002/0471755575

Lichman, M. (2013). *UCI machine learning repository*. Retrieved from http://archive.ics.uci.edu/ml

Mantas, C. J., & Abellán, J. (2014). Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, *41*(10), 4625 - 4637. doi: 10.1016/j.eswa.2014.01.017

Marqués, A., García, V., & Sánchez, J. S. (2012). Exploring the behaviour of base classifiers in credit scoring ensembles. *Expert Systems with Applications*, *39*(11), 10244–10250. doi: 10.1016/j.eswa.2012.02.092

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423. doi: 10.1002/j.1538-7305.1948.tb01338.x

Tsymbal, A., Pechenizkiy, M., & Cunningham, P. (2005). Diversity in search strategies for ensemble feature selection. *Information Fusion*, *6*(1), 83 - 98. doi: 10.1016/j.inffus.2004.04.003

Walley, P. (1996). Inferences from multinomial data; learning about a bag of marbles (with discussion). *Journal of the Royal Statistical Society. Series*

B (Methodological), *58*(1), 3-57. doi: 10.2307/2346164

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, *1*(6), 80–83. doi: 10.2307/3001968

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (Second ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Zaffalon, M. (2002). The naive credal classifier. *Journal of Statistical Planning and Inference*, *105*(1), 5 - 21. (Imprecise Probability Models and their Applications) doi: https://doi.org/10.1016/S0378-3758(01)00201-4

Table 2. Complete results obtained for DACC measure

Dataset	ICDT	Bagging-ICDT	
anneal	0.9957	**0.9967**	
arrhythmia	0.6625	**0.7150**	∘
audiology	0.7887	**0.8232**	
autos	0.7817	**0.8278**	∘
balance-scale	0.6961	**0.6977**	
bridges-version1	0.6375	**0.6503**	
bridges-version2	0.5729	**0.6199**	
car	0.9168	**0.9299**	∘
cmc	0.4884	**0.4931**	
dermatology	0.9405	**0.9500**	
ecoli	0.7993	**0.8054**	
flags	0.5554	**0.6034**	∘
hypothyroid	0.9935	0.9935	
iris	0.9337	**0.9390**	
letter	0.7714	**0.8277**	∘
lymphography	0.7275	**0.7591**	
mfeat-pixel	0.7702	**0.8837**	∘
nursery	0.9628	**0.9654**	∘
optdigits	0.7716	**0.8647**	∘
page-blocks	0.9619	**0.9663**	∘
pendigits	0.8812	**0.9175**	∘
postoperative-patient-data	**0.7104**	0.7100	
primary-tumor	0.3815	**0.4239**	∘
segment	0.9406	**0.9502**	∘
soybean	0.9178	**0.9276**	
spectrometer	0.4430	**0.5127**	∘
splice	0.9270	**0.9447**	∘
sponge	0.9293	**0.9475**	
tae	0.4678	0.4678	
vehicle	0.6899	**0.7025**	
vowel	0.7635	**0.7953**	∘
waveform	0.7371	**0.7777**	∘
wine	0.9194	**0.9290**	
zoo	0.9592	**0.9612**	
Average	0.7763	**0.8023**	

∘, • statistically significant improvement or degradation

Table 3. Complete results obtained for MIC measure

Dataset	ICDT	Bagging-ICDT	
anneal	1.7825	**1.7847**	
arrhythmia	1.7861	**1.9316**	○
audiology	2.5156	**2.5936**	
autos	1.4535	**1.5553**	○
balance-scale	**0.6033**	0.6006	
bridges-version1	1.0247	**1.0446**	
bridges-version2	0.8755	**0.9767**	
car	1.2330	**1.2568**	○
cmc	0.2599	**0.2636**	
dermatology	1.6637	**1.6844**	
ecoli	1.6128	**1.6182**	
flags	1.0322	**1.1398**	
hypothyroid	1.3744	1.3744	
iris	0.9911	**0.9982**	
letter	2.5135	**2.6771**	○
lymphography	0.8857	**0.9417**	
mfeat-pixel	1.7194	**2.0066**	○
nursery	1.5350	**1.5398**	○
optdigits	1.7275	**1.9579**	○
page-blocks	1.5332	**1.5418**	○
pendigits	2.0042	**2.0925**	○
postoperative-patient-data	**0.6213**	0.6207	
primary-tumor	1.1476	**1.2278**	
segment	1.8119	**1.8331**	○
soybean	2.7004	**2.7203**	
spectrometer	1.7353	**1.9527**	○
splice	0.9784	**1.0077**	○
sponge	0.9822	**1.0121**	
tae	**0.2218**	0.2216	
vehicle	0.8171	**0.8372**	
vowel	1.7889	**1.8594**	○
waveform	0.6656	**0.7325**	○
wine	0.9658	**0.9817**	
zoo	1.8532	**1.8578**	
Average	1.3652	**1.4248**	

○, • statistically significant improvement or degradation

Table 4. Summary of the results obtained for DACC and MIC measures. In the "Wilcoxon test" rows, if one classifier is significantly better than the other one it is expresed by "*". The row "Paired t-test" indicates the number of datasets where one the algorithm in the column is better than the other one under the corrected-paired t-test

		ICDT	Bagging-ICDT
DACC:	Average	0.7763	0.8023
	Wilcoxon t-test		*
	Paired t-test	0	16
MIC:	Average	1.3652	1.4248
	Wilcoxon t-test		*
	Paired t-test	0	14

Table 5. Average results obtained for basic metrics by each algorithm. Best scores are marked in bold.

Algorithm	Determinacy	Single Accuracy	Set Accuracy	Indeterminacy size
ICDT	0.9477	0.8023	**0.8877**	5.2290
Bagging-ICDT	**0.9965**	**0.8037**	0.7792	**2.7013**