Cautious classification with nested dichotomies and imprecise probabilities

Gen Yang, Sébastien Destercke, Marie-Hélène Masson

Noname manuscript No.
(will be inserted by the editor)

Cautious Classification with Nested Dichotomies and Imprecise Probabilities

Gen Yang · Sébastien Destercke · Marie-Hélène Masson

the date of receipt and acceptance should be inserted later

Abstract In some applications of machine learning and information retrieval (*e.g.*, medical diagnosis, image recognition, pre-classification...), it can be preferable to provide less informative but more reliable predictions. This can be done by making partial predictions in the form of class subsets when the available information is insufficient to provide a reliable unique class. Imprecise probabilistic approaches offer nice tools to learn models from which such cautious predictions can be produced. However, the learning and inference processes of such models are computationally harder than their precise counterparts. In this paper, we introduce and study a particular binary decomposition strategy, nested dichotomies, that offer computational advantages in both the learning (due to the binarization process) and the inference (due to the decomposition strategy) processes. We show with experiments that these computational advantages do not lower the performances of the classifiers, and can even improve them when the class space has some structure.

Keywords: multi-class classification, binary decomposition, imprecise probabilities, indeterminate prediction, ordinal regression

1 Introduction

The classification task consists in allocating new observations, described by a set of features, to one predefined category, or class, from a set of possible ones. The usual task of supervised machine learning algorithms is to learn, from a training set of data containing observations associated with a known category, a mapping (or the classifier) that will provide the best recognition rate on new data. However, classification error frequently occurs when multiple classes have high and similar probabilities of occurrence (uncertainty due to ambiguity), or when training data are in insufficient quantity (uncertainty due to a lack of information).

Both cases suggest that it is possible to increase classifiers reliability by allowing their outputs to better reflect these uncertain situations. *Indeterminate* classifiers [del Coz and Bahamonde, 2009; Corani et al., 2012], which are able to predict more than one class in case of high uncertainty, have been introduced for this purpose. For example, in a problem of obstacle recognition for smart vehicles, a classifier could state that there is an obstacle on the road, without being able to decide if it is a pedestrian or a bicycle.

Address(es) of author(s) should be given

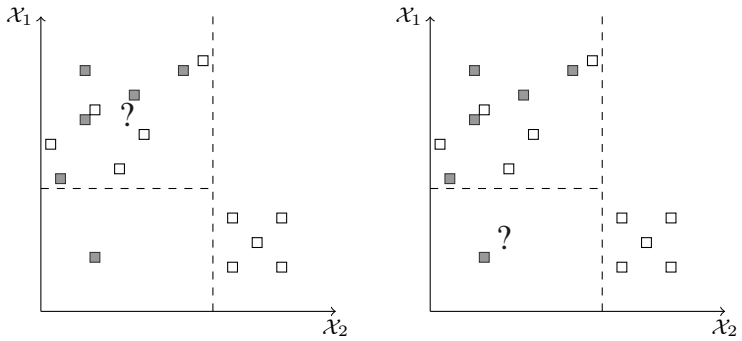


Fig. 1 Illustration of ambiguous vs uninformative situation

The idea of indeterminate prediction is close to the idea of classification with *reject option* [Chow, 1970]. In this latter approach, the classification of a new observation may simply not be performed (*i.e.*, rejected) when a classification error is likely to occur. The decision concerning the rejected observation is then left to a more specialized classifier or to a human expert. Rejection can therefore be seen as a very extreme indeterminate classifier, where only determinate and completely uninformative predictions (corresponding to the set of all classes) are allowed.

Different ways have been proposed to make indeterminate predictions other than completely uninformative ones. A first approach, directly inspired from the reject option, is to integrate costs of indeterminacy in the decision making [del Coz and Bahamonde, 2009]. One drawback of this approach is that it does not really differentiate between rejection due to ambiguity (almost uniform probabilities of classes estimated from lots of data) and rejection due to lack of information (probability issued from little and/or imprecise data). In fact, it may well produce determinate predictions even when having little information, provided that the inferred (precise) probability is not uniform. Also, how to mix misclassification costs (*e.g.*, of predicting no obstacle when there is a human) with indeterminacy costs (*e.g.*, costs of partial predictions like "human or bicycle") remains a quite tricky question. A second approach is to consider imprecise probability estimates rather than precise ones. In this case, the lack of information is represented by probability intervals, or by convex probability sets, the size of the intervals or sets reflecting the amount of available information (the smaller the set, the more the information). Several extensions of classical classifiers have been proposed in this framework, like the Credal C4.5 algorithm [Mantas and Abellan, 2014], the Naive Credal Classifier [Zaffalon, 2002] that extends the Naive Bayes Classifier, or the Credal Model Averaging [Corani and Zaffalon, 2008] that extends Bayesian Model Averaging. In comparison with indeterminacy through costs, imprecise probabilistic models may well differentiate ambiguity (narrow intervals with close highest probabilities) from lack of information (wide intervals). Figure 1 illustrates the two situations that result in very different models. However, imprecise probabilistic approaches may be computationally costly, either to learn or to take a decision from, while cost-based methods are most of the time designed to remain tractable.

In this paper, we explore the extension of nested dichotomies [Fox, 1997] to the imprecise probability framework. Nested dichotomies are binary decomposition methods that

transform a multiclass problem into a set of two-class problems deemed easier to solve than the original one [Dietterich and Bakiri, 1995; Allwein et al., 2000]. They are so-called meta-classifiers, as each two class problem can be solved by any classifier. In light of the previous remarks, this approach allows to limit the computational burden of learning and inferring from imprecise probabilistic model, making them more practical, and this whether misclassification costs exist or not.

The rest of this paper is organized as follows. We first establish the theoretical layout related to our approach in Section 2. This will allow us to detail how to adapt the nested dichotomies to the framework of imprecise probabilities, along with the expected advantages and drawbacks in Section 3. Experiments in Section 4 will show how the approach performs and compares to other approaches.

2 Classification, imprecision and binarisation

This section introduces the notations and tools used in this paper, in particular the basics of imprecise probabilities as well as of binarization approaches.

2.1 Classical classification problems

In a standard classification problem, the goal is to assign a class $\omega \in \Omega$ to an observation $\mathbf{x}$ from the input feature space $\mathbf{X} = X_1 \times \cdots \times X_m$. We also assume that a misclassification cost is specified: for all potential predictions $\hat{y} \in \Omega$, there is a cost function $c_{\hat{y}} : \Omega \to \mathbb{R}$ such that $c_{\hat{y}}(\omega)$ is the cost of predicting $\hat{y} \in \Omega$ when $\omega \in \Omega$ is the true class. The cost functions over all predictions $\hat{y}$ and all classes ω can be summarized into a $(|\Omega| \times |\Omega|)$ misclassification cost matrix (we denote by $|A|$ the cardinal of the set A).

Example 1 We consider as a running example the problem of obstacle recognition where a smart vehicle needs to recognize in a situation $\mathbf{x}$ whether it faces a *human* (h), a *bicycle* (b) or $nothing$ (n) (*i.e.*, $\Omega = \{h, b, n\}$).

In reality, as both human and bicycle are obstacles to be avoided, the confusion between h and b has little impact (but not null, as their movement patterns are different). However, predicting h or b when there is $nothing$ becomes more costly (the vehicle makes an unnecessary manoeuvre). Finally, predicting n when there is an obstacle h or b is a big mistake that could cause an accident. This kind of information can easily be expressed using generic cost functions. Table 1 provides cost functions modelling this information, as well as their difference (to be used in decision making).

Table 1 Example of misclassification costs for the obstacle recognition example

	truth		
$c_{\hat{y}}(\omega)$	$\omega = h$	$\omega = b$	$\omega = n$
c_h	0	1	2
c_b	1	0	2
c_n	4	4	0
$c_n - c_h$	4	3	-2
$c_n - c_b$	3	4	-2
$c_b - c_h$	1	-1	0

In a probabilistic framework, a conditional probability function $p(\cdot|\mathbf{x}) : \Omega \to [0, 1]$ of the classes given $\mathbf{x}$ is first learned or estimated, from which predictions are then made. For simplification purpose, we will note $p(\omega|\mathbf{x})$ by $p(\omega)$ when there is no risk of confusion. Once these probabilities are learned, a decision regarding a new observation can be made by comparing expected costs $\mathbb{E}[c_{\hat{y}}]$ for all possible outcomes $\hat{y}$:

$$\mathbb{E}[c_{\hat{y}}] = \sum_{\omega \in \Omega} p(\omega) c_{\hat{y}}(\omega). \tag{1}$$

Making prediction is then done by establishing a preference order $\succ$ over the expected costs of possible predictions to find the least costly one. Consider two predictions $\hat{y}_1$ and $\hat{y}_2$, we say that $\hat{y}_1$ is preferred to $\hat{y}_2$ (noted $\hat{y}_1 \succ \hat{y}_2$), if the expected cost of predicting $\hat{y}_1$ is less than the one of $\hat{y}_2$:

$$\hat{y}_1 \succ \hat{y}_2 \Leftrightarrow \mathbb{E}[c_{\hat{y}_1}] < \mathbb{E}[c_{\hat{y}_2}]. \tag{2}$$

Since $\mathbb{E}$ is linear, $\hat{y}_1 \succ \hat{y}_2$ is also equivalent to:

$$\hat{y}_1 \succ \hat{y}_2 \Leftrightarrow \mathbb{E}[c_{\hat{y}_2} - c_{\hat{y}_1}] > 0. \tag{3}$$

Eq. (3) can be interpreted as follows: $\hat{y}_1$ is preferred to $\hat{y}_2$ when exchanging $\hat{y}_1$ for $\hat{y}_2$ as prediction is costly (*i.e.*, has a positive expected cost). The selected class $\hat{y}^*$ is the maximal element of the ordering $\succ$:

$$\hat{y}^* = \arg\min_{\hat{y} \in \Omega} \mathbb{E}(c_{\hat{y}}). \tag{4}$$

The most common costs used in classification problems are the unitary costs, or 0-1 costs, defined by $c_{\hat{y}}(\omega) = \mathbb{1}_{\hat{y} \neq \omega}$ where $\mathbb{1}_A$ is the indicator function of event A ($= 1$ if A happens, 0 otherwise). In this latter case, Eq. (3) is equivalent to compare the probability estimates $p(\hat{y}_1), p(\hat{y}_2)$ ($\hat{y}_1 \succ \hat{y}_2$ if $p(\hat{y}_1) > p(\hat{y}_2)$), and the predicted class $\hat{y}^* = \arg\max_{\hat{y} \in \Omega} p(\hat{y})$ is simply the most probable one,

Note that the misclassification costs are only used here in the decision-making step, and not in the model learning process [Elkan, 2001; Masnadi-Shirazi and Vasconcelos, 2010], since we use probabilistic based models. This means that the estimation of the "objective" knowledge we have about the classes (i.e., the probabilities estimates) is separated from the subjectivity of decision-makers involved in the predictions (decision based on our risk aversion about car accidents). Yet, getting a reliable estimate of $p(\hat{y})$ can be challenging, for at least two reasons:

1. the data from which $p(\hat{y})$ must be estimated are scarce or of poor quality, in which case our estimate may be far from the true distribution;
2. estimating directly a distribution over the whole space Ω may be difficult, especially when $|\Omega|$ is high.

The first problem can be addressed by using imprecise probabilities (Section 2.2), while the second can be addressed by decomposing the initial problem into several, easier binary learning problems (Section 2.3).

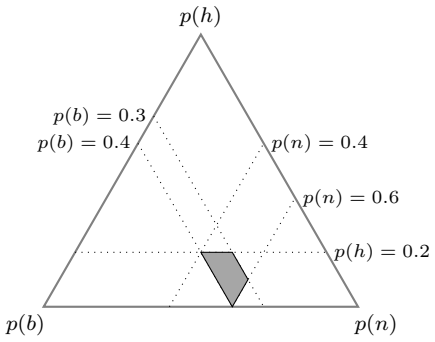


Fig. 2 Example 2 probability set in Barycentric coordinates.

2.2 Imprecise probabilities

Getting a reliable estimate $\hat{p}$ of the true distribution p is a challenging problem, as the true probability can never be perfectly identified (*e.g.*, due to noises, biases, lack of data, ...). An alternative solution is not to consider one estimate, but a (convex) set $\mathcal{P}$ of them, that may well contain the true one, and to adopt probabilistic mechanisms to such a setting. This is the goal of imprecise probability theory, whose basics we now recall (we refer to Walley [Walley, 1991] or Levi [Levi, 1983] for a detailed exposure). In the rest of the paper, $\mathcal{P}$ will be called *credal set*, as it represents our belief or knowledge about the class.

Example 2 We consider the obstacle recognition case in Example 1, in which a standard probabilistic classifier could yield an estimate such as

$$p(h) = 0.1, \qquad p(b) = 0.3, \qquad p(n) = 0.6.$$

In the imprecise probabilities framework, these estimates could be interval-valued and become, for example,

$$p(h) \in [0; 0.2], \qquad p(b) \in [0.3; 0.4], \qquad p(n) \in [0.4; 0.6],$$

with the width of the intervals reflecting the amount of information we have (the narrower the intervals, the more information/data we use). The corresponding credal set $\mathcal{P}$ is then the set of all precise probabilities $(p(h), p(b), (p(n))$ within these interval bounds. As $\mathcal{P}$ is a convex set, it can be described by considering the convex hull (noted CH) of its extreme points, which in our case is

$$\mathcal{P} = CH\{(0, 0.4, 0.6); (0.2, 0.3, 0.5); (0.2, 0.4; 0.4); (0.1, 0.3, 0.6)\}.$$

Finding these vertices can be done by using classical tools of convex geometry [Grunbaum et al., 1967], or by specific algorithms. The set $\mathcal{P}$ is represented in Figure 2 in barycentric coordinates.

Within the imprecise probabilistic setting, the notion of expectation is replaced by the notion of lower expectation $\underline{\mathbb{E}}[c]$ of a function $c : \Omega \to \mathbb{R}$, which is defined as the minimum expectation value obtained by considering distributions within $\mathcal{P}$:

$$\underline{\mathbb{E}}[c] = \min_{p \in \mathcal{P}} \mathbb{E}[c] = \min_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p(\omega) c(\omega). \tag{5}$$

The concept of upper expectations $\overline{\mathbb{E}}[c]$ is obtained by replacing $\min$ by $\max$ in Eq. (5), or by using the duality relation $\overline{\mathbb{E}}[c] = -\underline{\mathbb{E}}[-c]$. Computing $\underline{\mathbb{E}}[c]$ generally requires solving a linear program, which may be computationally costly if $|\Omega|$ is high or if $\mathcal{P}$ is defined by numerous constraints.

using lower (or upper) expectations, there are several ways to extend Equations (2)-(3) in order to take a decision. They can be divided into two groups depending on the type of decision: some rules give a unique output class, other may give a set of possible optimal classes (we refer to Troffaes [Troffaes, 2007] for a detailed exposure of the various decision rules and their relations). In our work, we concentrate on the second one, as we are interested in allowing indeterminate predictions. More precisely, we will consider the notion of maximality, that consists in constructing a partial order $\succ$ over classes and then to select the maximal (*i.e.*, non-dominated) ones in this partial order:

Definition 1 (Maximality) Under the maximality criterion,

$$\hat{y}_i \succ_{\mathcal{M}} \hat{y}_j \Leftrightarrow \underline{\mathbb{E}}[c_{\hat{y}_j} - c_{\hat{y}_i}] > 0. \tag{6}$$

This criterion extends Eq. (3), and can be interpreted as follows: $\hat{y}_i$ is preferred to $\hat{y}_j$ if exchanging $\hat{y}_i$ for $\hat{y}_j$ has a positive lower expected cost. To see that this is indeed a cautious rule and a robust version of Eq. (2), note that $\hat{y}_i \succ_{\mathcal{M}} \hat{y}_j$ if and only if $\underline{\mathbb{E}}[c_{\hat{y}_j}] > \mathbb{E}[c_{\hat{y}_i}]$ for all $p \in \mathcal{P}$. In particular, it reduces to Eq. (3) if $\mathcal{P}$ contains only one probability distribution. The (possibly) imprecise decision $\hat{Y}_{\mathcal{M}}$ obtained from this criterion is

$$\hat{Y}_{\mathcal{M}} = \left\{ \hat{y}_i \in \Omega \mid \nexists \hat{y}_j : \hat{y}_j \succ_{\mathcal{M}} \hat{y}_i \right\}. \tag{7}$$

Note that obtaining the order $\succ$ requires to perform at worst $K(K-1)$ computations ($K = |\Omega|$), one for each pair of classes. Also, while maximality has strong theoretical justifications [Walley, 1991, Sec. 3.9.], other decision criteria such as interval dominance [Yang et al., 2014] (also extending Eq. (2)) may be preferred if computational time is an important issue (*e.g.*, when the number of classes is high).

Example 3 Let us use the maximality criterion on the interval-valued probabilities given in Example 2 and the costs given in Example 1 to infer $\hat{Y}_{\mathcal{M}}$.

Let us first consider the pair $\{b, n\}$ and the difference $c_n - c_b$. We have

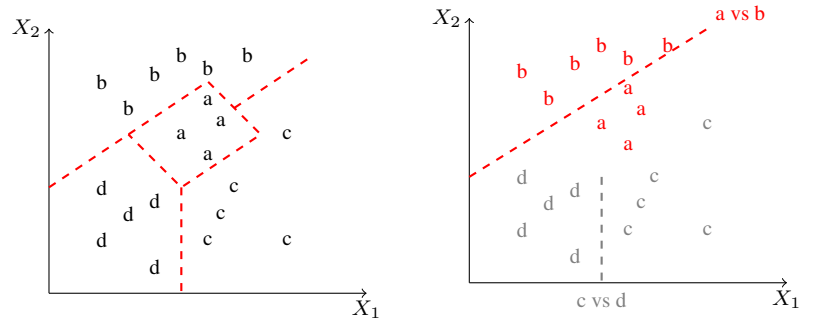
$$\underline{\mathbb{E}}[c_n - c_b] = \min\big(3p(h) + 4p(b) - 2p(n)\big) = 3 * 0.1 + 4 * 0.3 - 2 * 0.6 = 0.3$$

which is obtained for the extreme point $(0.1, 0.3, 0.6)$ of Example 2. As this is positive, we can infer $b \succ_{\mathcal{M}} n$. Furthermore, for the pair $\{h, b\}$ we have

$$\underline{\mathbb{E}}[c_h - c_b] = \min\big(-1p(h) + 1p(b) + 0p(n)\big) = -0.2 + 0.3 = 0.1$$

obtained for the extreme point $(0.2, 0.3, 0.5)$. This is again positive, so $b \succ_{\mathcal{M}} h$, and b ends up being the only non-dominated class, hence Eq. (7) gives us $\hat{Y}_{\mathcal{M}} = \{b\}$.

Roughly speaking, imprecise probability allows one to include in a knowledge model the amount of information it is based on. It means that if the amount of information is sufficient, it will behave as a classical model, and will only remain cautious when information is insufficient to make a precise inference or prediction. Using such models however imply an increase in computational complexity, or at least a complexity as high as the one of precise models.



Multi-class problem

Two separate binary sub-problems "a vs b" (in red) and "c vs d" (in gray)

Fig. 3 Multi-class problem and binary reduction

2.3 Reduction through binarization

Binary reduction/decomposition techniques [Dietterich and Bakiri, 1995; Rokach, 2006] have proved to be powerful approaches to efficiently solve multi-class problems. Their main idea is to decompose the original (potentially difficult) multi-class problem into a set of simpler, easier-to-solve binary problems. They also result in easier-to-interpret models that can be trained in parallel (allowing a computational gain). Figure 3 provides an illustration where the decision boundary (left side) of the original multi-class problem is much more complex than the pairwise boundaries involving two classes (right side). The downside of reduction techniques is that the results of sub-problems (for instance the boundaries "a vs b" and "c vs d" in Figure 3) need to be recombined in the end to find the global model.

Computationally speaking, there are at least two good reasons to use binary classifiers: it allows to use techniques providing probability bounds that are specific to binary problems (e.g., SVM evidential calibration [Xu et al., 2015], logistic regression [Corani and Mignatti, 2015]) in a multi-class problem, and many learning methods are computationally more tractable when the output space is binary (e.g., computing entropy bounds in credal decision trees [Abellán and Masegosa, 2012] is a lot simpler for binary spaces).

Formally speaking, binary decomposition consists in forming ℓ pairs of events $\{A_i, B_i\}$ ($i \in [1, \ell]$) where $A_i \cap B_i = \emptyset$ and $A_i, B_i \subseteq \Omega$ and to estimate whether the true class ω belongs to A_i or B_i for all $i = 1, \ldots, l$ instead of directly estimating the joint model over Ω. In a probabilistic setting, this means that we must provide estimates $\hat{p}(A_i \mid \{A_i, B_i\}) = \alpha_i$ and $\hat{p}(B_i \mid \{A_i, B_i\}) = 1 - \alpha_i$, using what is usually called a *base classifier* for each binary problems. From these conditional estimates can be derived the constraints

$$\begin{cases} \sum_{\omega \in A_i} \hat{p}(\omega) = \alpha_i \sum_{\omega \in A_i \cup B_i} \hat{p}(\omega) \ (i = 1, \ldots, l) \\ \sum_{\omega \in \Omega} \hat{p}(\omega) = 1 \end{cases} \tag{8}$$

on the joint probability over Ω.Generally, those constraints will be inconsistent [Hastie et al., 2009; Wu et al., 2004; Destercke and Quost, 2011], in the sense that no feasible solution to Eq. (8) will exist. How to solve this inconsistency is not an obvious problem and there is no unique best solution, even when one relax the constraints by allowing probabilities to become interval-valued [Destercke and Quost, 2011], in which case $\hat{p}(A_i \mid \{A_i, B_i\})$ and $\hat{p}(B_i \mid \{A_i, B_i\})$ are only known to lie in intervals. A usual strategy, in the precise case, is

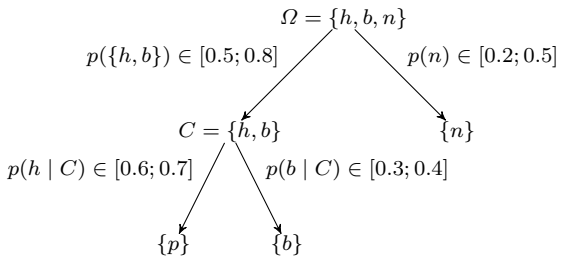


Fig. 4 Imprecise probabilistic nested dichotomy tree

to find a joint probability by minimizing a given distance [Hastie and Tibshirani, 1998; Wu et al., 2004] to the estimates $\hat{p}(y \mid \{A_i, B_i\})$. This however requires to solve an optimization problem at each classification.

One particular type of binary decomposition does not face this issue, as it always provides probability estimates resulting in consistent constraints: nested dichotomies [Fox, 1997], on which we will focus. As the constraints induced by this decomposition are ensured to be consistent, we will use the notation p instead of $\hat{p}$ in the rest of the paper. In summary, nested dichotomies present the same advantages as binary decompositions, without sharing their main drawback.

3 Imprecise nested dichotomies

The principle of nested dichotomies is to form a binary tree structure with the classes, which determines the binary sub-problems to be solved. The technique consists in recursively partitioning a tree node $C \subseteq \Omega$ into two subsets A and B (a dichotomy), until every leaf-node corresponds to a single class value ($|C| = 1$). The root node is the whole set of classes Ω.

Therefore, each node C is associated with a binary classification problem (solved by a chosen base classifier) where we should decide if the class belongs to the set A or B. If a standard (precise) probabilistic base classifier is used, then we obtain the conditional probabilities $p(A|C)$ and $p(B|C) = 1 - p(A|C)$, resulting in what we call precise nested dichotomies.

If an imprecise probabilistic base classifier is used, each node C is associated to an interval $p(A|C) \in [\underline{p}(A \mid C); \overline{p}(A \mid C)]$ rather than a single value. By duality of imprecise probabilities [Walley, 1991, sec.2.7.4.], we have $\underline{p}(B \mid C) = 1 - \overline{p}(A \mid C)$ and $\overline{p}(B \mid C) = 1 - \underline{p}(A \mid C)$. Precise nested dichotomies are retrieved when $\underline{p}(A \mid C) = \overline{p}(A \mid C)$ for every node C, meaning that imprecise nested dichotomies generalize the precise case. Figure 4 pictures a nested dichotomy tree together with its conditional probability constraints.

Let us also note that local models can be trained independently: once the tree structure is determined, the computation of conditional probabilities by base classifiers can be done simultaneously, for both training and testing. Even if we do not make copies of the data set, we may still parallelise the computation for tree nodes of the same depth as they work on disjoint parts of the data.

Making decisions with precise nested dichotomies by evaluating Eq. (3) is very easy, using the nested structure. Assume we have a split $\{A, B\}$ of a node C, then given a real-valued function c defined over $\{A, B\}$ the expected cost of the node C is defined as

$$\mathbb{E}_C(c) = p(A \mid C)c(A) + p(B \mid C)c(B) \tag{9}$$

and Eq. (9) remains true for imprecise nested dichotomies [De Cooman and Hermans, 2008], in the sense that the lower expectation $\underline{\mathbb{E}}_C(c)$ can simply be computed by:

$$\underline{\mathbb{E}}_C(c) = \min \begin{pmatrix} \underline{p}(A \mid C)c(A) + \overline{p}(B \mid C)c(B); \\ \\ \overline{p}(A \mid C)c(A) + \underline{p}(B \mid C)c(B) \end{pmatrix}. \tag{10}$$

with $\underline{p}(A \mid C)$,$\overline{p}(A \mid C)$ estimated by the local model of node C. As Eq. (10) consists in applying Eq. (9) twice, computations with imprecise nested dichotomies are only twice as costly as with precise ones, a quite acceptable increase.

If A, B are singletons (leaf nodes), then $c(A), c(B)$ take as values their associated cost functions. If A, B are inner nodes, then $c(A), c(B)$ correspond to the expected cost of the nodes A, B which can be computed recursively. Therefore, the expected cost of the global model can be obtained easily by backward recursion starting from the leaf nodes to the root. This again remains true in the imprecise case [De Cooman and Hermans, 2008]. If we use the maximality criterion for decision-making, then function c at the leaf nodes corresponds to the difference of two cost functions $c_{\hat{y}_1} - c_{\hat{y}_2}$ ($\hat{y}_1, \hat{y}_2 \in \Omega$).

We can derive a recursive algorithm for computing $\mathbb{E}_\Omega$. For clarity purpose, we only write the algorithm in the precise case, which derives Algorithm 1 from Eq. (9). In the imprecise case, we just replace every *return* statement by the corresponding one in Eq. (10).

Algorithm 1: Function $\mathbb{E}$ (computing the global expectations of c)

Input: Cnode=current node, which is initiated to the whole set Ω
$A, B \leftarrow$ children node of $Cnode$;
if *A and B are singletons* **then**
  return $p(A|Cnode)c(A) + p(B|Cnode)c(B)$
else if *A is singleton* **then**
  `/* recursion over node` B`:computing` $\mathbb{E}(B)$ `*/`
  return $p(A|Cnode)c(A) + p(B|Cnode)\mathbb{E}(B)$
else if *B is singleton* **then**
  `/* recursion over node` A`:computing` $\mathbb{E}(A)$ `*/`
  return $p(A|Cnode)\mathbb{E}(A) + p(B|Cnode)c(B)$
else both children nodes are not singletons
  `/* recursion over both` A, B`:computing` $\mathbb{E}(A), \mathbb{E}(B)$ `*/`
  return $p(A|Cnode)\mathbb{E}(A) + p(B|Cnode)\mathbb{E}(B)$
end

Example 4 We show in this example what can be said about the preference relation between the classes b and h with the imprecise conditional probabilities given in Figure 4.

Using the maximality criterion, we compute recursively the expected cost $\underline{\mathbb{E}}_\Omega[c_b - c_h]$ (see Fig. 5). We first compute Eq. (10) for the node $C = \{h, b\}$:

$$\underline{\mathbb{E}}_{\{h,b\}}[c_b - c_h] = \min\big(0.6 - 0.4; 0.7 - 0.3\big) = 0.2,$$

and we can now apply it to the node $\{h, b, n\}$, as both its children have a precise value. this gives

$$\begin{aligned} \underline{\mathbb{E}}_{\{h,b,n\}}[c_b - c_h] &= \min\big(0.2 \cdot 0.8 + 0 \cdot 0.2; 0.2 \cdot 0.5 + 0 \cdot 0.5\big) \\ &= 0.1 > 0. \end{aligned}$$

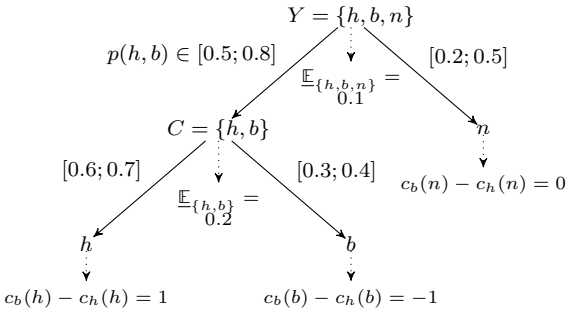


Fig. 5 Computation of the expected cost $\underline{\mathbb{E}}[c_b - c_h]$ with imprecise probabilities

Therefore, we can conclude that the class "human" is preferred to "bicycle" using the maximality criterion in this example.

As Example 4 and Eq. (10) show, computing with imprecise nested dichotomies remains quite tractable whatever the cost function is, therefore not facing the tractability issue associated to other imprecise probabilistic approaches.

Imprecise nested dichotomies are therefore quite attractive computationally speaking, but as their tree structure is not uniquely defined (in contrast with other decomposition methods such as one-vs-one or one-vs-all), there is a need to provide a method to do so. This will be discussed in our experiments.

4 Experiments

In Section 3, we have discussed the computational advantages of imprecise nested dichotomies, but it now remains to test whether they can provide good classifiers in terms of predictions.

This is the purpose of this section, where we perform experiments, first on standard multi-class problems (Section 4.3), then on ordinal classification [Frank and Hall, 2001] (Section 4.4), this latter problem having the particular feature that Ω has some structure.

Before presenting our results, we explain how the problem of tree structure is tackled in Section 4.1, and provide our general experimental set-up (base classifier and evaluation measures) in Section 4.2.

4.1 Choosing a dichotomy structure

Two common solution to the problem of choosing a dichotomy structure is to use a set or ensemble of classifiers [Frank and Kramer, 2004], or to select a structure providing a good accuracy. Indeed, picking the optimal tree structure is not an easy task, as the number of such structures grows very fast with the size of Ω [Frank and Kramer, 2004, Sec 3.]. How we implemented the first and second solutions is explained in Sections 4.1.1 and 4.1.2, respectively.

4.1.1 Forest of nested dichotomies

One way around the issue of finding/building an optimal tree structure is that we can use a set of Λ randomly and uniformly generated trees instead. In this case, the decision pro-

cess specified in Section **??** has to be adapted, as the results of the different trees need to be aggregated. There are two main usual ways to perform this aggregation: voting scheme over the decisions made by each classifier, or aggregating the (imprecise) probabilities provided by each classifier, from which can then be deduced the decision. Both ways will be investigated in our experiments.

Voting techniques are widely used in the literature for ensemble learning methods [Rokach, 2010]. In our experiments, we base our aggregation on the majority voting technique. Let $(\hat{Y}^\lambda)_{\lambda\in\Lambda}$ be the predictions obtained by each tree given the input features $\mathbf{x}$, we define the final prediction set as

$$\hat{Y} = \left\{\omega \in \Omega : \sum_{\lambda\in\Lambda} \mathbb{1}_{\hat{Y}^\lambda}(\omega) > \frac{|\Lambda|}{2}\right\} \text{ if not empty, or } \Omega \text{ otherwise.} \tag{11}$$

$\hat{Y}$ contains classes which are predicted by more than half of the classifiers in the set (or all classes in case none reaches majority). As each classifier may vote for more than one class, then $\hat{Y}$ may be imprecise, and we may hope that the votes of bad classifiers will be less important.

To aggregate imprecise probability estimates, we will simply use a standard average or arithmetic mean, which is equivalent to compute the sum of expected costs. If we denote by $\underline{\mathbb{E}}^\lambda$ the lower expected cost associated to a tree-structure λ, then the aggregated lower expected cost and the resulting decision are given by

$$\hat{y}_1 \succ_{\mathcal{M}} \hat{y}_2 \Leftrightarrow \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda} \underline{\mathbb{E}}^\lambda[\hat{y}_2 - \hat{y}_1] > 0 \tag{12}$$

Contrary to the majority voting, this approach takes the strength of belief of the estimations (*i.e.*, the values of expected costs) into consideration. For instance, if one classifier estimates a very high expected cost for a preference (so, in a sense, we can say that it is sure about its estimation), then another classifier stating a weak disagreement (a slightly negative expected cost) about the same preference will not cancel out the belief of the first classifier. This particularity may be either beneficial or detrimental depending on the capacity of the base classifier to give accurate probability estimations.

Another issue is then to determine what is a good number Λ of trees to use. Our experiments suggested that this is highly correlated with the size of Ω: an higher number of trees seem desirable when $|\Omega|$ increases. In our experiments, we fixed this number to 50, which gave better results than the 20 recommended by Frank and Kramer [Frank and Kramer, 2004] on the data sets counting more then 5 classes. Note that we could have sought to optimize the parameters of our ensemble approach, however this would be out of the scope of these paper experiments, as (1) we want to estimate the efficiency of nested dichotomies, not of ensemble techniques and (2) how to properly optimise ensemble techniques in imprecise probability settings largely remains an open issue.

4.1.2 Building a single dichotomy structure

While forests of nested dichotomies avoid the issue of choosing one structure, they do not solve it. Yet, it may be useful to know whether a single structure can perform as well as (or better) than other classifiers or than an ensemble approach. Indeed, a single structure is far easier to read and interpret. Furthermore, it could also be derived from some expert opinions: family of genes, species of animals, images containing the same objects.

It is clear that in benchmark experiments, we cannot derive the tree structure from expert information, and that problem structures will not be specific enough to pick a single structure (it may discard some particular structures, though, as we shall see in Section4.4). One possibility is then to use statistical measures to separate the classes, many of which are reviewed by Lorena and De Carvalho [2010], yet there are no theoretical guarantees that one measure will provide a better structure than another, and the performance of a given measure may greatly vary between data sets.

In our experiments, we simply selected, out of the forest of 50 nested dichotomies, the one obtaining the best performance (according to the evaluation measure of Section 4.2.2) on the training set in a cross-validation setting. Again, we could probably think of more powerful selection techniques, yet we will see that this simple approach already provides quite good performances.

4.2 Experimental set-ups

We will now detail the rest of our experimental setup, in particular our choice of base classifier to be trained for each node of the nested dichotomies, and the measure we use (that extends classical accuracy) to evaluate indeterminate classifiers.

4.2.1 Choice of base classifier

In principle any base classifier returning probability bounds can be combined with nested dichotomies. Our particular choice of base classifier was guided by two considerations: first, it had to be a probabilistic classifier that could be applied directly to multi-class problem (to compare it with its decomposed counterpart) and that had an imprecise counter-part (to evaluate the benefits of adding imprecision in estimates); second, it had to be relatively simple to train, as decomposition techniques involve training a set of binary classifiers (one for each node).

For these reasons, we use the Naive Credal Classifier (NCC) [Zaffalon, 2002] which is an extension of the Naive Bayesian Classifier (NBC) to the imprecise probability framework. These classifiers are known to provide good predictive accuracy despite their simplicity. Technical details about this classifier are given in Appendix A. In every use of the NCC, we set $s = 2$ as value of its hyper-parameter .

As NCC cannot handle continuous variables natively, all continuous features in data sets were discretized by dividing their domain in 5 intervals of equal width. We did not use a supervised discretization method [Fayyad and Irani, 1993], since the involved classes change between the initial multi-class problem and each binary sub-problem.

4.2.2 Performance evaluation criterion

Comparing classifiers that return indeterminate predictions with classifiers returning determinate ones is a complex problem. Indeed, compared to the usual setting where all classifiers are determinate, measures of performance have to include the informativeness of the predictions in addition to the accuracy. Zaffalon *et al.*[Zaffalon et al., 2012] discuss this issue in details under a unitary loss assumption, using a betting interpretation. They show that the discounted accuracy, which rewards a cautious prediction $\hat{Y}$ class with $1/|\hat{Y}|$ if the true class is in $\hat{Y}$, and zero otherwise, is a measure satisfying a number of appealing properties.

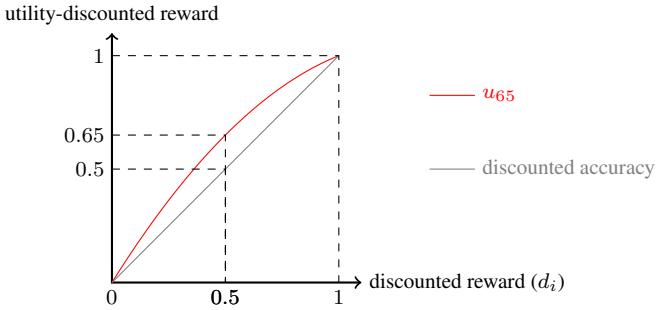


Fig. 6 Quadratic utility function u_{65} obtained from the discounted accuracy

However, they also show that the discounted accuracy makes no difference between an imprecise classifier providing indeterminate predictions and a random classifier: for instance, in a binary setting, supposing an imprecise classifier which always returns both classes, then it would have the same discounted accuracy as a classifier picking the class at random, yet the imprecise classifier displays a lower variance (it always receives $1/2$ as reward, while the random one would receive a reward of 1 half of the time, and 0 the other half).

This is why a decision maker that wants to reward cautiousness and reliability should consider modifying the discounted accuracy by a risk-averse utility function [Zaffalon et al., 2012]. Here, we consider the u_{65} function: let $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$ be the set of test data and $\hat{Y}_i$ our (possibly indeterminate) predictions, then u_{65} is

$$u_{65} = \frac{1}{n} \sum_{i=1}^{n} -0.6 d_i^2 + 1.6 d_i, \tag{13}$$

where d_i is the discounted accuracy $d_i = \frac{\mathbb{1}_{\hat{Y}_i}(y_i)}{|\hat{Y}_i|}$, and $\mathbb{1}_{\hat{Y}_i}$ the indicator function of $\hat{Y}_i$.

The reason of this specific utility function is that we want to define a quadratic function which keeps some appealing properties of the discounted accuracy: when the correct class is not in $\hat{Y}$ the accuracy should stay at 0, and when the prediction is both precise and accurate ($|\hat{Y}| = 1, y_i \in \hat{Y}$) then the accuracy should be 1. Moreover, this utility function should express our risk-aversion by giving a higher reward to imprecise but correct predictions compared to the discounted accuracy. We choose to retain the u_{65} score, that gives a reward of 0.65 for a prediction having a discounted accuracy of 0.5. u_{65} is a quadratic function, shown in Figure 6, satisfying these properties.

It has been shown by [Zaffalon et al., 2012] that this approach is consistent with the use of F_1 measures proposed by [del Coz and Bahamonde, 2009] as a way to measure the quality of indeterminate classifications. Also, u_{65} is less in favour of indeterminate classifiers than the F_1 measure, meaning that we remain quite fair to the determinate classifier.

4.3 Results on multiclass problems

Our first batch of experiments concern the classical multi-class problem. We will first presents the data sets and methods compared, before analysing the obtained results.

Table 2 Data sets details for multiclass problems

Name	(C)ont/(D)isc features	# instances	# classes
balance-scale	D	625	3
wine	C	178	3
iris	C	150	3
car	D	1728	4
lymph	D	148	4
grub-damage	C	155	4
nursery	D	12960	5
page-blocks	C	5473	5
glass	C	214	6
zoo	D	101	7
segment	C	2310	7
ecoli	C	336	8
pendigits	C	10992	10
soybean	D	562	15

4.3.1 Data sets and methods

For this first set of experiments, we took 14 data sets of the UCI machine learning repository [Lichman, 2014], whose details are given in Table 2. They are presented in increasing order of the number of classes.

We note that these are general purpose machine learning data sets: there is no predefined structure for the classes (no prior information nor expert opinion) and no misclassification costs. As it is hard to compute non unitary cost functions using NCC, contrary to the nested dichotomies, we have used exclusively data sets with unitary costs.

Once the continuous attributes were discretised (in 5 intervals of equal width), we applied the following methods

- ND+NBC: nested dichotomies with the naive Bayes classifier as base classifier of each node,. The selected structure is the one obtaining the best accuracy out of 50 randomly generated ones, as specified in Section 4.1.2;
- NCC: the naive credal classifier used as a reference multi-class classifier;
- DC: the method developed by [del Coz and Bahamonde, 2009] to derive imprecise predictions from precise probability estimates (provided by ND+NBC). See Appendix B for a more detailed presentation of the method;
- ND+NCC: same as ND+NBC, but with NCC as base classifier, thus predictions can be imprecise;
- Forest(vote): ensemble of nested dichotomies with the NCC where the majority voting technique is used as shown in Section 4.1.1;
- Forest(mean): same as above but with the mean of expected costs as aggregator.

This allows us to perform three kinds of comparisons: (1) precise (ND+NBC) vs imprecise dichotomies, (2) imprecise nested dichotomies vs their imprecise multi-class counterpart (NCC) and (3) indeterminate predictions derived from imprecise nested dichotomies vs indeterminate predictions derived from precise nested dichotomies (DC).

4.3.2 Comparison of performances

The results in terms of u_{65} are presented in Table 3. The results are obtained from a 10-fold cross validation and using 50 dichotomy trees. To make the table more readable, the best

Table 3 Comparison of discounted accuracy (u_{65})

	u_{65} score expressed in percentage (rank)					
	ND+NBC	NCC	DC	ND+NCC	Forest(vote)	Forest(mean)
balance-scale	90.72 (5)	90.78 (3)	84.39 (6)	**90.88 (1)**	90.77 (4)	90.82 (2)
wine	95.51 (6)	97.07 (2.5)	95.84 (5)	96.13 (4)	97.07 (2.5)	**97.16 (1)**
iris	93.33 (5)	93.27 (6)	93.5 (3)	**93.7 (1)**	93.5 (3)	93.5 (3)
car	88.37 (2)	86.16 (4)	86.35 (3)	**89.03 (1)**	85.85 (5)	85.46 (6)
lymph	82.64 (2)	70.07 (6)	**84.36 (1)**	82 (3)	73.46 (5)	75.48 (4)
grub-damage	47.52 (6)	52.48 (2)	49.86 (5)	50.44 (4)	52.2 (3)	**52.9 (1)**
nursery	91.54 (2)	90.46 (4)	88.17 (6)	**91.73 (1)**	90.34 (5)	90.57 (3)
page-blocks	91.67 (3)	91.17 (6)	**92.03 (1)**	91.83 (2)	91.56 (5)	91.61 (4)
glass	53.74 (2)	51.38 (6)	**58.7 (1)**	51.81 (5)	53.38 (3)	52.37 (4)
zoo	91.73 (2)	83.79 (5)	**92.38 (1)**	85.22 (3)	84.68 (4)	81.6 (6)
segment	86.58 (5)	**89.92 (1)**	86.84 (3.5)	86.4 (6)	86.84 (3.5)	88.34 (2)
ecoli	80.95 (4)	79.47 (5)	**82.22 (1)**	78.41 (6)	81.24 (2.5)	81.24 (2.5)
pendigits	81.41 (6)	**85.81 (1)**	82.13 (4)	81.42 (5)	82.73 (3)	84.48 (2)
soybean	87.37 (5)	**90.26 (1)**	87.62 (4)	82.99 (6)	89.27 (2)	88.01 (3)
average rank	3.93	3.75	3.18	3.43	3.61	3.11

scores are indicated in bold and the ranks of the methods are also given (if there are ties, then tied elements are given the averaged rank).

To statistically verify the differences between the algorithms, we follow the approach suggested by Demšar [Demšar, 2006] and we apply the Friedman test [Friedman, 1937] on the ranks obtained by the algorithms for each data set. We find a p-value of 0.84 so that we can not reject the null hypothesis. It means that all methods have comparable performances in terms of accuracy and the differences are not statistically significant.

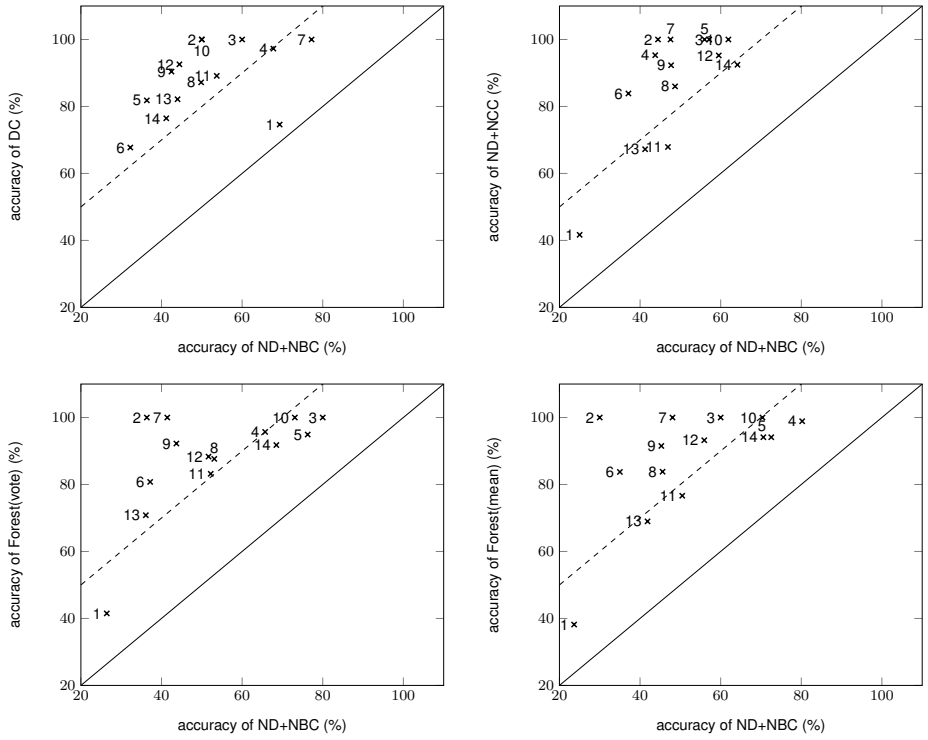
We can notice that our approaches have generally a very different behaviour compared to the DC method: the difference of ranks (and of performance) on a given data set is often very important. It is also interesting to note that the single dichotomy tree method seems to perform well when there are few classes (equal or less than 5), and becomes less efficient when data sets have more than 5 classes. This is likely due to the fact that the "optimal" tree structure is determined with 50 trees only, which is poorly representative of the set of possible structures when the number of class is high. This indicates that a single tree may provide results as good as a forest of trees, provided an *optimal* or good dichotomy structure can be identified.

Finally, it is also useful to notice that both forests and binary decompositions offer more room to improve classification accuracies: one could, in principle, optimize the aggregation procedure (similarly to what is done in other methods based on sets of models [Corani and Zaffalon, 2008; Cesa-Bianchi et al., 1997]) or optimize the base classifier for each binary sub-problem.

4.3.3 Gain of accuracy on indeterminate predictions

The main goal of imprecise classifiers is to make indeterminate predictions including the true class on cases (and ideally only on those) where the determinate classifier fails. To show that this is indeed the case here, Figure 7 displays, on the instances where indeterminate predictions are made by the imprecise classifier (DC, ND+NCC, Forest(vote) and Forest(mean)), the percentages of times the true class is within the predictions, both for the imprecise classifier and the precise ND+NBC.

While it is trivial that the imprecise classifier is always more accurate than its precise counterpart, it is important to note that the accuracy of the precise classifier on these inde-



1 = balance-scale	2 = wine	3 = iris	4 = car	5 = lymph	6 = grub-damage	7 = nursery
8 = page-blocks	9 = glass	10 = zoo	11 = segment	12 = ecoli	13 = pendigits	14 = soybean

Fig. 7 Non-discounted accuracy of the precise method "ND+NBC" vs different imprecise classifiers when a indeterminate prediction is made by the latter.

terminate predictions is usually much lower than their accuracy on the whole data set (see Table 3). One of the most illustrative case is the *wine* data set (number 2 in Fig. 7), where the accuracy of both precise and imprecise classifiers are above 90% on the whole data set, but the accuracy of precise classifier drop to approximatively 50% on the data instances where indeterminate predictions are made by the imprecise classifiers.

This drop of performance for the precise classifier, when we restrain to instances where indeterminate predictions are made, indicates that they are among the "hard to classify" instances for the precise method. All four studied imprecise classifiers share this property. However, if we consider the number of data sets where the gain of accuracy is more than 30% (points above the dotted line in Fig. 7), then it seems that DC (11 data sets above the dotted line) and ND+NCC (10 data sets above) succeed slightly better at finding these "hard to classify" instances than the two ensemble learning methods (8 for "vote" and 7 for "mean").

Finally, one can notice that some points have very different positions in the graph corresponding to Del Coz *et al.* approach (on the top-left of Fig. 7) and in the graphs corresponding to imprecise probabilistic models. This difference can be explained by the fact that Del Coz *et al.* approach will produce indeterminate predictions only in case of ambiguity (see Figure 1), while imprecise probabilistic ones will be indeterminate in both cases of ambi-

Table 4 Percentage of indeterminate predictions made by imprecise classifiers

	Percentages of imprecise predictions				
	NCC	DC	ND+NCC	Forest(vote)	Forest(mean)
balance-scale	10,4	30,24	**7,68**	8,48	8,8
wine	6,18	**2,25**	5,06	6,18	5,62
iris	4	3,33	4,67	3,33	3,33
car	5,38	33,91	**3,7**	4,05	5,27
lymph	50	**7,43**	16,89	39,86	34,46
grub-damage	52,9	**20**	40	50,32	51,61
nursery	1,07	27,48	1,1	**0,54**	0,59
page-blocks	1,86	5,28	2,74	1,48	**1,24**
glass	61,21	34,11	**30,37**	48,13	49,53
zoo	23,76	**1,98**	20,79	25,74	26,73
segment	**3,07**	7,58	3,51	4,89	4,63
ecoli	22,62	**8,04**	12,5	17,86	17,56
pendigits	1,16	8,01	1,61	**0,66**	1,17
soybean	8,72	**3,02**	18,86	12,99	15,12

guity and lack of information. A significant difference between the placement of points for these two approaches therefore provides useful indications to the decision maker about the data sets, for example:

- for the *balance-scale* data set (number 1 in Fig. 7), and to a lesser extent for the *nursery* data set (number 7), we can see that the precise approach still has a high accuracy (around $70 - 80\%$) on those instances for which Del Coz *et al* approach make indeterminate prediction, while this accuracy drops on the instances for which imprecise probabilistic approaches make indeterminate prediction (respectively around 20% and 40%). This indicates that "hard to classify" instances for these data sets are mainly those for which we lack information. Therefore, trying to collect more data may significantly improve our results;
- for the soybean data set (number 14), the situation is reversed, indicating that for this data set, "hard to classify" instances are mainly those for which some ambiguity arises. In such cases, trying other classifiers and/or adding more discriminating features may prove more useful than collecting new data.

4.3.4 Comparison of indeterminacy

We show the percentages of indeterminate predictions made by the imprecise classifiers in Table 4. We can see that our methods are most of the time more determinate than NCC.

We can again notice that the behaviour of our approach is very different from the approach of Del Coz *et al.*. Indeed, even if the performance of both approaches have similar u_{65} scores, the imprecision level is very different for nearly all data sets. They even seem to be antagonistic: every time one method has a low imprecision level, the other method would have a several times higher one.

Table 4 also shows that the level of imprecision range from very low (around 1% for *pendigits*) to very high (around 50% for *grub-damage*), which shows that the imprecise probabilities is well capable of adapting the level of imprecision according to the data set. Also, there is no apparent correlation between the performance and the imprecision level: a higher level of imprecision does not imply a higher u_{65} accuracy, which supports the fact that u_{65} remains a fair criterion for comparing precise and imprecise classifiers.

Finally, this table also allows us to give some insights about the different ways to build dichotomy trees. We can see that using a single dichotomy tree (ND+NCC) allows to obtain more determinate predictions while achieving a good predictive accuracy, as the average imprecision level is lower than the one of forests and the performance is similar. This suggests that using a single tree induced by expert opinion or by the class structure will not necessarily lessen the performances (in fact, we will see in Section 4.4 that the performances can even be increased). On the other hand, using forest of dichotomy trees allows for more cautious predictions, and can be very efficient at dealing with problems where there is no available information about the structure of the classes.

4.4 Results on ordinal classification problems

In the second set of experiments, we consider the problem of ordinal classification and related data sets. Compared to multiclass problems, ordinal classification has the particular feature that some structure exist between the classes, as these latter are ordered. For instance, the rating of movies can be one of the following labels: *Very-Bad*, *Bad*, *Average*, *Good*, *Very-Good* that are ordered from the worst situation to the best.

This particular structure can be accounted for when building nested dichotomies: as the classes are ordered, it makes poor sense to make binary split where one (or both) subset contains non-adjacent classes. For instance, given the labels $\{$*Bad*, *Average*, *Good*$\}$, grouping *Bad* and *Good* together and leaving *Average* is contradictory to the given order. Therefore, given a node C of a dichotomy tree, it only makes sense to split the set of labels $\{\omega_i, \ldots, \omega_j\}$ into $A = \{\omega_i, \ldots, \omega_k\}$ and $B = \{\omega_{k+1}, \ldots, \omega_j\}$ with $i \leq k < j$.

4.4.1 Data sets and methods

As there is a general lack of benchmark data sets for ordinal classification data, we used regression problems that we turned into ordinal classification by discretizing the class variable, except for the data sets LEV that has 5 ordered classes and ESL, ERA that have 9 ordered classes. The details of the 15 data sets of the UCI machine learning repository [Lichman, 2014] are given in Table 5.

The results reported have been obtained with a discretisation of the class into 7 class values of equal width. We also performed experiments with 5 and 9 discretised classes, obtaining the same conclusions.

Applying a similar procedure than in multiclass problems, the results in this section are obtained from a 10-fold cross validation and using the u_{65} score with 50 dichotomy trees. We compare six methods :

- logreg: ordinal logistic regression used as a baseline classifier to compare our results. We use the Python implementation made by [Pedregosa, 2013].
- ND+NBC: same as for the multi-class setting, except that splits of generated tree respect the adjacency constraint.
- ADC [Alonso et al., 2008]: method which allows to derive imprecise predictions from precise probability estimates. The approach follows the same guidelines as the the one detailed in Appendix B, except that only predictions in form of adjacent classes are allowed.
- ND+NCC: same as ND+NBC, but with NCC as base classifier, thus predictions can be imprecise.

Table 5 Data sets details for ordinal classification

Name	#instances	#features	#classes
autoPrice	159	16	7
bank32NH	8192	33	7
boston housing	506	14	7
california housing	20640	9	7
delta ailerons	7129	6	7
elevators	16599	19	7
delta elevators	9517	7	7
friedman	40768	11	7
house8L	22784	9	7
house16H	22784	17	7
kinematics	8192	9	7
puma32H	8192	33	7
ERA	1000	4	9
ESL	488	4	9
LEV	1000	4	5

- Forest(mean): forest of nested dichotomies with arbitrary splits (not necessarily respecting the adjacency constraint).
- Forest(ordinal): forest of nested dichotomies with splits respecting the adjacency constraint.

This setting allows us to compare determinate methods (ND+NBC, logreg) to indeterminate ones (the 4 other methods). But most importantly, we are interested in evaluating the impact of adding this prior knowledge about the ordinal class structure. This is why we evaluate the forest approach with and without taking this information into consideration. The method developed by Alonso *et al.* is used as a state of art reference to compare performance.

4.4.2 Test results

Table 6 shows the obtained results in terms of u_{65} (that reduces to classical accuracy for the two determinate methods) as well as the rank of each classifier. As the compared methods are similar to the ones used in the multi-class case, interpretations of Section 4.3.3 and 4.3.4 about the gain of accuracy and indeterminacy remain true. Therefore, we will only focus on what differs from the multiclass case.

By applying the Friedman test on the ranks of the algorithms for each dataset, we obtain a p-value lower than 10^{-4} so that the null hypothesis can be safely rejected. It means that the performances of the classifiers are significantly different.

This confirms that in average the introduced indeterminacy in the predictions is not too important and is compensated by more reliable predictions. As the null hypothesis has been rejected, we use a Nemenyi test [Nemenyi, 1963] as a post-hoc test (see Fig. 8), and obtain that two classifiers are significantly different (with a significance level of 0.10) if the difference between their mean rank is higher than 1.8. Therefore, our approach with a single dichotomy tree outperforms the baseline (logistic regression) and is competitive with similar state of art methods.

The most remarkable point is that, while using forest of nested dichotomies allowed us to obtain very good performance in the multi-class case, they yield significantly worse performance than the state of art method in the case of ordinal problems, especially if the

Table 6 u_{65} scores (and ranks) for different methods on ordinal data sets

	u_{65} scores expressed in percentage (rank)					
Data set	Logreg	ND+NBC	ADC	ND+NCC	Forest(mean)	Forest(ordinal)
auto Price	62.89 (4)	63.52 (3)	63.93 (2)	**64.21 (1)**	62.54 (5)	61.23 (6)
bank32NH	75.89 (3)	75.84 (4)	**76.43 (1)**	76.4 (2)	75.59 (5)	69.44 (6)
boston housing	44.47 (6)	51.38 (4)	51.99 (2)	51.8 (3)	**52.5 (1)**	45.84 (5)
california housing	38.36 (6)	43.08 (3)	**45.07 (1)**	43.26 (2)	41.67 (5)	42.18 (4)
delta ailerons	53.71 (6)	75.16 (2)	74.91 (3)	**75.27 (1)**	74.66 (4)	74.61 (5)
elevators	**77.61 (1)**	63.78 (4)	64.26 (2)	63.84 (3)	61.39 (6)	61.58 (5)
delta elevators	55.84 (6)	66.29 (5)	66.51 (2)	66.44 (4)	66.5 (3)	**66.8 (1)**
friedman	46.37 (6)	60.82 (2)	60.33 (3)	**60.96 (1)**	56.84 (5)	58.42 (4)
house8L	**84.3 (1)**	82.78 (4)	83.04 (2)	82.8 (3)	82.68 (5)	82.6 (6)
house16H	**82.09 (1)**	80.05 (4)	80.47 (2)	80.08 (3)	79.84 (5)	79.81 (6)
kinematics	37.46 (6)	39.9 (3)	**42.77 (1)**	40.66 (2)	38.48 (5)	39.85 (4)
puma32H	36.33 (6)	51.27 (5)	53.03 (3)	**53.07 (1)**	53.05 (2)	51.91 (4)
ERA	23.2 (6)	24.4 (5)	25.65 (4)	**26.56 (1)**	26.52 (2)	26.31 (3)
ESL	47.75 (6)	62.7 (5)	63.44 (3)	63.27 (4)	**65.54 (1)**	64.84 (2)
LEV	46.3 (6)	59.4 (3)	**60.33 (1)**	59.48 (2)	57.94 (4)	56.49 (5)
Average rank	4.67	3.73	2.13	2.2	3.87	4.4

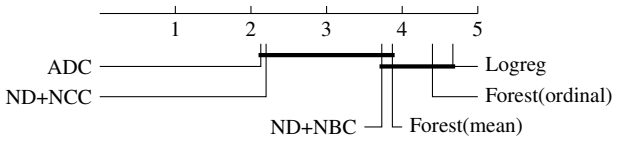


Fig. 8 Nemenyi post-hoc test results on algorithms. Groups of algorithms that are not significantly different (at a significance level of 0.10) are linked with a bold line

ordinal structure is taken into consideration. On the other hand, our approach using one single dichotomy tree selected among the forest remains competitive. Since ensemble learning techniques requires the majority of component to be competent, a possibility is that the results are affected by classifiers with bad performance. This is most likely what happened here, despite that the forest contains adequate dichotomy structures (including the one used for ND+NCC, the single tree approach), the final results of the forest are strongly biased by non-adequate dichotomy structures.

In the multiclass case, there is no strong presumption about the structure of the classes, so the ensemble approach is well-suited. Here in ordinal classification, choosing a specific dichotomy structure to integrate the prior knowledge becomes much more interesting and significant.

5 Conclusion

In this paper, we have proposed a method to learn indeterminate classifiers, in the sense that they provide indeterminate predictions when information is insufficient to provide reliably a determinate one. More precisely, this approach extend the nested dichotomies, a special binary decomposition technique which guarantee consistency of solutions, to the imprecise probability framework. The extension consists in representing probabilities with interval-valued estimates rather than precise ones, the width of the interval reflecting the lack of knowledge.

Our experiments on different data sets show that allowing for indeterminacy can increase the reliability and the cautiousness of predictions, which is interesting or even crucial for many application fields. More specifically, the added indeterminacy tends to focus on those instances that are hard to classify for determinate classifiers. Moreover, combining nested dichotomies with classical imprecise methods allowed us to reduce the level of indeterminacy while maintaining the same level of accuracy. Besides, we showed that a wise use of prior knowledge in the case of ordinal classification can be beneficial and can increase significantly the performance of our approach.

We could probably improve both the efficiency of inferences, *e.g.*, by studying extensions of labeling trees to imprecise trees [Bengio et al., 2010], or their accuracy by using more complex classifiers, *e.g.*, credal averaging techniques [Corani and Zaffalon, 2008]. Yet, as the advantages of using binary decompositions are usually lower when using complex estimation methods, the benefits of such extensions would be limited and counter-balanced by their computational complexity.

In our experiments, we have focused on unitary misclassification costs and their extensions to indeterminate predictions, since it is not obvious how to compare determinate and indeterminate classifiers with generic cost functions. Yet, our approaches can easily handle generic costs (in contrast with the multiclass naive credal classifier [Zaffalon, 2002] and the method of Del Coz *et al.*), as shown in Section 2.2 and Section **??**. However, there are many problems where unitary costs are not the most natural ones, this is the case for instance in ordinal classification problems, where costs should integrate the structure of the classes (using, for example, absolute error). Our future efforts will therefore focus on determining meaningful ways to compare determinate and indeterminate classifiers using non unitary cost functions.

A Presentation of the Naive Credel Classifier

The NCC preserves the main properties of the Naive Bayesian Classifier, such as the assumption of attribute independence conditionally to the class. For the standard NBC, the assumption of attribute independence can be written as:

$$p(x_1, \ldots, x_m \mid \omega) = \prod_{i=1}^{m} p(x_j \mid \omega), \tag{14}$$

where $(x_1, \ldots, x_m) \in (X_1, \ldots, X_m)$ are the input features and $\omega \in \Omega$.

In binary problems where we have to differentiate between two complementary events (set of classes) A and B, the NCC consists in using lower/upper bounds of prior probabilities to estimate the posterior ones [Zaffalon, 2002]:

$$\underline{p}(A|x_1,\ldots,x_m) = \min \left(\begin{array}{c} \dfrac{\underline{p}(A)\prod_{i=1}^{m}\underline{p}(x_i \mid A)}{\prod_{i=1}^{m}\underline{p}(x_i \mid A)\underline{p}(A) + \prod_{i=1}^{m}\underline{p}(x_i \mid B)\overline{p}(B)}, \\ \dfrac{\overline{p}(A)\prod_{i=1}^{m}\underline{p}(x_i \mid A)}{\prod_{i=1}^{m}\underline{p}(x_i \mid A)\overline{p}(A) + \prod_{i=1}^{m}\underline{p}(x_i \mid B)\underline{p}(B)} \end{array} \right)$$
$$= 1 - \overline{p}(B|x_1,\ldots,x_m). \tag{15}$$

The lower probability $\underline{p}(B|x_1,\ldots,x_m) = 1 - \overline{p}(A|x_1,\ldots,x_m)$ can be obtained in the same way. Using the Imprecise Dirichlet Model (IDM) [Bernard, 2005], we can compute these probability estimates using the training data by simply counting occurrences :

$$\underline{p}(x_i \mid A) = \frac{occ_{i,A}}{occ_A + s}, \qquad \overline{p}(x_i \mid A) = \frac{occ_{i,A} + s}{occ_A + s}, \tag{16}$$

$$\underline{p}(A) = \frac{occ_{i,A}}{occ_{A,B} + s}, \qquad \overline{p}(A) = \frac{occ_{i,A} + s}{occ_{A,B} + s}, \tag{17}$$

where $occ_{i,A}$ is the number of instances in the training set where the attribute X_i is equal to x_i and the class value is in A. occ_A is the number of instances in the training set where the class value is in A. $occ_{A,B}$ is the number of training sample whose class is either in A or B. The hyper-parameter s sets the imprecision level of the IDM and is usually equal to 1 or 2 [Walley, 1996]. In our experiments, we will set s to 1 every time NCC is involved.

We also note that when s is set to 0, the lower and upper bounds coincide, and the model then reduces to a standard NBC with a Dirichlet prior. Therefore we can easily pass from the imprecise method to the precise one by changing s to 0, and this is how we obtain the estimates of NBC in our experiments.

B Method developed by Del Coz *et al.*

[del Coz and Bahamonde, 2009] propose to derive indeterminate predictions from precise probability estimates by adapting the well-known F_β measure to do so. Their proposal results in the formula

$$F_\beta(\hat{Y}, \omega) = \frac{1+\beta^2}{\beta^2 + |\hat{Y}|} \times \mathbb{1}_{\omega \in \hat{Y}}. \tag{18}$$

In most cases, the parameter β is set to 1, hence the F_1 measure.

The method then consists in predicting the set of classes $\hat{Y}$ which minimizes the expected cost $\mathbb{E}[c_{\hat{Y}}]$ where

$$c_{\hat{Y}} : \left\{ \begin{array}{l} \Omega \to \mathbb{R}^+ \\ \omega \to 1 - F_\beta(\hat{Y}, \omega) \end{array} \right. \tag{19}$$

Del Coz *et al.* show that, for a prediction $\hat{Y}_r$ composed of r classes, the expected cost can be expressed as:

$$\Delta_r = \mathbb{E}[c_{\hat{Y}_r}] = 1 - \frac{1+\beta^2}{\beta^2 + r} \sum_{\omega \in \hat{Y}_r} p(\omega). \tag{20}$$

We can see that if an indeterminate prediction is composed of r classes, then those must be the r classes with the highest probabilities. Indeed, for a fixed number r, Eq. (20) is minimized if $\sum_{\omega \in \hat{Y}_r} p(\omega)$ is maximized. This important feature allow Del Coz *et al.* to propose a simple algorithm, linear in the number of classes, to

find the best prediction. The principle is first to sort the classes according to their posterior probabilities, so that we have a descending order $(\omega_1, \ldots, \omega_K)(K = |\Omega|)$ where, if $i < j \in [1; K]$ then $p(\omega_i) > p(\omega_j)$. An indeterminate prediction of size r is defined as $\hat{Y}_r = \{\omega_1, \ldots, \omega_r\}$; The algorithm consists in computing the sequence of values Δ_r (starting from $r = 1$ and incrementing r) and then to retain the $\hat{Y}_r$ minimizing the sequence. Note that it is not necessary to compute Δ_r for all values of r, as the algorithm stops as soon as an increase is detected in the sequence.

We can note that, due to the definition of F_β, every wrong prediction yields zero reward (or conversely, a cost of 1), which means that this method is only valid in the unitary misclassification cost setting. The problem of treating non-unitary costs is not mentioned in [del Coz and Bahamonde, 2009].

References

Abellán, J. and Masegosa, A. (2012). Imprecise Classification With Credal Decision Trees. *Intl J of Uncertainty, Fuzziness and Knowledge-Based Systems*, 20(05):763–787.

Allwein, E., Schapire, R., Singer, Y., and Kaelbling, P. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *J of Machine Learning Research*, 1:113–141.

Alonso, J., Del Coz, J., Díez, J., Luaces, O., and Bahamonde, A. (2008). Learning to predict one or more ranks in ordinal regression tasks. In *Machine Learning and Knowledge Discovery in Databases*, pages 39–54. Springer.

Bengio, S., Weston, J., and Grangier, D. (2010). Label embedding trees for large multi-class tasks. In *NIPS*, volume 23, page 3.

Bernard, J.-M. (2005). An introduction to the imprecise dirichlet model for multinomial data. *Intl J of Approximate Reasoning*, 39(2-3):123–150.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., and Warmuth, M. K. (1997). How to use expert advice. *J of the ACM (JACM)*, 44(3):427–485.

Chow, C. (1970). An optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46.

Corani, G., Antonucci, A., and De Rosa, R. (2012). Compression-based AODE classifiers. In *Eur Conference on Artificial Intelligence*, pages 264–269.

Corani, G. and Mignatti, A. (2015). Credal model averaging for classification: representing prior ignorance and expert opinions. *Intl J of Approximate Reasoning*, 56:264–277.

Corani, G. and Zaffalon, M. (2008). Credal model averaging: an extension of bayesian model averaging to imprecise probabilities. In *Machine Learning and Knowledge Discovery in Databases*, pages 257–271. Springer.

De Cooman, G. and Hermans, F. (2008). Imprecise probability trees: Bridging two theories of imprecise probability. 172:1400–1427.

del Coz, J. and Bahamonde, A. (2009). Learning nondeterministic classifiers. *J of Machine Learning Research*, 10:2273–2293.

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

Destercke, S. and Quost, B. (2011). Combining binary classifiers with imprecise probabilities. In *Proceedings of the 2011 international conference on Integrated uncertainty in knowledge modelling and decision making*, IUKM'11, pages 219–230, Berlin, Heidelberg. Springer-Verlag.

Dietterich, T. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *J of Artificial Intelligence Research*, 2:263–286.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Intl joint Conference on artificial intelligence*, volume 17, pages 973–978.

Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, pages 1022–1029.

Fox, J. (1997). *Applied regression analysis, linear models, and related methods*. Sage.

Frank, E. and Hall, M. (2001). A simple approach to ordinal classification. In *Proceedings of the 12th Eur Conference on Machine Learning*, pages 145–156. Springer-Verlag.

Frank, E. and Kramer, S. (2004). Ensembles of nested dichotomies for multi-class problems. *ICML 2004*, page 39.

Friedman (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J of the American Statistical Association*, 32(200):675–701.

Grunbaum, B., Klee, V., Perles, M. A., and Shephard, G. C. (1967). *Convex polytopes*. Springer.

Hastie, T. and Tibshirani, R. (1998). Classification by pairwise coupling. *The Annals of Statistics*, 26:451–471.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning*. Springer.
Levi, I. (1983). *The enterprise of knowledge: An essay on knowledge, credal probability, and chance*. MIT press.
Lichman, M. (2014). *UCI Machine Learning Repository.* `http://archive.ics.uci.edu/ml`.
Lorena, A. C. and De Carvalho, A. (2010). Building binary-tree-based multiclass classifiers using separability measures. *Neurocomputing*, 73(16-18):2837–2845.
Mantas, C. and Abellan, J. (2014). Credal-c4.5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with applications*, 41(10):4625–4637.
Masnadi-Shirazi, H. and Vasconcelos, N. (2010). Risk minimization, probability elicitation, and cost-sensitive svms. In *Intl Conference Machine Learning*, pages 759–766.
Nemenyi, P. (1963). *Distribution-free Multiple Comparisons*. PhD thesis, Princeton University.
Pedregosa, F. (2013). *Logistic Ordinal Regression.* `https://github.com/fabianp/minirank/tree/master/minirank`.
Rokach, L. (2006). Decomposition methodology for classification tasks: a meta decomposer framework. *Pattern Analysis and Applications*, 9(2-3):257–271.
Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39.
Troffaes, M. (2007). Decision making under uncertainty using imprecise probabilities. *Intl J of Approximate Reasoning*, 45(1):17–29.
Walley, P. (1991). *Statistical reasoning with imprecise probabilities*. Chapman and Hall.
Walley, P. (1996). *Inferences from multinomial data: learning about a bag of marbles*. JSTOR.
Wu, T., Lin, C., and Weng, R. (2004). Probability estimates for multi-class classification by pairwise coupling. *J of Machine Learning Research*, 5:975–1005.
Xu, P., Davoine, F., Zha, H., and Denoeux, T. (2015). Evidential calibration of binary svm classifiers. *Intl J of Approximate Reasoning*.
Yang, G., Destercke, S., and Masson, M.-H. (2014). Nested dichotomies with probability sets for multi-class classification. In *Eur Conference on Artificial Intelligence*.
Zaffalon, M. (2002). The naive credal classifier. *J of statistical planning and inference*, 105(1):5–21.
Zaffalon, M., Corani, G., and Maua, D. (2012). Evaluating credal classifiers by utility-discounted predictive accuracy. *Intl J of Approximate Reasoning*, 53(8):1282 – 1301.