ECAI 2014
T. Schaub et al. (Eds.)
© 2014 The Authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License.
doi:10.3233/978-1-61499-419-0-363

Nested Dichotomies with probability sets for multi-class classification

YANG Gen[1] and DESTERCKE Sébastien and MASSON Marie-Hélène

Abstract. Binary decomposition techniques transform a multi-class problem into several simpler binary problems. In such techniques, a classical issue is to ensure the consistency between the binary assessments of conditional probabilities. Nested dichotomies, which consider tree-shaped decomposition, do not suffer from this issue. Yet, a wrong probability estimate in the tree can strongly biase the results and provide wrong predictions. To overcome this issue, we consider in this paper imprecise nested dichotomies, in which binary probabilities become imprecise. We show in experiments that the approach has many advantages: it provides cautious inferences when only little information is available, and allows to make efficient computations with imprecise probabilities even when considering generic cost functions.

1 Introduction

The usual goal of machine learning algorithms is to learn, from a set of data, a model that will provide accurate predictions on new data. Most current techniques focus on achieving a better rate of *accuracy* while preserving the determinacy of predictions, even if they are based on few information. However, in some applications of machine learning (*e.g.* medical diagnosis, image recognition for intelligent vehicles, risk analysis), the reliability of predictions is as essential as their accuracy. In such cases, providing indeterminate but more reliable predictions makes sense. For example, in the problem of obstacle recognition for vehicles, it is preferable to state "I do not know" rather than predicting "no obstacles" if the available information is not sufficient to reliably say that there is an obstacle.

There are two main approaches to make indeterminate predictions in a classification problem: integrating costs of indeterminacy in the decision making [8] and considering imprecise probabilities as estimates rather than precise probabilities [6]. The former approach, close in spirit to rejection methods [3], does not really allow to differentiate between rejection due to ambiguity (almost uniform probability estimated from lots of data) and rejection due to lack of information (probability issued from little and/or imprecise data). It also tends to mix costs of errors (e.g., of predicting no obstacle when there is a pedestrian) with costs of being indeterminate (e.g., costs of partial predictions). On the other hand, the latter approach based on imprecise probabilistic estimates [21] perfectly makes the difference between indeterminacy due to ambiguity and lack of information (the less data we have, the larger the estimated probability set) and uses costs only to model error costs. However, integrating costs of errors in such methods is computationally challenging, which is an important drawback, since applications where indeterminacy and reliability are important will typically include such costs.

In this paper, we propose a classification method relying on imprecise probabilities and extending the notion of nested dichotomies [13] (a particular binary decomposition) to such models. This method has the advantage that it can make indeterminate classification, while having a computational burden similar to its precise counterpart, even when non-trivial costs are considered in the decision making.

We first introduce some basic notions of multi-class classification and establish some notations in Section 2. We then present the "nested dichotomies" technique in Section 3, before detailing in Section 4 how imprecise probabilities can be integrated in the method to provide indeterminate (set-valued) predictions. Finally, experiments provided in Section 5 show that using our approach provide cautious but informative predictions, in the sense that we add indeterminacy mainly when determinate predictions are unreliable.

2 Context and definitions

2.1 Notations

We consider the multi-class classification problem, where we want to learn the conditional probability function $p_{\mathbf{x}}(\cdot) : Y \rightarrow [0,1]$ of the class $y \in Y$ $(Y = \{\omega_1, \ldots, \omega_k\})$ given m input features $\mathbf{x} \in \mathbf{X} = X^1 \times \ldots \times X^m$. p is usually learnt from a set of data $\mathcal{D} = (\mathbf{x}_i, y_i)_{i \in [1;n]}$. For simplification purpose, we will drop the subscript $_{\mathbf{x}}$ and will denote $p_{\mathbf{x}}(y)$ by $p(y)$ when there is no risk of confusion.

For each class $y \in Y$, we assume that a cost function $c_y : Y \rightarrow \mathbb{R}$ is defined, where $c_y(y')$ is the cost of predicting y when y' is the true class. The expected cost $E_Y(c_y)$ of predicting/selecting y is then defined as follow:

$$E_Y(c_y) = \sum_{y' \in Y} p(y')c_y(y')$$

A common cost for y is the *unitary cost* such that $c_y(y') = 1$ if $y' \neq y$ and 0 otherwise. It is related to the indicator function I_y ($I_y(y') = 1$ if $y = y'$, 0 otherwise) through the equality $c_y = -I_y + 1$. By using this, we have $E_Y(I_y) = p(y) = -E_Y(c_y) + 1$.

Making prediction can also be seen as establishing a preference order $\succ$ over the classes to find the most preferred one. This order is derived from the expected cost such that $y \succ z$ (read "y is preferred to z") if the expected cost of choosing y is less than the one of z :

$$y \succ z \Leftrightarrow E_Y(c_y) < E_Y(c_z) \tag{1}$$

Since E_Y is linear, $y \succ z$ is also equivalent to :

$$y \succ z \Leftrightarrow E_Y(c_z - c_y) > 0. \tag{2}$$

[1] Heudiasyc Laboratory, Université de Technologie de Compiègne, France, email: gen.yang@hds.utc.fr

Eq. (2) can be interpreted as follows: y is preferred to z when exchanging y for z is costly (i.e., has a positive expected cost). We could note that, if c_y, c_z are unitary, this is equivalent to compare the probability values $p(y), p(z)$ ($y \succ z$ if $p(y) > p(z)$). The selected class is therefore the maximal element of the ordering $\succ$, *i.e.*, $\arg\max_{y \in Y} E_Y(c_y)$. This is this view (constructing an order $\succ$) that we will extend when using probability sets.

Example 1 *The interest of cost functions is to model the costs of making a wrong decision (i.e., making a prediction different from the truth). For example, consider the problem of obstacle recognition where a vehicle needs to recognize in situation* $\mathbf{x}$ *whether it faces a pedestrian (p), a bicycle (b) or nothing (n) (i.e.* $Y = \{p, b, n\}$*).*

As both pedestrian and bicycle are obstacles to be avoided, a confusion between p *and* b *is not very important. Predicting* p *or* b *when there is nothing becomes more costly (the vehicle makes a manoeuvre which is not necessary). Finally, predicting* n *when there is an obstacle* p *or* b *is a big mistake that could cause an accident. This kind of information can easily be expressed using non unitary cost functions. The following table provides an example of 3 cost functions modelling these information, as well as an example of their difference :*

	truth		
$c_y(y')$	$y' = p$	$y' = b$	$y' = n$
c_p	0	1	2
c_b	1	0	2
c_n	4	4	0
$c_p - c_n$	-4	-3	2
$c_b - c_n$	-3	-4	2
$c_b - c_p$	1	-1	0

With these cost functions, we have translated the fact that a confusion between a pedestrian and a bicycle has little effect, whereas a confusion with the absence of obstacle is penalizing.

2.2 Binary decomposition

Binary decomposition techniques [11] have proved to be good approaches to solve the multi-class problem (for a review of methods, see [1]). Such techniques propose to decompose the original (difficult) multi-class problem into a set of simpler and easier-to-solve binary problems. 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 $A_i, B_i \subseteq Y$ and to estimate whether a class y belong to A_i or B_i for all $i = 1, \ldots, l$ instead of directly estimating the joint $p(y)$ for each $y \in Y$. Therefore, for each pair we must solve a binary classification problem and estimate $\hat{p}(A_i \mid \{A_i, B_i\}) = \alpha_i$ and $\hat{p}(B_i \mid \{A_i, B_i\}) = 1 - \alpha_i$, using what is usually called the *base classifier*. From these conditional estimates can be derived the following constraints on the joint probability:

$$\begin{cases} \sum_{y \in A_i} \hat{p}(y) = \alpha_i \sum_{y \in A_i \cup B_i} \hat{p}(y) \ (i = 1, \ldots, l) \\ \sum_{y \in Y} \hat{p}(y) = 1 \end{cases} \tag{3}$$

A frequent problem with such a general set of estimated conditional probabilities is that the constraints (3) are most of the time inconsistent [15, 23, 10], in the sense that no feasible solution will exist. How to solve this inconsistency is not an obvious problem and there is no unique best solution, even when one allows probabilities to become imprecise [10]. A usual strategy is to find a joint probability by minimizing a given distance [15, 23] to the estimates $\hat{p}(B_i \mid \{A_i, B_i\})$. One particular type of binary decomposition does not have this problem and always provide consistent constraints: nested dichotomies [13], on which we will focus. As the constraints induced by this decomposition are ensured to be consistent, we drop the ˆ sign and will use p in the rest of the paper.

3 Nested dichotomies : how it works

The principle of nested dichotomies is to form a tree structure using the class values $y \in Y$. A nested dichotomy consists in recursively partitioning a tree node $C \subseteq Y$ into two subsets A and B, until every leaf-nodes correspond to a single class value ($card(C) = 1$). The root node is the whole set of classes Y. As shows the next example, this partitioning makes it straightforward to get the global multi-class problem probability distribution (in contrast with other decompositions [15, 23]).

Example 2 *Let us consider again the example of obstacle recognition. Figure 1 pictures a nested dichotomy tree together with the conditional probability constraints.*

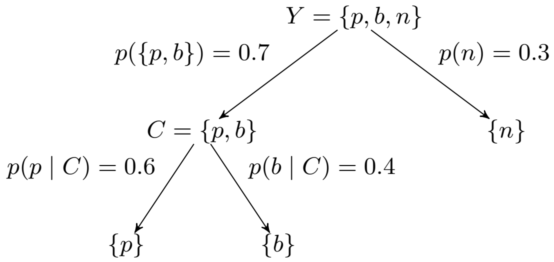

Figure 1. A probabilistic nested dichotomy

In this example, in order to estimate the original multi-class problem probability $p(y = p)$*, we need to calculate :*

$$\begin{aligned} p(y = p) &= p(y \in \{p, b\} \mid Y) \times p(y = p \mid y \in \{p, b\}) \\ &= 0.7 \times 0.6 = 0.42. \end{aligned}$$

We can see that to compute the joint probability of a given class, we just need to multiply the conditional probabilities of the branches that links this class (leaf node) to the root (Y*). The full joint can then be obtained by doing that for every class.*

3.1 The construction of the dichotomy tree

The main issue when building the tree is that there are many possible tree structures to choose from. In the cases where we have prior knowledge about the class structure (such as in ordinal classification [17]), the nested dichotomies are very adapted as they can naturally account for this additional information.

When there is no such prior knowledge, there are two ways to deal with this issue : one is to use an ensemble of dichotomy structures [14]. We will not consider ensemble method in this work, as our goal is to study the extension of dichotomy tree to imprecise probabilities. Ensembling over such models is left for future works.

Another approach when no prior knowledge is available, is to use statistics or data mining on the training dataset. [19] reviews several ways to build binary tree using separability measures. The basic idea of such techniques is to group classes according to their *statistical similarity*, in order to build binary problems whose subsets of classes are well separated. A commonly used approach (retained in this paper) is to build a $k \times k$ distance matrix $\mathbf{M}$ between every pair ω_i, ω_j of classes and then to use hierarchical clustering techniques to obtain the tree. The next matrix

$\mathbf{M}$	p	b	n
p	0	2	5
b	2	0	6
n	5	6	0

illustrates a distance matrix for our obstacle recognition example, where n is further away from the other classes, suggesting that p, b should be kept together.

3.2 Classification with nested dichotomies

Let us now detail how prediction and inferences can be obtained using conditional probabilities estimated for each partition (using a base probabilistic classifier). Note that for a given node C partitioned into A, B, we have $p(A \mid C) = 1 - p(B \mid C)$ by duality.

The inferences in nested dichotomies are made using the expected costs defined in Section 2.1. Assume we have a split $\{A, B\}$ of a node C, and a real-valued cost function $c : \{A, B\} \to \mathbb{R}$ defined on $\{A, B\}$. We can compute the (local) expectation associated with the node C by :

$$E_C(c) = E_{p(|C)}(c) = p(A \mid C)c(A) + p(B \mid C)c(B). \quad (4)$$

Now, if we start from a cost function $c : Y \to \mathbb{R}$ defined on the classes (i.e. leaf nodes), then using the law of iterated expectation with nested conditioning sets [4, P. 449], we can apply Eq. (4) recursively from the leafs to the root Y, in order to get the global expectation. This is because we can view any expected cost E_C associated with a node C as a cost function $c(y) = E(| \ C = y)$ on C.

Example 3 *In Example 2, to decide between "pedestrian" and "nothing" with the tree pictured in Figure 1, we just need to compute the expected cost $E_{\{p,b,n\}}(c_n - c_p)$ as recalled in Section 2.1. Local expectation computations are noted under the nodes of Figure 2. Finally we have :*

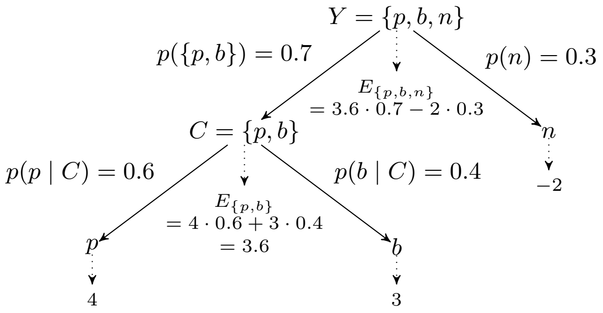

Figure 2. Expectation computation for $c_n - c_p$.

As shown in the Figure 2, we have

$$E_{\{p,b,n\}}(c_n - c_p) = 0.7 \cdot 3.6 - 0.3 \cdot 2 = 1.92 > 0$$

Therefore we have $p \succ n$, as choosing n instead of p is costly.

4 Imprecise nested dichotomies

Clearly the accuracy of a nested dichotomy will heavily depends on the tree structure. Indeed, a wrong estimate of one conditional probability may biase the whole structure, leading to unreliable and potentially wrong inferences. Therefore it seems interesting to replace the precise estimates by interval-valued ones, the width of which reflects the lack of information.

Such intervals define an imprecise probabilistic classifier that we study in this section. We will see that one advantage of this classifier, in contrast with other imprecise probabilistic classifiers [24], is that it can handle generic and unitary costs with the same computational complexity. Moreover, this complexity is of the same order as its precise counterpart. In the rest of this section, we explain how to make indeterminate predictions from such imprecise nested dichotomies.

4.1 Generalization to imprecise probability

We now allow every local model to be imprecise, that is to each node C can be associated an interval $[\underline{p}(A \mid C); \overline{p}(A \mid C)]$, precise nested dichotomies being retrieved when $\underline{p}(A \mid C) = \overline{p}(A \mid C)$ for every node C. By duality of the imprecise probabilities [21, Sec.2.7.4.], we have $\underline{p}(A \mid C) = 1 - \overline{p}(B \mid C)$ and $\overline{p}(A \mid C) = 1 - \underline{p}(B \mid C)$. Such an imprecise nested dichotomy can be associated to a set $\mathcal{P}$ of joint probabilities, obtained by considering all precise selection $p(A \mid C) \in [\underline{p}(A \mid C); \overline{p}(A \mid C)]$ for each node C. This set can then be associated with lower and upper expectations $[\underline{E}_Y(c); \overline{E}_Y(c)]$ such that

$$\underline{E}_Y(c) = \min_{p \in \mathcal{P}} E_Y(c) = \min_{p \in \mathcal{P}} \sum_{y \in Y} p(y)c(y),$$

$$\overline{E}_Y(c) = \max_{p \in \mathcal{P}} E_Y(c) = \max_{p \in \mathcal{P}} \sum_{y \in Y} p(y)c(y).$$

Given a cost function c, computing $\underline{E}$ and $\overline{E}$ can be done as in the precise case shown in section 3.2. For instance, the lower local expected cost of a node C becomes :

$$\underline{E}_C(c) = \min \begin{pmatrix} \underline{p}(A \mid C)c(A) + \overline{p}(B \mid C)c(B); \\ \overline{p}(A \mid C)c(A) + \underline{p}(B \mid C)c(B) \end{pmatrix} \quad (5)$$

Similarly to Section 3.2, the law of iterated expectation can be applied to compute $\underline{E}_Y$ and $\overline{E}_Y$ [21, Sec. 6.3.5] [7] recursively by going from the leaves to the root. The upper expected cost $\overline{E}_Y$ is obtained by replacing $\min$ by $\max$ in (5) since we have the duality $\underline{E}(c) = -\overline{E}(-c)$.

Moreover, as for the precise version, lower/upper probabilities of a class correspond to $\underline{p}(\omega) = \underline{E}(I_\omega)$ and $\overline{p}(\omega) = \overline{E}(I_\omega)$.

Example 4 *We consider Example 3 in the imprecise probabilities framework : now all conditional probabilities estimated by the local base classifiers are interval-valued (see Figure 3). Let us see how the expected cost $\underline{E}_Y(c_b - c_p)$ is calculated :*

Similarly than in the precise case, by using (5) and knowing that :

$$\underline{E}_{\{p,b\}}(c_b - c_p) = \min\big(0.6 - 0.4; 0.7 - 0.3\big) = 0.2$$

We have :

$$\begin{aligned} & \underline{E}_{\{p,b,n\}}(c_b - c_p) \\ = \ & \min\big(0.2 \cdot 0.8 + 0 \cdot 0.2; 0.2 \cdot 0.5 + 0 \cdot 0.5\big) \\ = \ & 0.1 > 0 \end{aligned}$$

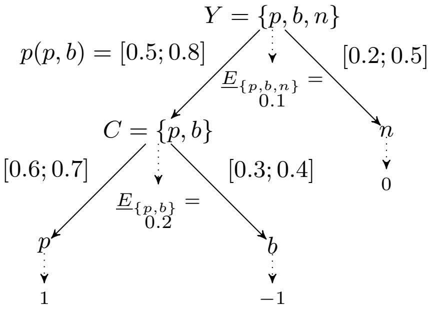

Figure 3. Example of nested dichotomies with imprecise probabilities

As the example shows, computing with imprecise nested dichotomies is as easy as with precise one: lower and upper estimates are still multiplicative along a branch. This is in contrast with other imprecise models, where adding imprecision makes inferences computationally costly.

4.2 Decision making with imprecise nested dichotomies

Since $\underline{E}_Y$ and $\overline{E}_Y$ are not linear, Eqs. (1) and (2) used as decision criteria in the precise case, are no longer equivalent in the imprecise one.

Actually, there are several ways to extend the classical expected cost criterion to imprecise probabilities [20]. They can be grouped in two groups depending on the type of decision : some rules give a unique output class (*e.g.* maximin), other may give a set of possible optimal classes (*e.g.* interval dominance, maximality). In our work, we concentrate on the second one, as we are interested in indeterminate but reliable predictions. These rules consist in constructing a partial order $\succ$ over classes and then to select the maximal ones in this order.

Definition 1 (Maximality) *Under the maximality criterion,*

$$\omega_i \succ_{\mathcal{M}} \omega_j \Leftrightarrow \underline{E}(c_{\omega_j} - c_{\omega_i}) > 0. \qquad (6)$$

This criterion extends Eq. (2). Eq. (6) can be interpreted as follows: ω_i is preferred to ω_j if exchanging ω_i for ω_j has a positive lower expected cost. The (possibly) imprecise decision $Y_{\mathcal{M}}$ obtained from this criterion is

$$Y_{\mathcal{M}} = \Big\{\omega_i \in Y \mid \nexists \omega_j : \omega_i \succ_{\mathcal{M}} \omega_j\Big\}.$$

In Example 4, we have that $p \succ_{\mathcal{M}} b$. Note that obtaining the order $\succ$ requires to perform $k(k-1)$ computations (one for each pair). Also, while maximality has strong theoretical justifications [21, Sec. 3.9.], other decision criteria such as interval dominance may be preferred if computational time is an important issue (e.g., when the number of classes is high).

Definition 2 (Interval dominance) *Under interval dominance criterion,*

$$\omega_i \succ_{\mathcal{ID}} \omega_j \Leftrightarrow \overline{E}(c_{\omega_i}) < \underline{E}(c_{\omega_j}). \qquad (7)$$

The interval dominance criterion extends Eq. (1). The (possibly) imprecise decision $Y_{\mathcal{M}}$ obtained from this criterion is

$$Y_{\mathcal{ID}} = \Big\{\omega_i \in Y \mid \nexists \omega_j : \omega_i \succ_{\mathcal{ID}} \omega_j\Big\}.$$

Using this rule as our prediction criterion requires to compare lower expectation bounds of every class cost with the minimal upper bound, thus requiring only $2k$ computations.

It is known that $y \succ_{\mathcal{ID}} z$ implies $y \succ_{\mathcal{M}} z$, but not the reverse [20], hence interval dominance is more conservative than maximality. For instance, the tree pictured in Figure 3 is such that $\overline{E}_{p,b,n}(c_p) = 1.2$ and $\underline{E}_{p,b,n}(c_b) = 0.88$, so we do not have $p \succ_{\mathcal{ID}} b$. This is due to the fact that probabilities used within $\mathcal{P}$ to reach upper and lower expectations are most of the time different, hence interval dominance comparisons are done for different probabilities, while maximality comparisons are not. The latter makes more sense in our framework, as we assume that there is one true but ill-known probability. Therefore, we will use the maximality criterion in our experiments.

5 Experiments

In this section, our method is tested on 14 datasets of the UCI machine learning repository [2], whose details are given in Table 1. As base classifiers, we use the common naive Bayes classifier (NBC) and its imprecise counterpart, the naive credal classifier (NCC), which despite their simplicity provide good accuracies. For details on the NCC, we refer to [24]. This is sufficient in the present study, in which our goal is to compare the imprecise nested dichotomies to their precise and multi-class counterparts.

Name	(C)ont/(D)isc features	# instances	# classes
balance-scale	D	625	3
car	D	1728	4
lymph	D	148	4
LEV	D	1000	5
nursery	D	12960	5
zoo	D	101	7
soybean	D	562	15
iris	C	150	3
wine	C	178	3
grub-damage	C	155	4
page-blocks	C	5473	5
glass	C	214	6
ecoli	C	336	8
pendigits	C	10992	10

Table 1. data set details

5.1 Experimental set-ups

5.1.1 Discretization

As NBC and NCCs cannot handle continuous variables natively, continuous features in data sets (data sets with C in second column of Table 1) were discretized. We chose to discretize all continuous features by dividing their domain in 8 intervals of equal width. We did not use a supervised discretization method such as Fayyad and Irani [12], as the classes changes between the initial multi-class problem and each binary sub-problem.

5.1.2 Class distance

To apply Section 3.1 approach, we need to define a distance to establish the distance matrix used in the hierarchical clustering. Let us denote by $p_{\omega_i}(X^j = x) = {}^{occ_i^j(x)}/_{occ_i}$ the empirical probability that feature X^j takes value x given that the class is ω_i, with $occ_i^j(x)$ the number of samples $(\mathbf{x}, y)$ of data set D for which $\mathbf{x}^j = x$ when $y = \omega_i$, and occ_i the number of samples for which $y = \omega_i$. Once these probabilities have been estimated (note that we have to estimate them to build the naive classifiers anyway), we define distance $M_{i,i'}$ between classes ω_i and ω_i' as

$$\forall \omega_i, \omega_i' \in Y, M_{i,i'} = \sum_{j \in [1;m]} H(p_{\omega_i}(X^j), p_{\omega_i'}(X^j)),$$

where H is the Hellinger distance. H is defined for two probability distributions P and Q as $H(P,Q) = \sqrt{1 - BC(P,Q)}$, where BC is the Bhattacharyya coefficient :

$$BC(P,Q) = \sum_{x \in X} \sqrt{P(x)Q(x)}.$$

There are other distances between probability distributions we could use [5], yet our goal is not to make a comparative study of those distances, and we will see that the Hellinger distance provides good results.

Once the distance is defined, we use different hierarchical clustering linkage criteria (maximum, minimum, average [16, Sec. 14.3], Ward [22]) to build the tree, and select the one yielding the best predictive accuracy on the learning dataset.

5.2 Tests and results

This section summarizes the results of the test. Each result is obtained by a 10-fold cross validation on the (possibly discretized) data set. As we can make indeterminate predictions, we will use performance measures adapted to the comparison of indeterminate and determinate classifiers. We will also concentrate on unitary costs, as such measures are only valid for unitary costs and as the used benchmark data sets do not come with pre-defined costs.

5.2.1 *Performance comparison*

In order to fairly compare precise methods and imprecise ones, we need to evaluate both the precision and the accuracy at the same time. The idea is to penalize the imprecise prediction according to its imprecision level. Hence, we use a utility-discounted accuracy (u_{65}) introduced by [25]. Let $\mathbf{x}_i, y_i (i = [1;n])$ be the set of test data and Y_i our (possibly imprecise) predictions, then u_{65} is

$$u_{65} = \frac{1}{n} \sum_{i=1}^{n} -1.2d_i^2 + 2.2d_i,$$

where $d_i = {}^{\mathbb{1}_{Y_i}(y_i)}/_{|Y_i|}$ is the discounted accuracy ($\mathbb{1}_{Y_i}(y_i)$ is the indicator functions that has value 1 if $y_i \in Y_i$ and 0 otherwise). Compared to the discounted accuracy, u_{65} accounts for the fact that making the cautious statements that we are in Y_i (without saying more) is preferable to give as precise prediction a purely random guess within Y_i (see [25] for details). Also, u_{65} is less in favour of indeterminate classifiers than the F_1 measure proposed by Del Coz *et al.* [8], meaning that we remain quite fair to the determinate classifier.

In the experiments whose results are given in Table 2, we used three methods: nested dichotomies with the naive Bayes classifier (ND+NBC), nested dichotomies with the naive credal classifier (ND+NCC) and the naive credal classifier (NCC). This allows us to compare the precise and imprecise dichotomy, as well as the imprecise dichotomy with its multi-class counterpart.

	u_{65} (rank)		
Data	ND + NBC	NCC	ND + NCC
balance-scale	91.68% (1)	90.72% (3)	90.98% (2)
car	84.09% (3)	88.11% (1)	86.72% (2)
lymph	80.41% (1)	59.02% (3)	67.58% (2)
LEV	58.20% (3)	61.50% (1)	60.69% (2)
nursery	90.58% (3)	91.02% (1)	90.99% (2)
zoo	96.04% (1)	77.96% (3)	84.85% (2)
soybean	81.85% (2)	84.12% (1)	80.68% (3)
iris	94.67% (3)	95.47% (2)	95.60% (1)
wine	96.63% (1)	94.72% (3)	95.06% (2)
grub-damage	49.68% (3)	52.90% (2)	53.81% (1)
page-blocks	91.36% (3)	92.01% (1)	91.69% (2)
glass	60.75% (2)	60.51% (3)	65.43% (1)
ecoli	76.49% (1)	58.11% (3)	74.83% (2)
pendigits	70.25% (3)	87.3% (1)	71.61% (2)
average rank	2.14	2	1.86

Table 2. Comparison of discounted accuracy (u65) for the methods ND+NBC, NCC and ND+NCC.

First, we can notice that our imprecise classifier yields the best average rank over the 14 data sets. However, using Demsar's approach [9] by applying the Friedman test on the ranks of algorithm performances for each dataset, we find a value of 0.57 for the chi-squared test with 2 degree of freedom, so the p-value is 0.75 and we cannot reject the null hypothesis, meaning that all methods have comparable performances in terms of accuracy. Yet, our approach has several advantages that we now detail.

5.2.2 *Gain of accuracy on indeterminate predictions*

The main goal of indeterminate 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 4 displays, on the instances where the ND+NCC made indeterminate predictions, the percentage of times the true class is within the prediction, both for the ND+NBC and ND+NCC.

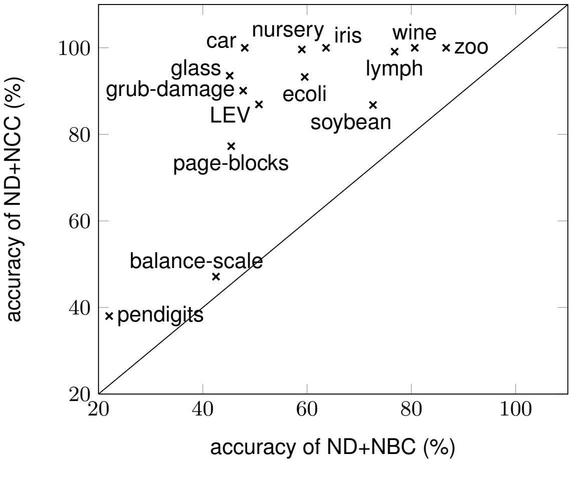

Figure 4. Accuracy of the methods "ND+NBC" and "ND+NCC" when a imprecise prediction is made by "ND+NCC".

We observe an important gain in accuracy on indeterminate predictions for all data sets, except on balance-scale for which the gain is lower. While an imprecise classifier will always be more accurate than its precise counterpart, we can notice that on those instances where the imprecise classifier is indeterminate, the accuracy of its precise counterpart is usually much lower than the average obtained for the whole data set displayed in Table 2 (e.g., page-blocks drops from 90% to 50%). This clearly shows that using imprecise estimates in the nested dichotomies is sensible, as indeterminate predictions are made on instances that are hard to classify for the precise method.

5.2.3 *Comparison of indeterminacy with NCC*

Figure 5 displays, for the ND+NCC and NCC, the percentage of indeterminate predictions. We can see that for all data sets, the ND+NCC method is more determinate than NCC, while keeping comparable performances (see Tab. 2). While the gain remains marginal in many data set, it can nevertheless be significant for some data sets (ecoli, glass, lymph, grub-damage, soybean, zoo).

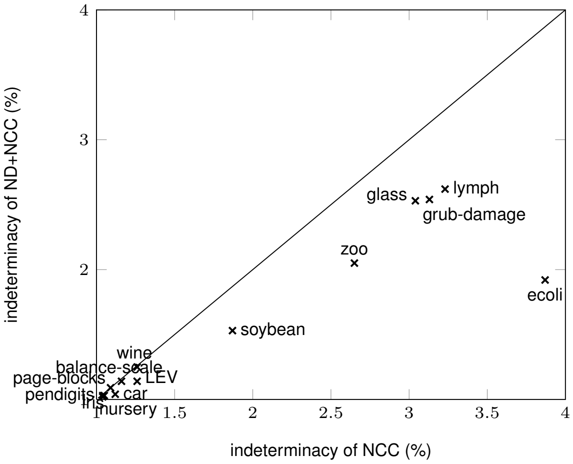

Figure 5. Percentage of set-valued predictions made by NCC and ND+NCC.

6 Conclusions

In this paper, we have introduced the notion of imprecise nested dichotomies and how to perform efficient inferences from them. Our experiments show that nested dichotomies have a very interesting behaviour: they allow to be cautious on hard to predict instances for precise classifiers, while being more determinate than imprecise multi-class approaches. More importantly, they remain efficient even when integrating error costs in the inferences, while other imprecise probabilistic classifiers tipycally necessitate more complex computations to do that.

In future works, we intend to explore other approaches to build the dichotomy tree, as well as the application of ensemble approaches in the imprecise context. We would also like to explore to which measure the efficiency of imprecise nested dichotomies can be improved, e.g., by suing results on label trees [18]. Finally, we intend to apply nested dichotomies on structured classes (e.g., ordinal regression [17]).

ACKNOWLEDGEMENTS

This work was carried out in the framework of the Labex MS2T, which was funded by the French National Agency for Research (Reference ANR-11-IDEX-0004-02).

REFERENCES

[1] M. Aly, 'Survey on multiclass classification methods', Technical report, California Institute of Technology, (November 2005).

[2] K. Bache and M. Lichman, *UCI Machine Learning Repository*. `http://archive.ics.uci.edu/ml`, (2014).

[3] P.L. Bartlett and M.H. Wegkamp, 'Classification with a reject option using a hinge loss', *The Journal of Machine Learning Research*, **9**, 1823–1840, (2008).

[4] P. Billingsley, *Probability and measure*, John Wiley & Sons.

[5] S.H. Cha, 'Comprehensive survey on distance/similarity measures between probability density functions', *International Journal of Mathematical Models and Methods in Applied Sciences*, **1**(4), 300–307, (2007).

[6] G. Corani, A. Antonucci, and R. De Rosa, 'Compression-based aode classifiers.', in *ECAI*, pp. 264–269, (2012).

[7] G. De Cooman and F. Hermans, 'Imprecise probability trees: Bridging two theories of imprecise probability', **172**, 14001427.

[8] J.J. Del Coz, J. Dez, and A. Bahamonde, 'Learning nondeterministic classifiers', *Journal of Machine Learning Research*, **10**, 2273–2293, (2009).

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

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

[11] T.G. Dietterich and G. Bakiri, 'Solving multiclass learning problems via error-correcting output codes', *Journal of Artificial Intelligence Research*, **2**, 263–286, (1995).

[12] U.M. Fayyad and K.B. Irani, 'Multi-interval discretization of continuous-valued attributes for classification learning', in *IJCAI*, pp. 1022–1029, (1993).

[13] J. Fox, *Applied Regression Analysis, Linear Models, and Related Methods*, Sage.

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

[15] T. Hastie and R. Tibshirani, 'Classification by pairwise coupling', *The Annals of Statistics*, **26**, 451–471, (1998).

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, volume 2, Springer, 2009.

[17] J.C. Huhn and E. Hullermeier, 'Is an ordinal class structure useful in classifier learning?', *International Journal of Data Mining, Modelling and Management*, **1**(1), 45–67, (2008).

[18] Baoyuan Liu, Fereshteh Sadeghi, Marshall Tappen, Ohad Shamir, and Ce Liu, 'Probabilistic label trees for efficient large scale image classification', in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 843–850. IEEE, (2013).

[19] A. C. Lorena and A. De Carvalho, 'Building binary-tree-based multiclass classifiers using separability measures', *Neurocomputing*, **73**(16-18), 2837–2845, (October 2010).

[20] M. Troffaes, 'Decision making under uncertainty using imprecise probabilities', *International Journal of Approximate Reasoning*, **45**(1), 17–29, (May 2007).

[21] P. Walley, *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall.

[22] J.H. Ward Jr, 'Hierarchical grouping to optimize an objective function', *Journal of the American statistical association*, **58**(301), 236–244, (1963).

[23] T.F. Wu, C.J. Lin, and R.C. Weng, 'Probability estimates for multi-class classification by pairwise coupling', *Journal of Machine Learning Research*, **5**, 975–1005, (2004).

[24] M. Zaffalon, 'The naive credal classifier', *Journal of statistical planning and inference*, **105**(1), 5–21, (2002).

[25] M. Zaffalon, G. Corani, and D. Mau, 'Evaluating credal classifiers by utility-discounted predictive accuracy', *International Journal of Approximate Reasoning*, **53**(8), 1282 – 1301, (2012).