Correcting Binary Imprecise Classifiers: Local vs Global Approach

Sébastien Destercke, Benjamin Quost

▶ **To cite this version:**

Sébastien Destercke, Benjamin Quost. Correcting Binary Imprecise Classifiers: Local vs Global Approach. Scalable Uncertainty Management, Sep 2012, Germany. pp.299-310. hal-00745589

HAL Id: hal-00745589
https://hal.science/hal-00745589v1

Submitted on 26 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Correcting binary imprecise classifiers: local vs global approach

Sébastien Destercke and Benjamin Quost

HEUDIASYC, 6599. Université de Technologie de Compiègne. Centre de Recherches de Royallieu. 60205 COMPIEGNE, France
desterc,quostben@hds.utc.fr

Abstract. This paper proposes a simple strategy for combining binary classifiers with imprecise probabilities as outputs. Our combination strategy consists in computing a set of probability distributions by solving an optimization problem whose constraints depend on the classifiers outputs. However, the classifiers may provide assessments that are jointly incoherent, in which case the set of probability distributions satisfying all the constraints is empty. We study different correction strategies for restoring this consistency, by relaxing the constraints of the optimization problem so that it becomes feasible. In particular, we propose and compare a global strategy, where all constraints are relaxed to the same level, to a local strategy, where some constraints may be relaxed more than others. The local discounting strategy proves to give very good results compared both to single classifier approaches and to classifier combination schemes using a global correction scheme.

1 Introduction

In complex multi-class classification problems, a popular approach consists in decomposing the initial problem into several simpler problems, training classifiers on each of these sub-problems, and then combining their results. The advantages are twofold: the sub-problems obtained are generally easier to solve and thus may be addressed with simpler classification algorithms, and their combination may yield better results than using a single classification algorithm.

In this paper, we consider a classical decomposition strategy where each simple problem is binary; then, each classifier is trained to separate two subsets of classes from each other. When the binary classifiers return conditional probabilities estimating whether an instance belongs to a given class subset or not, these conditional probabilities are seldom consistent, due to the fact that they are only approximations of the (admittedly) true but unknown conditional probabilities. Usually, this inconsistency problem is tackled by considering some optimization problem whose solution is a consistent probability whose conditional probabilities are close to each of the estimated ones [5, 9]. This consistent probability is then considered as the final predictive model.

Imprecise probabilities are concerned with the cases where the available information is not sufficient (or too conflicting) to identify a single probability,

and are therefore well adapted to the problem mentioned above. Due to their robustness, imprecise probabilistic models appear particularly interesting in those cases where some classes are difficult to separate, where some classes are poorly represented in the training set or when the data are very noisy. In a previous work [4], we proposed an alternative solution to classifier combination using imprecise probability theory [8]. In this framework, binary classifiers return lower and upper bounds instead of a single evaluation. The case of precise outputs is retrieved when lower and upper bounds coincide.

As even imprecise outputs can turn out to be inconsistent, we initially proposed to apply a global discounting factor (found through a heuristic) to the classifiers. In this paper, we reformulate the problem so that discounting factors can be found by the means of efficient linear programming techniques. This also allows us to easily affect a discounting factor specifically to each classifier, thus adopting a local correction approach. In Section 2, we remind the necessary elements about imprecise probabilities and their use in binary classifiers combination. Section 3 then describes and discusses our discounting strategies, both the global and local one. Finally, we compare in Section 4 the two strategies for the special case of one-vs-one classifiers on several classical real data sets.

2 Imprecise probability: a short introduction

Let $\mathcal{X} = \{x_1, \ldots, x_M\}$ be a finite space of M elements describing the possible values of (ill-known) variables (here, $\mathcal{X}$ represents the set of classes of an instance). In imprecise probability theory, the partial knowledge about the actual value of a variable X is described by a convex set of probabilities $\mathcal{P}$, often called *credal set* [6].

2.1 Expectation and probability bounds

A classical way to describe this set consists in providing a set of linear constraints restricting the set of possible probabilities in $\mathcal{P}$ (Walley's lower previsions [8] correspond to bounds of such constraints). Let $\mathcal{L}(\mathcal{X})$ denote the set of all real-valued bounded functions over $\mathcal{X}$, and let $\mathcal{K} \subseteq \mathcal{L}(\mathcal{X})$. Provided $\mathcal{K}$ is not empty, one can compute expectation and probability bounds on a function $f \in \mathcal{K}$.

When one starts from some lower bound $\underline{E} : \mathcal{K} \to \mathbb{R}$, it is possible to associate to it a (convex) set $\mathcal{P}(\underline{E})$ of probabilities such that

$$\mathcal{P}(\underline{E}) = \{p \in \mathbb{P}_{\mathcal{X}} | E(f) \geq \underline{E}(f) \text{ for all } f \in \mathcal{K}\}, \tag{1}$$

where $\mathbb{P}_{\mathcal{X}}$ denotes the set of all probability masses over $\mathcal{X}$.

Alternatively, one can start from a given set $\mathcal{P}$ and compute the lower expectation $\underline{E} : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$ and upper expectation $\overline{E} : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$ such that

$$\overline{E}(f) = \sup_{p \in \mathcal{P}} E(f) \quad \text{and} \quad \underline{E}(f) = \inf_{p \in \mathcal{P}} E(f),$$

These functions are dual, in the sense that $\overline{E}(f) = -\underline{E}(-f)$.

In Walley's terminology [8], $\underline{E}$ is said to *avoid sure loss* iff $\mathcal{P}(\underline{E}) \neq \emptyset$, and to be *coherent* iff for any $f \in \mathcal{K}$ we have $\underline{E}(f) = \inf_{p \in \mathcal{P}(\underline{E})} E(f)$, i.e. $\underline{E}$ is the lower envelope of $\mathcal{P}(\underline{E})$.

Lower and upper probabilities of an event $A \subseteq \mathcal{X}$ correspond to expectation bounds over the indicator function $\mathbf{1}_{(A)}$ (with $\mathbf{1}_{(A)}(x) = 1$ if $x \in A$, and 0 otherwise). When no confusion is possible, we will denote them $\underline{P}(A)$ and $\overline{P}(A)$ and they are computed as

$$\underline{P}(A) = \inf_{P \in \mathcal{P}} P(A) \text{ and } \overline{P}(A) = \sup_{P \in \mathcal{P}} P(A).$$

2.2 Imprecise probabilities and binary classifiers

The basic task of classification is to predict the class or output value x of an object knowing some of its characteristics or input values $y \in \mathcal{Y}$, with $\mathcal{Y}$ the input feature space. Usually, it is assumed that to a given input y correspond a probability mass $p(x|y)$ modeling the class distribution, knowing that the instance y has been observed. Then, classifying the instance amounts to estimating $p(x|y)$ as accurately as possible from a limited set of labeled (training) samples. A binary classifier on a set of classes $\mathcal{X}$ aims at predicting whether an instance class belongs to a subset $A \subseteq \mathcal{X}$ or to a (disjoint) subset $B \subseteq \mathcal{X}$ (i.e., $A \cap B = \emptyset$). For probabilistic classifiers, the prediction takes the form of an estimation of the conditional probability $P(A|A \cup B, y)$ that the instance belongs to A (notice that $P(B|A \cup B, y) = 1 - P(A|A \cup B, y)$ by duality).

In the case of imprecise classifiers, the prediction may be expressed as a set of conditional probabilities, expressed for example as a pair of values bounding $P(A|A \cup B)$ [1]. Let us denote by α_j, β_j the bounds provided by the j^{th} classifier:

$$\alpha_j \leq P(A_j|A_j \cup B_j) \leq \beta_j \tag{2}$$

and, by complementation, we have

$$1 - \beta_j \leq P(B_j|A_j \cup B_j) \leq 1 - \alpha_j. \tag{3}$$

Combining binary classifiers then consists in defining a set $\mathcal{P}$ of probability distributions over $\mathcal{X}$ compatible with the available set of conditional assessments. To get a joint credal set from these constraints, we will turn them into linear constraints over unconditional probabilities. Assuming that $P(A_j \cup B_j) > 0$, we first transform Equations (2) and (3) into

$$\alpha_j \leq \frac{P(A_j)}{P(A_j \cup B_j)} \leq \beta_j \text{ and } 1 - \beta_j \leq \frac{P(B_j)}{P(A_j \cup B_j)} \leq 1 - \alpha_j.$$

These two equations can be transformed into two linear constraints over unconditional probabilities:

$$\frac{\alpha_j}{1 - \alpha_j} P(B_j) \leq P(A_j) \text{ and } P(A_i) \leq \frac{\beta_j}{1 - \beta_j} P(B_j),$$

[1] From now on, we will drop the y in the conditional statements, as the combination always concerns a unique instance which input features remain the same.

or equivalently

$$0 \leq (1-\alpha_j) \sum_{x_i \in A_j} p_i - \alpha_j \sum_{x_i \in B_j} p_i, \tag{4}$$

$$0 \leq \beta_j \sum_{x_i \in B_j} p_i - (1-\beta_j) \sum_{x_i \in A_j} p_i, \tag{5}$$

where $p_i := p(x_i)$. Such constraints define the set of probability distributions that are compatible with the classifier outputs. Then, the probability bounds on this set may be retrieved by solving a linear optimization problem under Constraints (4) and (5), for all classifiers. Note that the number of constraints grows linearly with the number N of classifiers, while the number of variables is equal to the number M of classes. As the quantity of classifiers usually remains limited (between M and M^2), the linear optimization problem can be efficiently solved using modern optimisation techniques.

Example 1. Let us assume that $N = 3$ classifiers provided the following outputs:

$$P(\{x_1\}|\{x_1, x_2\}) \in [0.1, 1/3],$$
$$P(\{x_1\}|\{x_1, x_3\}) \in [1/6, 0.4],$$
$$P(\{x_2\}|\{x_2, x_3\}) \in [2/3, 0.8].$$

These constraints on conditional probabilities may be transformed into the following constraints over (unconditional) probabilities p_1, p_2, and p_3:

$$1/9 p_2 \leq p_1 \leq 1/2 p_2,\ 1/5 p_3 \leq p_1 \leq 2/3 p_3,\ 2p_3 \leq p_2 \leq 4p_3,$$

Note that the induced set of probability distributions is not empty, since $p_1 = 0.1, p_2 = 0.6$ and $p_3 = 0.3$ is a feasible solution. Getting the minimal/maximal probabilities for each class then comes down to solve 6 optimization problems (i.e., minimising and maximising each of the unconditional probabilities p_i, under the constraints mentioned above), which yields

$$p_1 \in [0.067, 0.182] \quad p_2 \in [0.545, 0.735] \quad p_3 \in [0.176, 0.31].$$

Here, we can safely classify the instance into x_2. □

Note that, in some cases, the classifiers may provide outputs that are not consistent. This is particularly the case when the classifiers are trained from distinct (non-overlapping) training sets, or when some of them provide erroneous information. Then, $\mathcal{P} = \emptyset$. A solution may still be found provided by (some of) the constraints be relaxed in order to restore the system consistency.

2.3 Vacuous mixture as discounting operator

In some situations, it may be desirable to revise the information provided by a source of information, in particular when the source is known to be unreliable

to some extent. Then, the knowledge induced by the source may be weakened according to this degree of unreliability. In most uncertainty theories, this so-called discounting operation consists in combining the original information with a piece of information representing ignorance through a convex combination.

In imprecise probability theory, the piece of information representing ignorance is the *vacuous* lower expectation $\underline{E}_{\inf}$, defined such that for any $f \in \mathcal{L}(\mathcal{X})$,

$$\underline{E}_{\inf}(f) = \inf_{x \in \mathcal{X}} f(x).$$

Given a state of knowledge represented by a lower expectation $\underline{E}$ on $\mathcal{K}$, the ϵ-discounted lower expectation $\underline{E}^{\epsilon}$ for any $f \in \mathcal{K}$ is

$$\underline{E}^{\epsilon}(f) = (1-\epsilon)\underline{E}(f) + \epsilon \underline{E}_{\inf} \tag{6}$$

with $\epsilon \in [0,1]$. We may interpret $\underline{E}^{\epsilon}$ as a compromise between the information $\underline{E}$ (which is reliable with a probability $1-\epsilon$) and ignorance. Note that we retrieve $\underline{E}$ when the source is fully reliable ($\epsilon = 0$), and ignorance when it cannot be trusted ($\epsilon = 1$).

2.4 Decision rules

Imprecise probability theory offers many ways to make a decision about the possible class of an object [7]. Roughly speaking, classical decision based on maximal expected value can be extended in two ways: the decision rule may result in choosing a single class or in a set of possible (optimal) classes. We will consider the maximin rule, which is of the former type, and the maximality rule, of the latter type.

First, let us remind that for any $x_i \in \mathcal{X}$, the lower and upper probabilities $\underline{P}(\{x_i\}), \overline{P}(\{x_i\})$ are given by the solutions of the constrained optimisation problem

$$\underline{P}(\{x_i\}) = \min p_i \quad \text{and} \quad \overline{P}(\{x_i\}) = \max p_i$$

under the Constraints (4)–(5), and the additional constraints $\sum_{x_i \in \mathcal{X}} p_i = 1$, $p_i > 0$. Then, the maximin decision rule amounts to classify the instance into class $\widehat{x}$ such that

$$\widehat{x} := \arg \min_{x_i \in \mathcal{X}} \underline{P}(\{x_i\}).$$

Using this rule requires to solve M linear systems with $2N + M + 1$ constraints and to achieve M comparisons.

The maximality rule follows a pairwise comparison approach: a class is considered as possible if it is not dominated by another one. Under the maximality rule, a class x_i is said to dominate x_j, written $x_i \succ_M x_j$, if $\underline{E}(f_{i \to j}) > 0$ with $f_{i \to j}(x_i) = 1$, $f_{i \to j}(x_j) = -1$ and $f_{i \to j}(x) = 0$ for any other element $x \in \mathcal{X}$. The set of optimal classes obtained by this rule is then

$$\widehat{X} := \{x_i \in \mathcal{X} |\ \nexists x_j \text{ s.t. } x_j \succ_M x_i\}.$$

This rule has been justified (and championed) by Walley [8]. Note that finding $\widehat{X}$ requires at most to solve $M^2 - M$ linear programs (one for each pair of classes). Using the maximality rule may seem computationally expensive; however, its computation is easier in a binary framework, as shows the next property.

Proposition 1. $0.5 < P(x_i|x_i \cup x_j) \Rightarrow x_j \notin \widehat{X}$

Proof. If $0.5 < P(x_i|x_i \cup x_j)$, then $p_i > p_j$ according to Equation (4). Assuming constraints (4) and (5) are feasible, i.e. induce a non-empty set $\mathcal{P}$, computing $\underline{E}(f_{i \to j}) > 0$ comes down to solve the following optimisation problem:

$$\min p_i - p_j \tag{7}$$

with $P \in \mathcal{P}$. Since any probability in $\mathcal{P}$ is such that $p_i > p_j$, the value of $p_i - p_j$ is guaranteed to be positive, hence x_j is preferred to x_i ($x_i \succ x_j$).

Note that Proposition 1 only holds if the associated constraint has not been discounted.

3 Discounting strategies for inconsistent outputs

As remarked in Section 2.2, multiple classifiers may provide inconsistent outputs, in which case the constraints induced define an empty credal set $\mathcal{P}$. In this section, we explore various discounting strategies to relax these constraints in order to make the set of probability distributions non-empty.

3.1 ϵ-discounting of binary classifiers

In this paper, we perform an ϵ-discounting for each classifier, in order to relax Constraints (4)–(5) as described by Equation (6). For the j^{th} classifier, we obtain from Constraints (4)–(5) that $\underline{E}_{\underline{f}_j} = 0$ and $\overline{E}_{\overline{f}_j} = 0$ with

$$\underline{f}_j(x) = \begin{cases} 1 - \alpha_j & \text{if } x \in A_j \\ -\alpha_j & \text{if } x \in B_j \\ 0 & \text{else} \end{cases} \quad \text{and} \quad \overline{f}_j(x) = \begin{cases} 1 - \beta_j & \text{if } x \in A_j \\ -\beta_j & \text{if } x \in B_j \\ 0 & \text{else} \end{cases}.$$

This gives the following discounted equations:

$$\epsilon_j(-\alpha_j) \leq (1 - \alpha_j)P(A_j) - \alpha_j P(B_j), \tag{8}$$
$$\epsilon_j(1 - \beta_j) \geq (1 - \beta_j)P(A_j) - \beta_j P(B_j). \tag{9}$$

Remark that the two discounted equations are here linear in variables p_i and ϵ_j. The constraints become empty when $\epsilon_j = 1$ and are then equivalent to state $P(A_j|A_j \cup B_j) \in [0, 1]$. This means that there always exists a set of coefficients $\{\epsilon_j\}_{j=1,\ldots,N}$ that makes the problem feasible.

The question is now how to compute the discounting rates ϵ_j, $j = 1, \ldots, N$ such that the constraints induce a non-empty credal set $\mathcal{P}$ while minimizing

the discounting in some sense. We propose the following approach to find the coefficients ϵ_j:

$$\min \sum_{j=1}^{N} \epsilon_j$$

under the constraints

$$\sum_{x_i \in \mathcal{X}} p_i = 1, \quad 0 \leq p_i \leq 1 \text{ for all } i = 1, \ldots, M, \quad 0 \leq \epsilon_j \leq 1 \text{ for all } j = 1, \ldots, N,$$

and Constraints (8)–(9). It is interesting to notice that this new approach is similar to strategies proposed to find minimal sets of infeasible constraints in linear programs [2].

3.2 Credal discounting vs ϵ-discounting

In a previous paper [4], we proposed a discounting strategy that was applied to directly to bounds α_j, β_j before transforming Equation (2). The obtained discounted equation for the jth classifier is

$$(1 - \epsilon_j)\alpha_j \leq P(A_j | A_j \cup B_j) \leq \epsilon_j + (1 - \epsilon_j)\beta_j, \quad j = 1, \ldots, N.$$

However, applying such a discounting (or other correction) operation results in quadratic constraints once Equation (2) is "deconditioned" and transformed in Constraints (8)–(9). Discounted constraints on $P(B_j | A_j \cup B_j)$ are obtained by complementation. Remark further that for each constraint, all the coefficients of the square terms p_i^2 and ϵ_j^2 are zero. This implies that the associated quadratic form is indefinite. Therefore, computing the minimum-norm vector of coefficients $\epsilon_1, \ldots, \epsilon_N$ by solving an optimisation problem is very difficult, and searching the space of all solutions is very greedy. To overcome this problem, all the discounting factors were assumed to be equal, and were computed empirically by searching the parameter space (if $\epsilon_1 = \cdots = \epsilon_N$, a dichotomic search can be performed).

In the present approach, we have N discounting rates to compute. However, they may be determined by solving a linear optimization problem under linear constraints, which may be addressed more efficiently than searching the space of discounting coefficients.

3.3 Global vs local discounting

In this work, we advocate a local discounting approach, where each classifier is associated with a specific rate ϵ_i. In order to illustrate why this approach seems preferable to a global strategy, where all discounting rates are assumed to be equal, we concentrate on the one-vs-one problem (i.e., each classifier was trained to separate a single class from another). Let us now consider the following simple example:

Example 2. Consider $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$ and the following results:

$$P(x_i|x_i, x_j) \in [0.6, 1]$$

for all pair $1 \leq i < j \leq 4$, except for $P(x_1|x_1, x_4) \in [0, 0.4]$. Thus, all classifier outputs are consistent with $p_1 > p_2 > p_3 > p_4$, except $P(x_1|x_1, x_4)$ from which one would conclude $p_4 > p_1$. Now, if we were to discount all of them in the same way, we would obtain as a minimal discounting $\epsilon_{ij} = 1/6$ (and $\sum \epsilon_{ij} = 1$, with ϵ_{ij} the discounting value of $P(x_i|x_i, x_j)$), with $p_1 = p_2 = p_3 = p_4 = 1/4$ being the only feasible solution. Thus, in this case, all the information provided by the classifiers is lost, and we are unable to choose between one of the four classes.

Now, assume that each classifier is discounted separately from the others; then, taking $\epsilon_{14} = 1/3$ restores consistency (e.g., $p_1 = 0.5$, $p_2 = 0.31$, $p_3 = 0.2$, $p_4 = 0.09$ is a solution) while still preserving the ordering $p_1 > p_2 > p_3 > p_4$.

4 Experiments

In this section, we present some experiments performed on classical and simulated data sets. We considered both decision rules presented in Section 2.4 to make decisions. Since the maximality rule provides a set of possible classes, we need to define a way to evaluate the accuracy of the decision system in this case. Section 4.1 addresses this topic.

4.1 Evaluating classifiers performances

Combined classifiers used with a maximin rule can be directly compared to classical classifiers or to more classical combinations, as both return a single class as output. In this case, accuracy is simply measured as a classical accuracy that will be referred to *acc* in the following. However, one of the main assets of imprecise probabilistic approaches is the (natural) ability to return sets of classes when information is ambiguous or not precise enough to return a single class. In this case, comparing the imprecise classification output with a classical unique decision is not straightforward.

A first (naive) solution consists in considering the classification as fully accurate whenever the actual class of an evaluated data point belongs to the predicted set of possible classes $\widehat{X}$. It amounts to consider that the final decision is left to the user, who always makes the good choice. The error rate thus computed is an optimistic estimate of the accuracy of the classifier. This estimate will thereafter be referred to as *set accuracy*, or $s - acc$.

Another solution is to use a *discounted* accuracy. Assume we have T observations for which the actual classes $x_i, i = 1, \ldots, T$ are known, and for which T sets of possible classes $\widehat{X}_1, \ldots, \widehat{X}_T$ have been predicted. The discounted accuracy $d - acc$ of the classifier is then

$$d - acc = \frac{1}{T} \sum_{i=1}^{T} \frac{\Delta_i}{g(|\widehat{X}_i|)},$$

with $\Delta_i = 1$ if $x_i \in \widehat{X}_i$, zero otherwise and g an increasing function such that $g(1) = 1$. Although $g(x) = x$ is a usual choice for the discounted accuracy, it has recently been shown [11] that this choice leads to consider imprecise classification as being equivalent to make a random choice inside the set of optimal classes. This comes down to consider that a Decision Maker is risk neutral, i.e., does not consider that having imprecise classification in case of ambiguity is an advantage. This also implies that the robustness of an imprecise classification is rewarded by concave (or risk-averse) functions g.

In our case, we used the function $g(\cdot) = \log_M(\cdot)$ that satisfies $g(1) = 1$ and takes account of the number of classes. Indeed, in the case of two classes, we should have $g(2) = 2$, because predicting two-classes out of two is not informative. However, as M increases, predicting a small number of classes becomes more and more interesting. This is why we pick $\log_M$.

4.2 Datasets and experimental setup

Table 1. UCI data sets used in experiments

Data set name	#classes M	#input features	#samples
glass	6	9	214
satimage	6	36	6435
segment	7	19	2310
vowel	11	10	990
waveform	3	8	5000
yeast	10	8	1484
zoo	7	18	101
primary tumor	21	17	339
anneal	5	38	898

We used various UCI data sets that are briefly presented in Table 1. For each of these datasets, we considered the classical one-vs-one decomposition scheme, in which each classifier is trained to separate one class from another. We used as base classifiers CART decision trees (so that comparisons between the precise and the imprecise approaches can be done) and the imprecise Dirichlet model [1] to derive lower and upper conditional probability bounds. This model depends on a hyper-parameter s that settles how quickly the probability converges to a precise value. More precisely, if a_j, b_j are the two classes for the jth binary classifier, and if n_{a_j}, n_{b_j} are the number of training data having respectively a_j and b_j for classes in the leaf of the decision tree reached by the instance, then the bounds are

$$\alpha_j = \frac{n_{a_j}}{n_j + s}, \quad \beta_j = \frac{n_{a_j} + s}{n_j + s},$$

where $n_j = n_{a_j} + n_{b_j}$. Then, s can be interpreted as the number of "unseen" observations, and $\alpha_j = \beta_j$ if $s = 0$.

Table 2 summarises the results obtained for $s = 4$. We compared our method to a single CART decision tree (DT) and to Naive Bayes classifiers (NB). We also displayed the accuracy obtained with maximin rule as well as the set accuracy and the discounted accuracy obtained with the maximality rule, both for the global and local correction methods. We used a 10-fold cross validation. The significance of the differences between the results was evaluated using a Wilcoxon-signed rank test at level 95%. The best results (outside set accuracy) are underlined and results that are not significantly different are printed in bold. Note that we excluded s-acc since it is strongly biased in favor of imprecise decisions.

Table 2. UCI data sets used in experiments

Data set	NB	DT	local			global		
			acc	s-acc	d-acc	acc	s-acc	d-acc
glass	70.55	71.00	**74.77**	79.59	**74.29**	**73.7**3	78.55	**73.79**
satimage	84.34	80.06	87.27	90.43	**88.16**	86.37	89.15	87.78
segment	**96.19**	85.88	**95.58**	97.36	**96.63**	**96.37**	96.88	**96.50**
vowel	**78.88**	72.10	**80.32**	82.84	**81.32**	77.58	77.88	77.27
waveform	71.10	**80.94**	73.24	81.70	75.11	73.44	81.86	75.28
yeast	48.84	45.33	57.13	68.32	**61.94**	55.11	62.39	59.19
zoo	95.17	90.17	**96.17**	96.17	94.59	**96.17**	96.17	94.59
p. tumor	38.05	**48.38**	45.75	45.75	45.75	42.22	42.22	42.22
anneal	**95.99**	93.10	81.74	81.74	81.74	81.86	81.86	81.86

Two main remarks can be made. First, the one-versus-one decomposition strategy provides good results for most data sets, as it gives better results on 6 data sets out of 9. Second, it is clear that the local discounting strategy gives significantly better results than the global discounting strategy. The local strategy dominates the global one on most data sets and gives results very close to the global one otherwise (here, for the "waveform" and "anneal" datasets).

Let us remark that the parameter s, which is directly proportional to the amount of imprecision, has remained the same for all data sets. However, the resulting imprecision also depends on the data. This partly explains the differences between the set accuracy and the discounted accuracy obtained on the datasets: for instance, the resulting imprecision is moderate for "glass" and "yeast", but zero for "anneal" and "primary tumor").

In order to provide an idea of the impact of increasing the overall degree of imprecision, Figure 1 shows the evolution of the discounted accuracy as a function of $\log_2(s)$ (let us remind that since $g(|\widehat{X}_i|) = \log_M(|\widehat{X}_i|)$, the classifier reaches a score of 0.5 when it retains all the classes for all the instances). It shows that moderately increasing the imprecision can give better results (the maximum

is reached for $s = 8$) and that the discounted accuracy starts to decrease once the degree of imprecision becomes too large. Similar behaviors could be observed for other data sets.

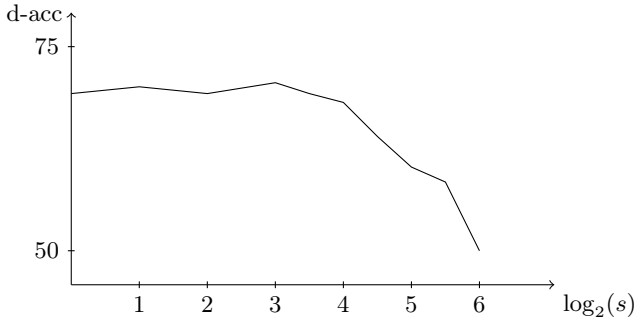

Fig. 1. Evolution of $d - acc$ for data set glass

5 Conclusions

We addressed the problem of pattern classification using binary classifier combination. We adopted imprecise probability theory as a framework for representing the imprecise outputs of the classifiers. More particularly, we consider classifiers that provide sets of conditional probability distributions. It encompasses both cases of precise and imprecise probabilistic outputs (including possibilistic, evidential [3] and credal classifiers [10]). The combination of such classifiers is done by considering classifier outputs as constraints. We presented a local discounting approach for relaxing some of these constraints when the classifiers provide inconsistent outputs. Our strategy computes the discounting rates by solving linear optimization problems, which can be efficiently solved by standard techniques.

Experiments demonstrate that our method give good results compared to the single classifier approach. Moreover, it performs almost always better than the global discounting approach that was presented in a former paper. In future works, we wish to extend our experimentation (by using precise classifiers and genuine imprecise classifiers) and to make a deeper analysis of their results (e.g., checking in which cases inconsistencies happen, verifying that imprecise classification correspond to instances that are hard to classify). We also wish to investigate on the properties of our approach from the point of view of decision making under uncertainty.

References

1. J.-M. Bernard. An introduction to the imprecise dirichlet model. *Int. J. of Approximate Reasoning*, 39:123–150, 2008.
2. J. W. Chinneck. Finding a useful subset of constraints for analysis in an infeasible linear program. *INFORMS Journal on Computing*, 9:164âĂŞ174, 1997.
3. T. Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Trans. Syst. Man. Cybern.*, 25:804–813, 1995.
4. S. Destercke and B. Quost. Combining binary classifiers with imprecise probabilities. In Y. Tang, V. Huynh, and J. Lawry, editors, *IUKM*, volume LNAI-7027 of *Lecture Notes in Artificial Intelligence*, pages 219–230. Springer-Verlag, 2011.
5. T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *The Annals of Statistics*, pages 507–513. MIT Press, 1996.
6. I. Levi. *The Enterprise of Knowledge*. MIT Press, London, 1980.
7. M. Troffaes. Decision making under uncertainty using imprecise probabilities. *Int. J. of Approximate Reasoning*, 45:17–29, 2007.
8. P. Walley. *Statistical reasoning with imprecise Probabilities*. Chapman and Hall, New York, 1991.
9. 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.
10. M. Zaffalon. The naive credal classifier. *J. Probabilistic Planning and Inference*, 105:105–122, 2002.
11. M. Zaffalon, G. Corani, and D. Maua. Utility-based accuracy measures to empirically evaluate credal classifiers. In *ISIPTA 11*, 2011.