Combining binary classifiers with imprecise probabilities

Sébastien Destercke, Benjamin Quost

Combining binary classifiers with imprecise probabilities.

Sébastien Destercke[1] and Benjamin Quost[2]

[1] INRA/CIRAD, UMR1208, 2 place P. Viala, F-34060 Montpellier cedex 1, France
sdestercke@gmail.com
[2] HEUDIASYC, 6599. UniversitÃĺ de Technologie de CompiÃĺgne. Centre de Recherches de Royallieu. 60205 COMPIEGNE, France quostben@hds.utc.fr

Abstract. This paper proposes a simple framework to combine binary classifiers whose outputs are imprecise probabilities (or are transformed into some imprecise probabilities, e.g., by using confidence intervals). This combination comes down to solve linear programs describing constraints over events (here, subsets of classes). The number of constraints grows linearly with the number of classifiers, making the proposed framework tractable even for problems involving a relatively large number of classes.

1 Introduction

In complex multi-class classification problems, it often makes sense to decompose the initial problem into several simpler ones, training simple classifiers on each of these problems and then combining their results. This is the central idea of technics such as Boosting or classifiers combinations [3, Ch.14].

In this paper, we deal with the case where each simple problem is a binary one, and where each classifier task is to tell two subsets of classes apart. When those classifiers return conditional probabilities estimating whether an instance belong to a given class subset, 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 optimisation problem whose solution is a consistent probability whose conditional probabilities are close to each of the estimated ones [7, 13]. This consistent probability is then considered as the final predictive model.

In this paper, we address the problem from a slightly different viewpoint, that is the one of imprecise probabilities. Imprecise probabilities [12] are concerned with the cases where the available information is not sufficient (or too conflicting) to identify a single probability distribution as our model of uncertainty. They are therefore fit to deal with the problem of combining inconsistent (precise or imprecise) conditional probabilities. Instead of searching for a close (w.r.t. some objective function) consistent solution, we suggest to weaken the given conditional assessments to make them consistent, and to consider the resulting set

of probabilities as our final predictive model. 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 this paper, after a brief reminder about imprecise probabilities (Section 2), we first describe (Section 3) how binary classifiers returning imprecise probabilities (precise probabilities then becoming a special case) as their conditional assessments can be combined. As this combination can still lead to inconsistent results (i.e., non-feasible linear programs), we then propose a discounting strategy ensuring that a consistent (but possibly non-informative) result will be reached (Section 4). Finally, we make some first experiments for the special case of one-vs-one classifiers on simulated and well-known data sets, considering both the case of precise and imprecise classifiers (Section 5).

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, $\mathcal{X}$ consists in the classes of an instance). In imprecise probability, the uncertainty about a variable X true value is described by a convex set of probablities $\mathcal{P}$, often called *credal set* [9]. A classical way to describe this set is by giving a set of linear constraints restricting the set of possible probabilities in $\mathcal{P}$ (Walley's lower previsions [12] correspond to bounds of such constraints).

Credal sets have been proposed as models of uncertainty when available information does not allow one to identify a unique probability of interest. In this paper, we propose to apply the imprecise probabilistic framework to the combination of binary classifiers.

From a credal set $\mathcal{P}$, one can compute lower and upper probabilities $\underline{P}, \overline{P}$ such that, for any event $A \subseteq \mathcal{X}$,

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

They are dual, in the sense that $\underline{P}(A) = 1 - \overline{P}(A^c)$, with A^c the complement of A. More generally, given a real-valued and bounded function f on $\mathcal{X}$, one can compute lower and upper expectation bounds $\underline{E}, \overline{E}$ such that

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

with E the expected value of f w.r.t. p. They are also dual, in the sense that $\overline{E}(f) = -\underline{E}(-f)$. Note that lower and upper probabilities of an event A equal the lower and upper expectations of its indicator function. Alternatively, one can start from constraints on expected or probability values and take the set of probabilities satisfying these constraints.

3 Combining binary classifiers with imprecise probabilities

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 assuming their value in some space $\mathcal{Y}$. Usually, it is assumed that to a given input y correspond a probability mass $p(x|y)$ modelling the class distribution under input y. Classification then amounts to estimate as accurately as possible $p(x|y)$ from a limited set of 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., $A \cap B = \emptyset$). Its prediction is then an estimation of the conditional probability $P(A|A \cup B, y)$ that the instance belongs to A ($P(B|A \cup B, y) = 1 - P(A|A \cup B, y)$ by duality).[3] Combining binary classifier then consists in finding p from a set of such conditional assessments.

To model a set of binary classifiers, we will use the language of code correction matrices. A code correction matrix is a matrix C with general element $c_{ij} \in \{+1, 0, -1\}$, $i \in 1, \ldots, M$ with M the number of classes, and $j \in 1, \ldots, N$ with N the number of binary classifiers to combine. For a given column j, the sets $A_j = \{x_i | c_{ij} = 1, i = 1, \ldots, M\}$ and $B_j = \{x_i | c_{ij} = -1, i = 1, \ldots, M\}$ are the positive and negative classes that classifier j separates. We now recall the combination problem in a precise setting and then extend it to an imprecise setting.

3.1 The precise case

In the precise case, classifier j returns a precise evaluation $P(A_j|A_j \cup B_j) = \alpha_j$. Using the fact that $P(A_j|A_j \cup B_j) = {}^{P(A_j)}/_{P(A_j \cup B_j)}$ and $P(B_j|A_j \cup B_j) = {}^{P(B_j)}/_{P(A_j \cup B_j)} = 1 - P(A_j|A_j \cup B_j)$, we obtain from these two equations the following equality[4]

$$P(A_j) = \frac{\alpha_j}{1 - \alpha_j} P(B_j).$$

This gives N equalities that describe partial knowledge about the true but unknown probabilities. As the number of equalities will be usually much higher than the number M of elements of $\mathcal{X}$, the problem will often be over-constrained and without solutions, as shows the next example.

Example 1. Consider a 3 classes problem $\mathcal{X} = \{x_1, x_2, x_3\}$. Assuming we are working with a one-against-one framework (each A_j, B_j is reduced to a singleton), consider the following output of classical probabilistic classifiers:

$$P(\{\{x_1\}|\{x_1, x_2\}\}) = 0.2, P(\{\{x_1\}|\{x_1, x_3\}\}) = {}^1/_3, P(\{\{x_2\}|\{x_2, x_3\}\}) = 0.8.$$

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

[4] We assume here that $p(\{x\})$ is strictly positive for any $x \in \mathcal{X}$. In a practical setting, this does not appear as a restrictive assumption, as $p(\{x\})$ can be as small as possible.

These statements, once transformed to express unconditional constraints, respectively give the equalities (using the notation $p_i = p(x_i) = P(\{x_i\})$)

$$p_1 = 1/4 p_2,\ p_1 = 1/2 p_3,\ p_2 = 4p_3,$$

which lead (together with the consistency constraints $\sum_{x_i \in \mathcal{X}} p_i = 1, p_i \geq 0$) to a system with no solutions.

3.2 The imprecise case

Let us now consider imprecise binary classifiers. The output of classifier j (or the transformation into imprecise probabilities of its precise output) will be a pair of values bounding the conditional probabilities on A_j, B_j. We will denote by α_j, β_j the bounds of $P(A_j | A_j \cup B_j)$, that is

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

and, by complementation, we have

$$1 - \beta_j \leq P(B_j | A_j \cup B_j) \leq 1 - \alpha_j. \tag{2}$$

To get a joint credal set from these constraints, we will turn them into linear constraints over unconditional probabilities. To get such constraints, we first transform equations (1) and (2) into (again assuming that $P(A_j \cup B_j) > 0$)

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

By dividing these two equations, we reach the following inequality

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

which can be transformed into two linear constraints

$$\frac{\alpha_j}{1 - \alpha_j} P(B_j) \leq P(A_j) \quad \textit{and} \quad P(A_i) \leq \frac{\beta_j}{1 - \beta_j} P(B_j).$$

These equations can be restated as

$$\frac{\alpha_j}{1 - \alpha_j} \sum_{x_i \in B_j} p_i \leq \sum_{x_i \in A_j} p_i \quad \textit{and} \quad \sum_{x_i \in A_j} p_i \leq \frac{\beta_j}{1 - \beta_j} \sum_{x_i \in B_j} p_i \tag{3}$$

with $p_i := p(x_i)$. Such constraints correspond to a linear program inducing a credal set $\mathcal{P}$ of possible probabilities. If N classifiers are trained, then there are $2N$ such equations. This means that the number of constraints grows linearly with the number of classifiers, while the number of variables remains constant ($= M$). As the quantity of classifiers will remain limited (usually between M and M^2), induced linear programs can be efficiently handled by modern optimisation techniques.

Example 2. Consider the same situation as in Example 1, but that classifiers provide the (slightly) relaxed system such that

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

corresponding to the classifier outputs

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

and

$$P(\{\{x_2\}|\{x_2, x_3\}\}) \in [2/3, 0.8].$$

Note that the constraints of Example 1 are included in these ones. The above system is no longer without solution, e.g., $p_1 = 0.1, p_2 = 0.6$ and $p_3 = 0.3$ is an admissible solution. Getting the minimal/maximal probabilities for each class then comes down to solve 6 optimization problems (i.e., minimising and maximising the probabilities p_i, that give

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

Hence, we can safely choose x_2 as the right class in this case.

Admitting imprecision in the classifier output will not always result in a feasible problem. Such a situation corresponds to the case $\mathcal{P} = \emptyset$. In the next section, we suggest a relaxation strategy to ensure that a given set of classifier outputs will end up in a feasible system (possibly providing a vacuous, i.e. non-informative, solution).

4 Handling inconsistent output: a discounting strategy

Here, we assume again that binary classifiers provide imprecise probabilistic outputs in the form of bounds $[\alpha_j, \beta_j], j = 1, \ldots, N$ (precise classifiers correspond to the case $\alpha_j = \beta_j$).

When the induced linear problem is not feasible, we propose to consider a discounting factor $\epsilon \in [0, 1]$ and to increase this factor up to the point where the linear problem becomes feasible (i.e. the associated credal set is no longer empty). For a given value ϵ, the ϵ-discounted problem corresponds, for $j = 1, \ldots, N$, to the constraints

$$(1 - \epsilon)\alpha_j \leq P(A_j|A_j \cup B_j) \leq \epsilon + (1 - \epsilon)\beta_j. \tag{4}$$

Discounted constraints on $P(B_j|A_j \cup B_j)$ are obtained by complementation. Note that this discounting strategy is common in robust Bayesian literature as well as in other imprecise probabilistic approaches, since it corresponds to the ϵ-contamination model [2] and to the basic discounting operation in evidence theory [10]. We denote by $\mathcal{P}^\epsilon$ the credal set obtained by discounting the initial problem with a value ϵ. Such a strategy ensures that there will be at least one value of ϵ for which the problem will be feasible, as $\epsilon = 1$ corresponds to trivial

constraints $P(A_j|A_j \cup B_j) \in [0,1]$, meaning that the set $\mathcal{P}^1$ corresponds to the set of all probability measures on $\mathcal{X}$. This alone is sufficient to ensure that the linear problem given by Eq. (4) will be feasible for some value ϵ.

For a given instance, let us denote ϵ_* the lowest value such that $\epsilon_* = \min_{\epsilon \in [0,1]} \mathcal{P}^\epsilon \neq \emptyset$. ϵ_* gives an indication of the *global level of conflict* of the various classifiers. Indeed, if $\epsilon_* = 0$ this means that all classifiers are consistent and no discounting is needed. On the contrary, if $\epsilon_* \simeq 1$ this means that *at least* one classifier gives a conditional information that is strongly conflicting with the others, and that a closer look should be taken to understand why such a conflict happens.

Also note that the obtained credal sets for different values of ϵ are nested in each others (i.e., $\mathcal{P}_\epsilon \subseteq \mathcal{P}_{\epsilon'}$ for any $\epsilon \leq \epsilon'$. This makes the current approach close to other similar models proposed in the imprecise probabilistic literature [4, 1, 5]. In the present work, the ϵ value should not be interpreted as having any statistical meaning in terms of confidence value. Indeed, linking ϵ to some statistical confidence value is the matter of further work.

5 Experiments

In this section, we perform some experiments on some classical and simulated data sets. In order to assess the results of Imprecise classifiers, we need at least two elements: how decision are taken, and how to evaluate performances when decisions are allowed to be imprecise (since imprecise probabilistic approaches allow for imprecise decisions. We first describe how this is done in the current study, before the detailing the experiment results.

5.1 Decision rules

Consider the set of classes $\mathcal{X}$, and the knowledge we have about the class of an object given by the Constraints (3) (or a discounted problem). Imprecise probability theory offers many ways to make a decision about the possible class of an object [11]. Roughly speaking, classical decision based on maximal expected value can be extended in two ways: by a decision rule whose result is a unique class, or by a decision rule whose result is a set of possible optimal classes. We will retain a rule of each type: the maximin and interval dominance rules.

For a class $x_i \in \mathcal{X}$, its lower and upper probabilities $\underline{P}(\{x_i\}), \overline{P}(\{x_i\})$ are given by the solutions of the constrained optimisation

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

under the Constraints (3) and the additional constraints $\sum_{x_i \in \mathcal{X}} p_i = 1$, $p_i > 0$.

The maximin decision rule amounts to choose as the object 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 interval dominance rule amounts to select as the possible optimal classes the set $\widehat{X}$ such that

$$\widehat{X} := \{x_i \in \mathcal{X} |\ \nexists x_j \text{ s.t. } \overline{P}(\{x_i\}) \leq \underline{P}(\{x_j\})\}.$$

Using this rule requires to solve $2M$ linear systems with $2N + M + 1$ constraints and to make $M(M-1)$ comparisons at most. Its complexity is thus only slightly higher than the maximin rule. Also note that we have $\widehat{x} \in \widehat{X}$.

5.2 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 (acc)

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, it can be hard to compare the IP classification output with a more classical one.

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

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

with $\Delta_i = 1$ if $x_i \in \widehat{X}_i$, zero otherwise and f an increasing function such that $f(1) = 1$. Although $f(x) = x$ is a usual choice for the discounted accuracy, it has recently been shown [15] that this choice leads to consider imprecise classification as being equivalent to make a random choice inside the set 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 means that to reward the robustness of imprecise classification, one should take a concave (i.e., risk-averse) function f. Since we think that returning an imprecise classification is an interesting feature and that classification robustness should be rewarded, we take here $f(x) = \sqrt(x)$.

In addition, when using the interval dominance rule, we have computed the set-accuracy $s - acc = {}^1/_T \sum_{i=1}^{T} \Delta_i$ as well as the average number of retained optimal classes on the instances that were well classed with the precise criterion and on those that were badly classed with the same criterion. This gives complementary information, as it allows to check whether imprecise classification mostly reject those cases that are difficult to classify.

5.3 Data-sets and set-up

In this case study, we performed the experiments on data sets issued from the UCI repository, whose details are recalled in Table 1, along with those of a synthetic data set **Synth**. **Synth** is a simulated data set consisting of four (partially overlapping) classes sampled according to 2-dimensional Gaussian mixtures.

Data set name	#classes M	#input features	#samples training set	test set
glass	6	9	139	75
pageblocks	5	10	3284	2189
satimage	6	36	2921	2573
segment	7	19	1400	910
Synth	4	2	2250	750
vowel	11	10	528	462
waveform	3	8	1491	3509
yeast	10	8	890	594

Table 1. UCI data sets used in experiments

Since the optimisation problem to find the discounting factor ϵ_* is not easy to solve, we have adopted the following strategy in our first experiments: for each test data, we start with a low $\epsilon = 0.001$ and increment it gradually and linearly by steps of 0.05 (note that we are certain to reach a feasible solution).

Precise case In the precise case, the classifiers used to perform the experiment were obtained by logistic regression [8]. Logistic regression being a linear classifier, it is well adapted to the combination of binary classifiers (it is easy to train but can only provide simple decision bounds).

For the Synthetic data set, Figures and respectively pictures the discounting level ϵ and the number of optimal decisions on the 2 dimensional input space. Roughly speaking, the behaviour of the combination methods corresponds to what can be expected: discounting (or, equivalently, conflict) increase on regions where multiple classes significantly overlap, and the number of optimal decisions increases along the decision boundaries of the different binary classifiers.

Table 2 summarises the results obtained by our method for the precise classifier. Among the results, we can distinguish two kinds of data sets: those for which interval dominance often gives precise results (waveform, pageblocks, synth), and those for which interval dominance often gives imprecise results. In the former sets, results are relatively stable in terms of all accuracy, meaning that binary classifiers gives relatively consistent results most of the time. In the latter sets, imprecision can be important, and s is the difference between $d-acc$ and $s-acc$. In such data sets, conflict among classifiers is likely to be important, and instances difficult to classify. In most of them, imprecision brings robustness (low

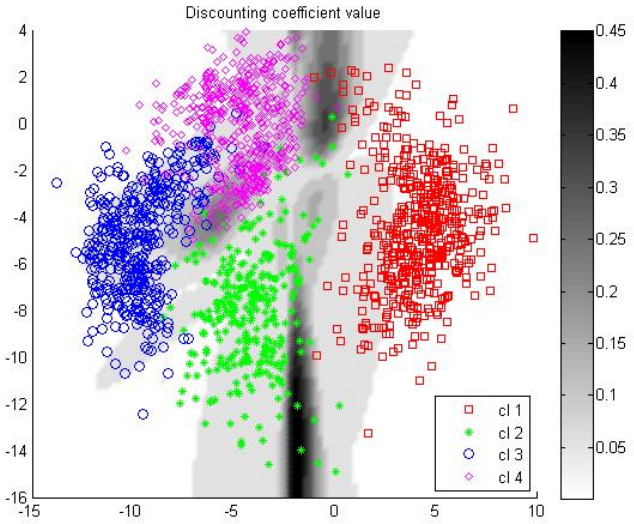

Fig. 1. Discounting levels for **Synth** test data set

$s - acc$), but at the price of important imprecision (high mean number of optimal classes). In all problems, imprecise classifications seem evenly distributed among well and wrongly classified items. That is not so surprising, since difficult to classify items will sometimes be well, sometimes be wrongly classified by a precise classifier.

It should also be noted that interval dominance is one of the most "cautious" imprecise probabilistic decision rule, in the sense that it results in the largest possible sets of optimal classes. Other more precise, i.e. resulting in narrower sets, decision rules could have been used. Devising a method to pick the "best" decision rule is the matter of future work.

data set	acc	$d - acc$	$s - acc$	mean # optimal classes		
				all data	well classed	wrongly classed
Synth	4.80	4.98	4.00	1.04	1.04	1.00
glass	41.3	44.69	28.00	2.16	2.50	3.05
pageblocks	3.79	5.80	2.10	1.23	1.21	1.20
satimage	14.54	24.66	8.05	2.05	2.12	2.08
segment	4.18	20.12	2.53	2.57	2.59	1.00
vowel	50.43	61.24	7.14	8.47	8.33	6.91
waveform	13.85	12.26	9.09	1.11	1.11	1.12
yeast	41.98	47.40	9.60	4.08	4.18	4.89

Table 2. Results of binary combination with logistic regression

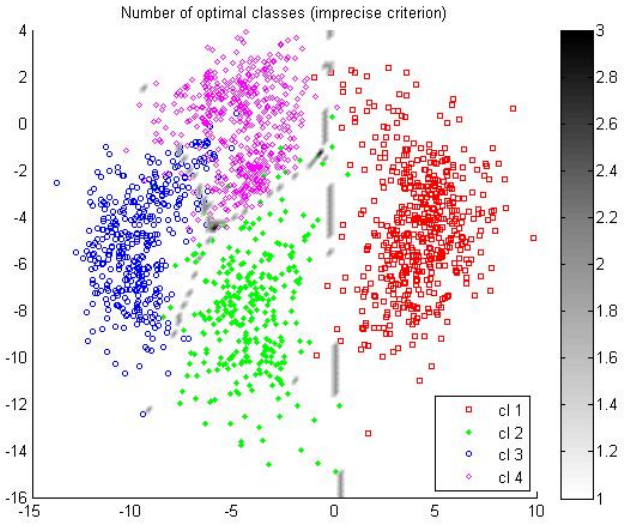

Fig. 2. Number of optimal classes for **Synth** test data set

Imprecise case In the imprecise case, we used the evidential k-nn algorithm [6] as a binary classifier. A first remark is that results are quite different from the ones obtained for the precise model. In particular, imprecision resulting from the use of interval dominance is here quite limited, as the mean value of optimal classes is always below two (for well and wrongly classed items).

This means that in many cases, results are both precise and fairly consistent. We see two main reasons for this: k-nn classification method has more complex and non-linear decision boundaries than logistic regression, and return imprecise probabilistic estimations that are usually not as close as 1 or 0 than the precise estimations returned by logistic regression models. We can therefore expect less conflict in the final assessment, whence more precise decisions. These results clearly show that the choice of the algorithm (and of its parameters) is important, and that this matter should be treated in further works.

Finally, Table compares the accuracy obtained with the multi-class equivalent of the binary classifiers with the accuracy obtained with our binary approach. For the logistic approach, results are either comparable or significantly better for the binary case (this is especially true for the segment data sets). For the k-nn approach, results are roughly comparable. In summary, the proposed binary decomposition either give better or comparable results. It is also worthwhile to note that, in the case of the k-nn approach, allowing imprecision systematically gives a better accuracy (in terms of $s-acc$) while not giving too imprecise results (mean number of optimal classes below 2).

data set	acc	$d-acc$	$s-acc$	mean # optimal classes		
				all data	well classed	wrongly classed
Synth	4.80	4.76	4.67	1.00	1.00	1.00
glass	44.0	45.42	42.67	1.19	1.07	1.06
pageblocks	5.39	9.87	4.75	1.36	1.38	1.31
satimage	10.57	10.20	9.72	1.02	1.02	1.00
segment	9.12	14.96	8.24	1.55	1.59	1.03
vowel	39.18	36.74	33.77	1.14	1.15	1.06
waveform	16.47	18.85	8.44	1.36	1.38	1.49
yeast	37.88	38.30	35.52	1.23	1.23	1.38

Table 3. Results of binary combination with evidential k-nn

Data set name	#multiclass		#binary (acc)	
	logistic	k-nn	logistic	k-nn
Synth	4.80	4.67	4.80	4.80
glass	44.0	48.0	41.30	44.0
pageblocks	4.02	5.16	3.79	5.39
satimage	14.26	10.53	14.54	10.57
segment	18.02	8.57	4.18	9.12
vowel	51.30	39.39	50.43	39.18
waveform	13.94	16.44	13.85	16.47
yeast	46.30	37.37	41.98	37.88

Table 4. Accuracy comparison of multiclass and binary methods

6 Conclusions

In this paper, we have introduced a method to combine binary classifiers based on an imprecise probabilistic approach. It handles classifiers with both precise and imprecise probabilistic outputs (including possibilistic, evidential [6] and credal classifiers [14]). We have also proposed a method that allows to always reach a solution, possibly leading to non-informative predictive model if classifiers outputs are too conflicting.

We also use imprecise decision rule, so that conflicting classifiers output points out to sets of possible optimal classes, and no longer to single classes, therefore producing more robust and trustable results.

First results on precise classifier (i.e. logistic regression) look very promising, while conclusions on the imprecise classifier are more mitigated, as they give good results but make poor use of imprecision. However, we think that this may be due both to the choice of the classification algorithm (as there are better choice for binary decomposition) and to the chosen parameters.

Future works therefore include the design of optimisation methods fitted to the imprecise framework, the use of the method on other well-known imprecise classification algorithms (such as credal networks or classification trees), and the design of optimisation mehods for the discounting value ϵ

Bibliography

[1] C. Baudrit, I. Couso, and D. Dubois. Joint propagation of probability and possibility in risk analysis: towards a formal framework. *Int. J. of Approximate Reasoning*, 45:82–105, 2007.

[2] J. O. Berger. An overview of robust Bayesian analysis. *Test*, 3:5–124, 1994. With discussion.

[3] C. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.

[4] M. Cattaneo. Likelihood-based statistical decisions. In *Proc. 4th International Symposium on Imprecise Probabilities and Their Applications*, pages 107–116, 2005.

[5] G. de Cooman. A behavioural model for vague probability assessments. *Fuzzy Sets and Systems*, 154:305–358, 2005.

[6] T. Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Trans. Syst. Man. Cybern.*, 25:804–813, 1995.

[7] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *The Annals of Statistics*, pages 507–513. MIT Press, 1996.

[8] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer-Verlag, 2001.

[9] I. Levi. *The Enterprise of Knowledge*. MIT Press, London, 1980.

[10] D. Mercier, B. Quost, and T. Denoeux. Refined modeling of sensor reliability in the belief function framework using contextual discounting. *Information Fusion*, 9(2):246–258, 2008.

[11] M. Troffaes. Decision making under uncertainty using imprecise probabilities. *Int. J. of Approximate Reasoning*, 45:17–29, 2007.

[12] P. Walley. *Statistical reasoning with imprecise Probabilities*. Chapman and Hall, New York, 1991.

[13] 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.

[14] M. Zaffalon. The naive credal classifier. *J. Probabilistic Planning and Inference*, 105:105–122, 2002.

[15] M. Zaffalon, G. Corani, and D. Maua. Utility-based accuracy measures to empirically evaluate credal classifiers. In *ISIPTA 11*, 2011.