Learning from data with uncertain labels by boosting credal classifiers

Benjamin Quost
HeuDiaSyC laboratory
deptartment of Computer Science
Compiègne University of Technology
Compiègne, France
quostben@hds.utc.fr

Thierry Denœux
HeuDiaSyC laboratory
deptartment of Computer Science
Compiègne University of Technology
Compiègne, France
tdenoeux@hds.utc.fr

ABSTRACT

In this article, we investigate supervised learning when training data are associated with uncertain labels. We tackle this problem within the theory of belief functions. Each training pattern $\mathbf{x}_i$ is thus associated with a basic belief assignment, representing partial knowledge of its actual class. Here, we propose to use the approach known as boosting to solve the classification problem. We propose a variant of the AdaBoost algorithm where the outputs of the classifiers are interpreted as belief functions. During training, our algorithm estimates the reliability of each classifier to identify patterns from the various classes. During test phase, the outputs of the classifiers are first discounted according to these reliabilities, and then combined using a suitable rule. Experiments conducted on classical datasets show that our algorithm is comparable to AdaBoost in accuracy. Processing EEG data with imperfect labels clearly demonstrates the interest of taking into account the reliability of the labelling, and thus the relevance of our approach.

Keywords

Dempster-shafer theory, theory of belief functions, theory of evidence; classification, uncertainty, classifier combination, AdaBoost.

1. INTRODUCTION

In a typical learning problem, we have a training set composed of p-dimensional input vectors $\mathbf{x}_i \in \mathbb{R}^p$, associated with labels taking values in a set of classes $\Omega = \{\omega_1, \dots, \omega_K\}$. A classifier is trained, on the basis of these labeled examples, to identify the class of any unknown test pattern $\mathbf{x}$. In some cases, however, collecting training patterns whose actual class is known with certainty may be expensive, difficult, or even impossible. For example, this is the case in medical applications, when the phenomena observed may be interpreted differently when analyzed by various physicians. Therefore, considerable interest has grown for representing and exploiting such imperfectly labeled data within the machine learning community.

The theory of belief functions [23, 28], also known as the theory of evidence, is a powerful tool for dealing with imperfect information. In this framework, belief functions express partial knowledge of the value of an unknown variable. Various kinds of belief about the class of a pattern may thus be represented, from full certainty to complete ignorance. Hence, the framework is particularly well suited to learning from data associated with uncertain labels [4, 6, 7, 29]. Another advantage of the belief functions theory lies in the existence of various mathematical tools for combining items of evidence. Thanks to this wide variety of rules, the theory of belief functions has been successfully applied to a broad range of classifier combination problems [17, 1, 20, 18, 19], an alternative approach that consists in training several classifiers for solving a learning problem.

Constructing accurate classification systems by combining simple algorithms has raised considerable interest in the field of machine learning. The approach known as boosting [12, 13] consists in training a *strong learner*, that is, a classifier with good classification accuracy, by combining *weak learners*. A set of weak learners is built as a sequence of classifiers $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_T$. A distribution of weights is maintained on the training examples, so that a classifier $\mathcal{C}_t$ concentrates harder on the training examples that were misclassified in the previous iterations.

In this article, we propose a variant of the widely known AdaBoost algorithm within the evidential framework. A weak learner produces, by definition, imperfect information about the class of a pattern. As a consequence, it may be best suited to identify some particular classes. We propose to take this knowledge into account by weakening the outputs of the weak learner proportionally to its unreliability in predicting the various classes. As we represent these outputs using belief functions, the contextual discounting operation [15] is well suited to this purpose. Finally, these discounted belief functions may be combined using a suitable combination rule, after what a decision may be taken. Our approach allows for learning from imperfectly labeled data. Indeed, partial knowledge on the actual class of a training pattern may be represented using a belief function. Hence, the outputs of a weak learner may be computed from data associated with such credal labels, rather than with crisp labels. Thus, training patterns with uncertain labels may still have an influence on the final decision, without being

U'09, June 28, 2009, Paris, France

given as much consideration as those whose actual class is known with certainty.

The article is organized as follows. Section 2 summarizes basic knowledge on belief functions, with emphasis on the discounting operation as well as on combination rules. Section 3 is an introduction to boosting, and more precisely the AdaBoost algorithm. In Section 4, we present an algorithm derived from AdaBoost for performing boosting in the framework of belief functions. We report experiments on synthetic and real life datasets in Section 5, and in particular on a classification problem where labels are uncertain. Section 6 concludes the paper.

2. BELIEF FUNCTIONS

In this article, we adopt the Transferable Belief Model (TBM) [28, 26] as an interpretation of the theory of belief functions. The main notions are recalled in this section.

2.1 Representing Bodies of Evidence

2.1.1 Basic Definitions

Let $\mathcal{C}$ be a classifier giving information on the actual class of a test pattern $\mathbf{x}$. This information may be represented by a basic belief assignment (bba) m, defined as a mapping from 2^{Ω} to $[0;1]$ satisfying $\sum_{A \subseteq \Omega} m(A) = 1$ (here, 2^{Ω} denotes the powerset of Ω). Let $m(A) > 0$: then, $A \subseteq \Omega$ is called a focal set of m, and the basic belief mass (bbm) $m(A)$ quantifies the belief that the actual class of $\mathbf{x}$ is in A. This belief could be transferred to more precise hypotheses $B \subseteq A$, if additional knowledge became available. The bbm $m(\emptyset)$ may be strictly positive: it represents the total amount of conflict between all the pieces of information available.

A bba is said to be:

- dogmatic, if Ω is not a focal set;
- simple, if it has at most two focal sets, including Ω; such a bba m, such that $m(A) = 1 - w$ (with $A \subsetneq \Omega$) and $m(\Omega) = w$, may be written A^{w};
- vacuous, if Ω is the unique focal set;
- contradictory, if $\emptyset$ is the unique focal set;
- categorical, if it has only one focal element that is not Ω;
- normal, if $m(\emptyset) = 0$ (otherwise, it is a subnormal bba);
- Bayesian, if its focal sets are singletons only;
- consonant, if all its focal sets $A_1, \ldots, A_N$ are nested: $\emptyset \subseteq A_1 \subseteq \cdots \subseteq A_N \subseteq \Omega$.

Any subnormal bba m can be normalized; the normalization operation is defined by:

$$m^{*}(A) = \frac{m(A)}{1 - m(\emptyset)}, \quad \forall A \subseteq \Omega. \tag{1}$$

Probability and possibility distributions may be easily represented using Bayesian and consonant bbas, respectively. The theory of belief functions is thus a very general framework that allows for exploiting tools (and in particular classification algorithms) developed within each of these theories.

Note that a body of evidence may also be represented by other functions, which are in one-to-one correspondence with m. For example, the commonality function is defined by:

$$q(A) = \sum_{A \subseteq B} m(B); \tag{2}$$

Conversely, one may compute m from q by:

$$m(A) = \sum_{A \subseteq B} (-1)^{|B|-|A|} q(B). \tag{3}$$

Example 1 Table 1 shows six basic belief assignments defined on a frame $\Omega = \{\omega_1, \omega_2, \omega_3\}$: each one encodes a belief on the actual class of a pattern $\mathbf{x}$. It can be seen that m_2 is Bayesian: it may be seen as a credal representation of the probability distribution $(p_1 = 0.2, p_2 = 0.3, p_3 = 0.5)$. Also, m_4 is consonant, and corresponds to the possibility distribution $(\pi_1 = 0.6, \pi_2 = 0.8, \pi_3 = 1)$. The categorical bba m_5 expresses the total certainty that the actual class of $\mathbf{x}$ is ω_3. The vacuous bba m_6 represents total ignorance of the actual class of $\mathbf{x}$. Let us further remark that m_2, m_3 and m_5 are dogmatic, and that only m_1 is subnormal.

Table 1: Six examples of bbas.

focal element	m_1	m_2	m_3	m_4	m_5	m_6
$\emptyset$	0.5	0	0	0	0	0
$\{\omega_1\}$	0.05	0.2	0	0	0	0
$\{\omega_2\}$	0.05	0.3	0.1	0	0	0
$\{\omega_1, \omega_2\}$	0	0	0.2	0	0	0
$\{\omega_3\}$	0.15	0.5	0.3	0.2	1	0
$\{\omega_1, \omega_3\}$	0	0	0.2	0	0	0
$\{\omega_2, \omega_3\}$	0.1	0	0.2	0.2	0	0
Ω	0.15	0	0	0.6	0	1

2.1.2 The Conjunctive Rule of Combination

Two bbas m_1 and m_2, provided by distinct sources of information $\mathcal{C}_1$ and $\mathcal{C}_2$, may be combined using the unnormalized Dempster's rule of combination, also known as the conjunctive rule of combination $\textcircled{\cap}$ [25]. The resulting bba, written $m_{1 \textcircled{\cap} 2}$, summarizes all the information provided by $\mathcal{C}_1$ and $\mathcal{C}_2$:

$$m_{1 \textcircled{\cap} 2}(A) = \sum_{X \cap Y = A} m_1(X) m_2(Y), \ \forall A \subseteq \Omega. \tag{4}$$

The normalized Dempster's rule of combination $\oplus$ [23] consists in first applying the unnormalized conjunctive rule $\textcircled{\cap}$, and then normalizing the result using Equation (1).

2.1.3 Partial Ordering on Bbas

The informational content of two bodies of evidence may be partially ordered in different ways. For example, the q-ordering [9] is defined as follows. Let m_1 and m_2 be two bbas, and q_1 and q_2 their associated commonality functions. Then m_1 is q-more committed than m_2, which we write $m_1 \sqsubseteq_q m_2$, iff:

$$q_1(A) \leq q_2(A), \text{for all } A \subseteq \Omega.$$

Example 2 Table 2 presents the commonality functions associated with bbas m_1, m_4 and m_6. It can be checked that $m_1 \sqsubseteq_q m_4$, $m_4 \sqsubseteq_q m_6$, and hence $m_1 \sqsubseteq_q m_6$. The second and third comparative statements are intuitive: since m_6 is the vacuous bba, it has thus by definition a very low informational content.

Table 2: Commonality functions associated with bbas m_1, m_4 and m_6 of Example 1.

focal element	q_1	q_4	q_6
$\emptyset$	1	1	1
$\{\omega_1\}$	0.2	0.6	1
$\{\omega_2\}$	0.3	0.8	1
$\{\omega_1, \omega_2\}$	0.15	0.6	1
$\{\omega_3\}$	0.4	1	1
$\{\omega_1, \omega_3\}$	0.15	0.6	1
$\{\omega_2, \omega_3\}$	0.25	0.8	1
Ω	0.15	0.6	1

2.1.4 Decision Making

The TBM distinguishes between the credal level, where the beliefs are entertained, and the pignistic level, where they are used to make decisions. In [28], it was shown that decisions should be based on probabilities for Dutch Books to be avoided. The *pignistic transformation* was then defined on the basis of elementary rationality requirements. The pignistic probability distribution associated with a bba m may be computed by first normalizing m, and then dividing each bbm $m^*(A)$ equally between the $\omega_k \in A$:

$$BetP(\omega_k) = \sum_{\omega_k \in A} \frac{m^*(A)}{|A|}, \ \forall \omega_k \in \Omega. \tag{5}$$

The operator Bet defined by $BetP = \text{Bet}(m)$ is clearly nonlinear. It should also be remarked that a same $BetP$ generally corresponds to various bbas; we may then define:

$$\text{Bet}^{-1}(BetP) = \{m : \text{Bet}(m) = BetP\}.$$

The q-ordering may be used to reverse the pignistic transform. To avoid giving unjustified support to any $A \subseteq \Omega$, we may select $\widehat{m} = \text{Bet}^{-1}_{qlc}(BetP)$, the least q-informative bba $\widehat{m}$ in $\text{Bet}^{-1}(BetP)$:

$$\begin{cases} \widehat{m} \in \text{Bet}^{-1}(BetP), \\ m \sqsubseteq_q \widehat{m}, \ \text{for all } m \in \text{Bet}^{-1}(BetP). \end{cases}$$

In [10], it was shown that $\widehat{m}$ is unique and that it is a consonant bba. It may be obtained by first computing $pl(\{\omega_i\})$ for all $1 \leq i \leq K$ and then deducing $pl(A)$, for all $A \subseteq \Omega$ with $|A| > 1$:

$$\begin{aligned} pl(\{\omega_i\}) &= \sum_{j=1}^{K} \min(p_i, p_j), & (6) \\ pl(A) &= \max_{\omega_k \in A} pl(\{\omega_k\}). & (7) \end{aligned}$$

Example 3 Bbas m_1, m_2, m_3 and m_4 of Example 1 have all the same pignistic probability distribution defined by:

$$\begin{aligned} BetP(\omega_1) &= 0.2 \\ BetP(\omega_2) &= 0.3 \\ BetP(\omega_3) &= 0.5. \end{aligned}$$

The bba m_4 was computed as the least q-committed bba leading to this pignistic probability distribution. Hence, $m_1 \sqsubseteq_q m_4$, $m_2 \sqsubseteq_q m_4$, and $m_3 \sqsubseteq_q m_4$; remark that the former relationship was checked in Example 2.

2.2 Canonical Conjunctive Decomposition of a Bba

2.2.1 The Notion of Separability

According to Shafer [23, Chapter 4], a normal bba is separable if it may be written as the normalized conjunctive combination of simple bbas:

$$m = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w(A)}, \tag{8}$$

with $w(A) \in [0; 1]$ for all $A \neq \emptyset$, $A \subset \Omega$. This *canonical representation* of m is unique for non dogmatic bbas.

Denœux [5] extended this representation to subnormal bbas. A bba m is separable if it may be written as the (unnormalized) conjunctive sum of simple bbas:

$$m = \bigcirc\!\!\!\!\cap_{A \subset \Omega} A^{w(A)}, \tag{9}$$

with $w(A) \in [0; 1]$, for all $A \subset \Omega$. Again, this representation (9) is unique for non-dogmatic bbas.

2.2.2 Practical Computation

Given a body of evidence, one may compute the corresponding weight function w from any other associated distribution. For example, one may compute w from q using:

$$w(A) = \prod_{A \subseteq B} q(B)^{(-1)^{|B|-|A|+1}}, \text{ for all } A \subseteq \Omega. \tag{10}$$

Remark that Equation (10) is equivalent to:

$$\ln w(A) = -\sum_{A \subseteq B} (-1)^{|B|-|A|} \ln q(B), \text{ for all } A \subseteq \Omega. \tag{11}$$

One notices the similarity with Equation (3). Hence, as pointed out in [5], any procedure suitable for transforming q to m can be used to compute $\ln w$ from $-\ln q$. Remark also that the transform into w from q is nonlinear. More generally, so is the transform into w of any distribution obtained linearly from m, as well as its converse.

Example 4 Table 3 shows the conjunctive weight functions associated with bbas m_1, m_4 and m_6 of Example 1.

Table 3: Conjunctive weight functions associated with bbas m_1, m_4 and m_6 of Example 1.

focal element	w_1	w_4	w_6
$\emptyset$	0.64	1	1
$\{\omega_1\}$	0.75	1	1
$\{\omega_2\}$	0.833	1	1
$\{\omega_1, \omega_2\}$	1	1	1
$\{\omega_3\}$	0.625	0.8	1
$\{\omega_1, \omega_3\}$	1	1	1
$\{\omega_2, \omega_3\}$	0.6	0.75	1

2.2.3 The Cautious Rule of Combination

The conjunctive combination of two bbas m_1 and m_2 may be easily processed using their associated weight functions w_1 and w_2:

$$w_{1 \bigcirc\!\!\!\!\cap 2}(A) = w_1(A) w_2(A), \quad \forall A \subset \Omega. \tag{12}$$

It is easy to see here that the conjunctive rule is commutative and associative. However, it is not idempotent: in particular, combining a separable cwf with itself results in decreasing all the weights $w(A) \neq 1$. More generally, $\textcircled{\cap}$-combining the outputs of two non-independent classifiers generally results in counting several times the identical items of evidence.

A cautious approach in combining two bodies of evidence would then consist in counting each item only once [24, 27, 5], considering that they may have been built on common information. In the most extreme case where two identical bodies of evidence are combined, the result should be the body itself — hence, the operator should be idempotent. The cautious rule of combination, written $\owedge$, consists in taking the minimum of the cwfs, instead of the product [5]. Let $w_{1\owedge 2}$ denote the resulting cwf; then:

$$w_{1\owedge 2}(A) = w_1(A) \wedge w_2(A), \quad \forall A \subset \Omega, \tag{13}$$

where $\wedge$ stands for the minimum operation.

As pointed out in [5], the product and minimum operators are both triangular norms (or t-norms for short) [14]. Hence, parameterized families of t-norms, counting both operators as particular cases, may be used to define families of combination rules intermediate between the conjunctive rule and the cautious rule for combining separable bbas. For example, Frank's family of t-norms is defined as

$$x \top_s y = \log_s \left(1 + \frac{(s^x - 1)(s^y - 1)}{s - 1} \right), \tag{14}$$

where $\log_s$ defines the logarithm function with base s. Any value $s \in]0; +\infty[$ defines a t-norm, and thus a combination rule $\textcircled{\top}_s$. The min operator is retrieved as $s \to 0$, and the product as $s = 1$. In [18, 19], we investigated the use of such rules for solving classification problems. It was shown that using an intermediate rule to combine a set of non-distinct classifiers may improve the classification accuracy of the ensemble.

Finally, we may remark that other rules of combination were defined within the theory of belief functions. For example, the mean rule consists in computing the element-wise average of two bbas. This rule has been applied, in particular, in [29, 11].

Example 5 Table 4 shows the bbas obtained by combining m_1 and m_4 using Dempster's rule, the cautious rule, an intermediate rule with $s = 0.5$, and the mean rule; the results are denoted by $m_{1\textcircled{\cap}4}$, $m_{1\owedge 4}$, $m_{1\textcircled{\top}_s 4}$, and mean, respectively.

Table 4: Combination of m_1 and m_4 of Example 1 using various combination rules.

focal element	$m_{1\textcircled{\cap}4}$	$m_{1\owedge 4}$	$m_{1\textcircled{\top}_s 4}$	mean
$\emptyset$	0.53	0.5	0.527	0.25
$\{\omega_1\}$	0.03	0.05	0.032	0.025
$\{\omega_2\}$	0.04	0.05	0.041	0.025
$\{\omega_1, \omega_2\}$	0	0	0	0
$\{\omega_3\}$	0.2	0.15	0.195	0.175
$\{\omega_1, \omega_3\}$	0	0	0	0
$\{\omega_2, \omega_3\}$	0.11	0.1	0.11	0.15
Ω	0.09	0.05	0.095	0.375

2.3 Discounting bodies of evidence

Suppose that several classifiers $\mathcal{C}_1, \ldots, \mathcal{C}_T$ provide information about the actual class of a test pattern $\mathbf{x}$; the output of each classifier $\mathcal{C}_t$ is represented with a bba m. Let us suppose further that classifier $\mathcal{C}_u$ has been wrongly trained. Thus, it provides erroneous information, and the bba m_u may be totally conflictuous with the others. Combining all the bbas together using a conjunctive rule of combination may then yield the bba such that $m(\emptyset) = 1$.

2.3.1 *Discounting*

If knowledge about the reliability classifier $\mathcal{C}_u$ is available, it should be used in order to avoid this uninformative result. The (classical) discounting operation consists in weakening the bba m_u, using a coefficient $\alpha \in [0; 1]$ that reflects our knowledge of the reliability of $\mathcal{C}_u$:

$$\begin{aligned} {}^{\alpha}m_u(A) &= (1-\alpha)m_u(A), \text{ for all } A \subset \Omega; \quad (15) \\ {}^{\alpha}m_u(\Omega) &= (1-\alpha)m_u(\Omega) + \alpha. \quad (16) \end{aligned}$$

Hence, discounting the bba m_u leads to transfering any bbm $m_u(A)$ (with $A \subset \Omega$) to Ω proportionnally to $\alpha \in [0; 1]$. The parameter α may be interpreted as the plausibility that the source of information is not reliable [15]. If the classifier $\mathcal{C}_u$ is fully reliable, then $\alpha = 0$ and ${}^{\alpha}m_u = m_u$; when it is totally unreliable, we have $\alpha = 1$ and ${}^{\alpha}m_u$ is the vacuous belief function.

2.3.2 *Contextual discounting*

Consider a classification problem with $K = 2$ classes: $\Omega = \{\omega_1, \omega_2\}$. Let us assume that a classifier $\mathcal{C}_u$ is perfectly able to identify patterns of class ω_1, but totally unreliable when processing data from ω_2. Then, by discounting m_u using $\alpha = 1$, crucial information $m(\{\omega_1\})$ may be ignored. In contrast, using $\alpha = 0$ would lead to taking totally unreliable information $m(\{\omega_2\})$ into account. Choosing $\alpha = .5$ as a compromise would result in weakening partially both bbms, hence giving less credit to ω_1, and more credit to ω_2, than should be.

The contextual discounting [15] makes it possible to cope with refined information about the reliability of the source $\mathcal{C}_u$. Instead of using a single coefficient α to model this reliability, we consider then one coefficient α_k for each atom $\omega_k \in \Omega$. The parameter α_k quantifies the plausibility that $\mathcal{C}_u$ is not reliable when the actual class of the pattern $\mathbf{x}$ observed is ω_k. Equivalently, it represents the plausibility that $\mathcal{C}_u$ is unable to recognize ω_k. The expression of the contextual discounting becomes much simpler in the case of two classes. Furthermore, if we consider normal bbas, we have then:

$$\begin{aligned} {}^{(\alpha)}m(\emptyset) &= 0; \quad (17) \\ {}^{(\alpha)}m(\{\omega_1\}) &= (1-\alpha_2)m(\{\omega_1\}), \quad (18) \\ {}^{(\alpha)}m(\{\omega_2\}) &= (1-\alpha_1)m(\{\omega_2\}), \quad (19) \\ {}^{(\alpha)}m(\Omega) &= m(\Omega) + \alpha_2 m(\{\omega_1\}) + \alpha_1 m(\{\omega_2\}). \quad (20) \end{aligned}$$

In other terms, $m(\{\omega_1\})$ (respectively, $m(\{\omega_2\})$) is transferred to Ω proportionally to the unability of the classifier to recognize class ω_2 (respectively, class ω_1).

Example 6 Let $\Omega = \{\omega_1, \omega_2\}$ be the set of classes in a two-class problem. Suppose that a classifier was trained using 100 patterns from each class; it misclassifies 20 % of the former, and 60 % of the latter. Table 5 shows two bbas m_7 and m_8 output by the classifier, their classical discountings ${}^{\alpha}m_7$ and ${}^{\alpha}m_8$ using a rate $\alpha = .4$, and their contextual discountings ${}^{(\boldsymbol{\alpha})}m_7$ and ${}^{(\boldsymbol{\alpha})}m_8$ using rates $\alpha_1 = 0.2$ and $\alpha_2 = 0.6$. It may be noticed that $m(\{\omega_1\})$ is more discounted, and $m(\{\omega_2\})$ less discounted, when using the contextual discounting instead of the classical discounting: indeed, $\alpha_1 < \alpha < \alpha_2$.

Table 5: Bbas m_7 and m_8; discountings ($\alpha = 0.4$), contextual discountings ($\alpha_1 = 0.2$ and $\alpha_2 = 0.6$).

focal element	m_7	${}^{\alpha}m_7$	${}^{(\boldsymbol{\alpha})}m_7$	m_8	${}^{\alpha}m_8$	${}^{(\boldsymbol{\alpha})}m_8$
$\emptyset$	0	0	0	0	0	0
$\{\omega_1\}$	0.7	0.42	0.28	0.3	0.18	0.12
$\{\omega_2\}$	0.2	0.12	0.16	0.6	0.36	0.48
Ω	0.1	0.46	0.56	0.1	0.46	0.4

3. BOOSTING CLASSIFIERS

Boosting [12, 13] aims at training an accurate classifier $\mathcal{C}$ by combining *weak learners*, i.e., classifiers that perform reasonably well on a given problem. We focus here on the AdaBoost algorithm [13], designed for two-class problems, and for which the pseudo-code is given in Figure 1. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be the training patterns, and $y_1, \ldots, y_n$ their associated labels. AdaBoost maintains a distribution of weights $w_1, \ldots, w_n$ on these examples. At each iteration, a weak learner is trained, and its performances are evaluated, on the weighted training examples. Weights associated with misclassified patterns are then increased, and the others are decreased: the next weak learner will attempt to correctly classify previous errors.

AdaBoost is known to overfit slowly (that is, for a very high number of weak learners). This obviously depends on the base classification algorithm employed. The weak learners are meant to be simple classifiers, that perform reasonably well on the training data. Earlier versions of boosting required that weak learners make less than 50 % errors. However, AdaBoost allows for this error ϵ_t to be arbitrarily high: weak learner $\mathcal{C}_t$ will have a negative weight in the final decision (as can be seen in Equation (21)).

Although many classification algorithms are eligible for boosting, classifiers with high sensibility to data are generally prefered. Indeed, changing the weights associated with training patterns is more likely to give a significantly different decision rule. Then, for a given number of weak learners, the diversity of the ensemble may be higher than with more stable classifiers. Note also that when the base algorithm does not permit to use weights on the training samples, the training set may be resampled according to the weight distribution $\mathbf{w}_t$ at each iteration. However, the distributions thus obtained are poorer, and the resulting classifier is more likely to overfit, than when the training examples are weighted.

Finally, it should be pointed out that AdaBoost may be extended to multiclass problems. A straightforward extention is the algorithm AdaBoost.M1 [13]. Other extentions have been proposed [12, 22]. Alternatively, boosting may be mixed with methods for decomposing a multiclass problem into binary subproblems, such as the one-versus-one or one-versus-all schemes, or error-correcting output codes [22].

```
Inputs:
  • labeled training examples: $\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
  • weight distribution defined over the training examples: $\mathbf{w} = (w_1, \ldots, w_n)$
  • weak learning algorithm: WeakLearn
  • number of weak learners to train: $T$
For $t = 1, \ldots, T$ do:
  1. Normalize the weights:
       $\mathbf{p}_t \leftarrow \frac{\mathbf{w}_t}{\sum_{i=1}^{n} w_{t,i}}$.
  2. Train weak learner $\mathcal{C}_t$ using the $\mathbf{p}_t$-weighted training examples; get back a decision rule $h_t : \mathbb{R}^p \to \{0; 1\}$.
  3. Estimate classification error $\epsilon_t$ of $h_t$:
       $\epsilon_t \leftarrow \sum_{i=1}^{n} p_{t,i} |h_t(\mathbf{x}_i) - y_i|$.
  4. Set $\beta_t \leftarrow \epsilon_t / (1 - \epsilon_t)$.
  5. Update the weight vector:
       $w_{t+1,i} \leftarrow w_{t,i} \beta_t^{1-|h_t(\mathbf{x}_i - y_i)|}$.
Output the hypothesis
  $h_f(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \left(\log \frac{1}{\beta_t}\right) h_t(\mathbf{x}) \geq \sum_{t=1}^{T} \left(\log \frac{1}{\beta_t}\right), \\ 0 & \text{otherwise.} \end{cases}$   (21)
```

Figure 1: The AdaBoost algorithm.

4. BOOSTING WITH BELIEF FUNCTIONS

4.1 Our credal boosting algorithm

Our algorithm is mainly inspired from AdaBoost (see Figure 2). The training of a sequence of classifiers $\mathcal{C}_1, \ldots, \mathcal{C}_T$ and the update of a weight distribution are similar; however, the evaluation of the accuracy of the classifier differs, as well as the combination rule employed. We are thus able to use weak learners that can produce any type of outputs, from mere decisions to probabilities, possibilities, or belief functions. However, for the sake of clarity, we will consider here classifiers that provide decisions in the form of categorical belief functions.

By construction, a weak learner is likely to make errors when evaluating data: its reliability in solving the classification problem may be arbitrarily low, should it for example concentrate on outliers of the distributions corresponding to the classes. Hence, we propose to take this reliability into account by contextually discounting the outputs of each weak learner, according to its ability to identify patterns from each class. The unreliability of the classifier may be easily estimated by the class-conditional training errors:

$$\alpha_k = \frac{1}{\sum_{\mathbf{x}_i \in \omega_k} p_{t,i}} \sum_{\mathbf{x}_i \in \omega_k} p_{t,i} |h_t(\mathbf{x}_i) - y_i|, \qquad (22)$$

where $p_{t,i}$ is the normalized weight associated at iteration t

with $\mathbf{x}_i$, and $h_t(\mathbf{x}_i)$ is the decision taken by $\mathcal{C}_t$ regarding $\mathbf{x}_i$.

As an example, consider a weak learner $\mathcal{C}_u$ that identifies all the training patterns $\mathbf{x}_i \in \omega_1$, but misclassifies 75 % of the training patterns $\mathbf{x}_j \in \omega_2$. Whenever it predicts ω_1, $\mathbf{x}$ may likely be from ω_2; hence, a proportion $\alpha_2 = 0.75$ of $m_u(\{\omega_1\})$ is transferred to $\{\omega_1, \omega_2\} = \Omega$. If it predicts ω_2 despite its difficulty to identify patterns from this class, $m_u(\{\omega_2\})$ is left unchanged. Note that, if $\mathcal{C}_t$ is completely unable to recognize class ω_2 ($\alpha_2 = 1$), we transfer the whole bbm $m_u(\{\omega_1\})$ to Ω: decision about the actual class of $\mathbf{x}$ is somehow left to other classifiers, which are best suited to this task than $\mathcal{C}_u$. Since misclassified examples have their weights increased, such classifiers will be trained in the next iterations.

The weak classifiers thus trained are obviously not independent from each other. Indeed, their training differs only by the weight distribution $\mathbf{w}_t$. Thus, the cautious rule may be best suited than Dempster's rule to combine their outputs. Another approach would be to evaluate the non-distinctness between two successive classifiers and choose an intermediate to combine their outputs. In [19], we proposed to use a distance between the bbas provided by two classifiers to assess their degree of distinctness. However, this approach may be quite cumbersome if the number of bbas is high. We rather propose to use the weight distributions used to train each of the classifiers, to compute a scalar s proportional to their distincness:

$$s_t = 1 - \frac{1 + \langle \mathbf{w}_{t-1}, \mathbf{w}_t \rangle}{2}, \tag{23}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product. This value s_t may then be directly used in Equation (14) to define an intermediate combination rule $\textcircled{T}_t$. Such a combination scheme, where each weak learner is associated with a combination rule, will be refered to as the adaptive rule in the remaining of the paper.

4.2 Reliability assessment

In some cases, further information about the reliability of the weak learners may be available. For example, decision trees [2, 16] partition the input space into regions, each one corresponding to a decision. In this case, it is possible to evaluate the reliability of the tree within each region, and thus to discount its outputs both contextually to the classes and to the location of the pattern being evaluated.

Consider, for example, a weak tree $\mathcal{C}_u$ with three leaves; the first one corresponds to patterns from ω_1 only, and the second one to patterns from ω_2 only; in the third one, however, classes are roughly balanced, although ω_1 prevails (which will thus be predicted in this region). Therefore, $\mathcal{C}_u$ will be reliable when predicting ω_1 or ω_2 whenever $\mathbf{x}$ is associated with the two first leaves. However, it will be completely unable to identify class ω_2 when $\mathbf{x}$ is associated with the third leave.

For each node $\mathcal{N}_p$ of the tree, the contextual reliability of the predictions may be evaluated in different ways. Let $n_{p,k}$ be the number of training patterns of ω_k associated with $\mathcal{N}_p$, and let $n_p = \sum_{k=1}^{K} n_{p,k}$ be the total number of training patterns falling into $\mathcal{N}_p$. As the decision associated with a node is the majority class, computing class-conditional training error rates will leave us with 0/1 discounting coefficients:

$$\alpha_{p,k} = \begin{cases} 0 & \text{if } n_{p,k} \geq n_{p,l}, \text{ for all } l = 1, \ldots, K, \\ 1 & \text{otherwise.} \end{cases} \tag{25}$$

```
Inputs:
  • labeled training examples $\mathcal{L}$
  • weight distribution $\mathbf{w}$ defined over the training data
  • weak learning algorithm: WeakLearn
  • number of weak learners to train: $T$
For $t = 1, \ldots, T$ do:
  1. Normalize the weights: get $\mathbf{p}_t$ from $\mathbf{w}_t$
  2. Train weak learner $\mathcal{C}_t$ using $\mathcal{L}$ and $\mathbf{p}_t$; get back a bba
     $m_t$, and the associated decision rule $h_t$.
  3. Estimate classification error $\epsilon_t$, as well as the class-
     conditional training errors, for $k = 1, \ldots, K$:
       $\alpha_{t,k} \leftarrow \frac{1}{\sum_{\mathbf{x}_i \in \omega_k} p_{t,i}} \sum_{\mathbf{x}_i \in \omega_k} p_{t,i} |h_t(\mathbf{x}_i) - y_i|.$
  4. Set $\beta_t \leftarrow \epsilon_t / (1 - \epsilon_t)$.
  5. (Optional) Estimate non-distinctness between $\mathcal{C}_t$ and
     $\mathcal{C}_{t-1}$:
       $s_t \leftarrow 1 - \frac{1 + \langle \mathbf{w}_{t-1}, \mathbf{w}_t \rangle}{2}.$
  6. Update the weight vector.
Output the bba
  $m_f\{\mathbf{x}\} = {}^{(\alpha)_1}m_1\{\mathbf{x}\} \bigoplus_{t=2}^{T} {}^{(\alpha)_t}m_t\{\mathbf{x}\},$    (24)
where $\oplus$ may be any rule suitable for combining bbas.
```

Figure 2: Our credal boosting algorithm.

Alternatively, we may imagine using the relative proportions of each class at node $\mathcal{N}_p$:

$$\alpha_{p,k} = \frac{n_{p,k}}{n_p}.$$

However, this solution will not allow us to have a discounting coefficient $\alpha_{p,k} = 1$, even if $\mathcal{C}_u$ is completely unable to identify class ω_k at node $\mathcal{N}_p$. Let k^* be the majority class index; instead, we may compute:

$$\alpha_{p,k} = \begin{cases} \dfrac{n_{p,k}}{n_{p,k^*}} & \text{if } k \neq k^*, \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

Hence, for any (non-majority) class ω_k, we have $\alpha_k \rightarrow 1$ as $n_{p,k} \rightarrow n_{p,k^*}$. Indeed, if only a few training examples $\mathbf{x}_i \in \omega_k$ fall into $\mathcal{N}_p$, our belief that the classifier is unable to identify ω_k is low; it increases with the relative size of ω_k with respect to the majority class. The discounting rate α_{p,k^*} is zero, since ω_{k^*} is always predicted at node $\mathcal{N}_p$.

Although using Equation (26) allows assessing the contextual reliability of a weak learner in a finer way, we will see in Section 5 that some situations will require switching to Equation (25) to avoid retrieving contradictory bbas m_t.

4.3 Learning from data with uncertain labels

In [4], a formalism for handling data with imprecise labels was proposed. It has been applied to k-nearest-neighbor classification, [4, 30, 7], decision trees [6, 29], and mixture models [3].

Suppose that the actual class of a training example could not be assessed with certainty. A belief function may be used to model this imperfect knowledge, rather than a mere class label. Various types of knowledge may thus be represented: in particular, probabilistic labels are retrieved with Bayesian bbas, and possibilistic labels using consonant bbas. A crisp label corresponds to a categorical bba, while the vacuous bba expresses total ignorance of the actual class of the pattern. Here, we suppose that each training pattern $\mathbf{x}_i$ is associated with a bba m_i. The training set becomes then $\mathcal{L} = \{(\mathbf{x}_1, m_1), \ldots, \mathbf{x}_n, m_n)\}$.

The classifier provides an output in the form of a belief function, that depends on the input patterns as well as on their associated bbas. For example, in the k-nearest-neighbors algorithm, the bba output for a test pattern $\mathbf{x}$ is computed by first discounting each bba m_i proportionally to the distance between $\mathbf{x}$ and $\mathbf{x}_i$, and then $\textcircled{\cap}$-combining the resulting bbas.

In the case of decision trees, a node $\mathcal{N}_p$ may be associated with a bba $m\{\mathcal{N}_p\}$, instead of the label of the majority class. This bba is computed from the bbas m_j labelling the training patterns $\mathbf{x}_j$ falling into $\mathcal{N}_p$. Details on the computation of $m\{\mathcal{N}_p\}$ may be found in [6, 29]. Then, during test phase, whenever a test pattern $\mathbf{x}$ falls into a leaf $\mathcal{N}_l$, it is associated with the bba $m\{\mathcal{N}_l\}$ instead of being classified into the majority class at $\mathcal{N}_l$.

5. EXPERIMENTS

In this section, we present experiments carried out on synthetic data as well as on real data sets. First, we compare our method to the AdaBoost algorithm, and we evaluate the impact of the combination rule as well as of the technique employed to assess the reliability of the weak learners. Then, we report experiments on a real-data set where data have uncertain labels.

5.1 Crisp-labeled data

Here, we present experiments on synthetic and real data, in which we compare our method using various combination rules to the AdaBoost algorithm. Throughout these experiments, we trained $T = 100$ classification trees as base weak learners. Trees were implemented using the CART algorithm [2]. The Gini index was used as purity measure for evaluating goodness of split. We limited the size of each tree by stopping the growth from a node $\mathcal{N}_p$ when the number n_p of associated training patterns satisfied $n_p \leq n/4$.

We generated a two dimensional synthetic dataset as follows. Each of the two classes is a mixture of 3 Gaussians, with covariance matrix:

$$\Sigma = \frac{1}{3}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

In class ω_1, the three Gaussians have centers $\mu_{11} = (1,3)$, $\mu_{12} = (3,4)$, and $\mu_{13} = (5,5)$; in class ω_2, the centers are $\mu_{21} = (3,2)$, $\mu_{22} = (5,3)$, and $\mu_{23} = (7,4)$. In each class, we generated 500 training examples and 100 test examples from each Gaussian. Table 6 presents the real datasets used, available from the UCI machine learning repository (http://archive.ics.uci.edu/ml/). We compared the performances of AdaBoost and of our method using the conjunctive rule, the cautious rule, the adaptive rule defined using Equation (23), and the mean combination rule. In the four latter cases, we assigned each test point to the class with maximum pignistic probability after combination of the weak learners. Table 7 shows the error rates obtained. For each dataset, the best results are underlined; results that were not judged significantly worse by a McNemar test [8] at level 5 % were printed in bold.

Table 6: Description of the real data sets.

	# features	training set size		test set size	
		ω_1	ω_2	ω_1	ω_2
breastcancer	30	127	214	85	143
ionosphere	34	134	75	90	51
sonar	60	67	58	44	39
spambase	57	1088	1673	725	1115

Table 7: Error rates obtained using AdaBoost, the conjunctive rule $\textcircled{\cap}$, the cautious rule $\textcircled{\wedge}$, an adaptive rule $\textcircled{\top}$, and the mean rule.

	AdaBoost	$\textcircled{\cap}$	$\textcircled{\wedge}$	$\textcircled{\top}$	mean
breastcancer	**8.8**	**8.8**	**$\underline{7.9}$**	**8.8**	**8.8**
ionosphere	**15.6**	**15.6**	**$\underline{14.9}$**	**$\underline{14.9}$**	**15.6**
sonar	**$\underline{30.1}$**	45.8	**38.6**	**36.1**	45.8
spambase	**11.4**	37.4	**10.7**	**$\underline{10.4}$**	37.4
synthetic data	11.5	15.0	**$\underline{8.7}$**	**$\underline{8.7}$**	15

The results obtained using the various methods are often close to each other. The best results are obtained using the cautious rule or the adaptive rule in four cases, and using AdaBoost in one case. AdaBoost does not perform significantly worse than the credal boosting algorithm (except on synthetic data). Remark that the cautious rule and the adaptive rule never perform significantly worse than the best results obtained. In addition, they are often very close to each other in terms of classification accuracy. Finally, Dempster's rule and the mean rule perform significantly worse than the best rule in three cases.

Figures 3, 4, 5 and 6 display the pignistic probabilities obtained on the synthetic dataset, using the conjunctive rule, the cautious rule, an adaptive rule, and the mean rule, respectively. The pignistic probabilities obtained using the conjunctive rule are very close to 0 or 1. Indeed, similar pieces of information are given more credit each time a source provides them. As a consequence, when the number of weak learners increases, the results obtained reflect the information they agree on. The pignistic probabilities obtained using the cautious rule and the adaptive rule better reflect the distribution of the data in the various regions of the space. This is also the case using the mean rule, although the results seem less representative of the distribution. This is confirmed by the poor results obtained, which are comparable to those of the conjunctive rule.

Finally, note that we also conducted experiments using a resampling of the training data, instead of weights. The results thus obtained were clearly worse that those reported here. Indeed, resampling data usually degenerates the distribution by removing the examples associated with lower weights. The tendency of the weak learners to overfit is then much higher. In this case, we used the method corresponding to Equation (26) to assess the reliability of the weak learners, in order to avoid retrieving the contradictory bba when combining their outputs.

5.2 Data with uncertain labels

5.2.1 Description of the data

Here, we report experiments conducted on real data with uncertain labels. The learning task was to identify different types of waveforms in sleep EEG data, and in particular to discriminate between K-complex signals and delta waveforms. Each data corresponds to 64 measurements of brain activity, separated by 2-second intervals: thus, it is described by 64 variables. K-complexes are difficult to identify, even by domain experts; hence, for each signal, five physicians were asked whether it contains a K-complex, by observing its graphical display. As the experts did not always agree on the nature of the signals, the labels thus obtained are uncertain. A thorough description of this problem may be found in [21].

We created a first dataset, refered to as eeg1, as follows. For a given training pattern, we estimated the probability of each class by computing the proportion of experts in favour of this class. Then, we determined the q-least committed bba associated with this probability distribution using Equations (6)-(7). Beside that, a crisp label was assigned to each signal by taking the class with maximum probability. We thus obtained 263 K-complex signals and 915 delta waveform signals. Then, we randomly selected 60 % of the K-complex data, and 60 % of the delta waveform data with certain label (that is, on which all experts agreed) for training, and the remaining for testing.

5.2.2 Experiments

We trained $T = 100$ weak learners as described in Section 5.1, except that tree growth was stopped at a given node $\mathcal{N}_p$ when n_p became less than $n/2$. We processed the data with AdaBoost, and our method using the rules already evaluated. For each rule, we used crisp labels and uncertain labels separately. Test patterns were classified as described in Section 5.1; decisions were compared to the crisp labels obtained as described in Section 5.2.1. The results are presented in Table 8.

Table 8: Error rates obtained on the eeg1 data, using AdaBoost, the conjunctive rule $\textcircled{$\cap$}$, the cautious rule $\textcircled{$\wedge$}$, an adaptive rule $\textcircled{T}$, and the mean rule.

	AdaBoost	$\textcircled{$\cap$}$	$\textcircled{$\wedge$}$	$\textcircled{T}$	mean
crisp labels	55.9	50.8	47.4	$\underline{\mathbf{44.9}}$	50.8
uncertain labels		$\underline{\mathbf{13.4}}$	24.0	24.0	$\underline{\mathbf{13.4}}$

When using crisp labels, the adaptive rule gives the best results. Dempster's rule and the mean rule give the best results with uncertain labels. The explanation is that these rules are far less sensitive to noise than the cautious rule or intermediate rules. One may remark a large difference between the two series of results. For explaining these differences and the overall bad performances obtained using crisp labels, we modified the dataset eeg1, by using the training data that were classified as K-complexes by three experts (i.e., the most uncertain ones) as test data. The results obtained on this new dataset eeg2 are presented in Table 9.

Table 9: Error rates obtained on the eeg2 data, using AdaBoost, the conjunctive rule $\textcircled{$\cap$}$, the cautious rule $\textcircled{$\wedge$}$, an adaptive rule $\textcircled{T}$, and the mean rule.

	AdaBoost	$\textcircled{$\cap$}$	$\textcircled{$\wedge$}$	$\textcircled{T}$	mean
crisp labels	36.4	$\underline{\mathbf{34.3}}$	37.6	36.0	$\underline{\mathbf{34.3}}$
uncertain labels		$\underline{\mathbf{23.7}}$	33.0	32.5	$\underline{\mathbf{23.7}}$

Clearly, the high error rates obtained on eeg1 are likely due to these K-complexes. These perturbating data could be either mislabeled delta waveforms, or outliers that influence the decision boundaries. Note that the results obtained here using uncertain labels are not as good as those obtained on eeg1, probably due to a lower training set size. As a conclusion, using uncertain labels increases robustness against perturbating information in training data, and therefore may be very interesting in cases where the the amount and the quality of the data are limited.

6. CONCLUSIONS AND PERSPECTIVES

In this article, we presented a variant of boosting within the theory of belief functions. Thus, weak learners that may produce any type of outputs, from mere decisions to probabilities, possibilities, or belief functions, can be used. Also, it allows for processing data whose labels are uncertain. At each iteration of our algorithm, a weak learner is trained using the training examples, which are weighted according to the difficulty the previous classifiers had to classify them. Additionally, the reliability of this weak learner to identify each class is evaluated. In the test phase, the outputs of each weak learner are interpreted as belief functions. These bbas are first discounted according to the reliability of the weak learner in identifying patterns from the various classes, using contextual discounting. Then, they are pooled using a suitable combination rule.

Experiments carried out on real datasets clearly demonstrate the interest of our method. On classical data, the accuracy of our algorithm is comparable to that of AdaBoost. When processing data with uncertain labels, results clearly show that taking into account the uncertainty on the actual class of the training patterns may increase robustness and accuracy. Thus, our approach may be a very interesting alternative when data pervaded with uncertainty are available.

In further work, we may investigate boosting of multiclass classifiers using belief functions. Our method might be directly extended to the multiclass case, as a variant of the AdaBoost.M1 algorithm. However, other algorithms have been proposed, that could also be modified using belief functions. Besides, we may experiment boosting credal classifiers, and in particular belief decision trees, that use the uncertain labels for building the tree. Finally, the strategy for constructing an adaptive combination scheme, and in particular the way of computing the non-distinctness between two classifiers, may be refined.

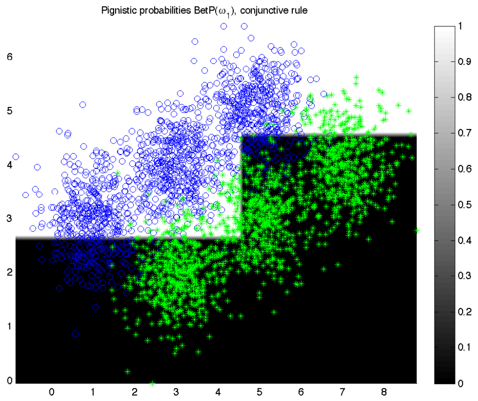

Figure 3: Pignistic probabilities $BetP(\omega_1)$ obtained using the conjunctive rule.

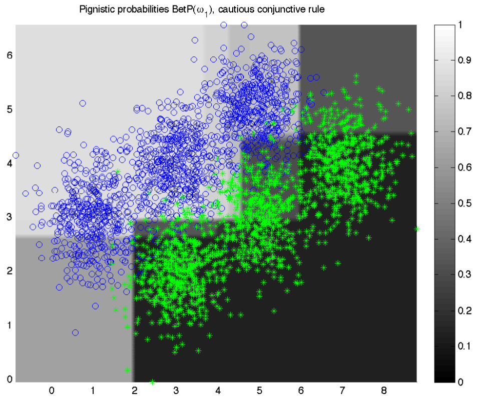

Figure 4: Pignistic probabilities $BetP(\omega_1)$ obtained using the cautious rule.

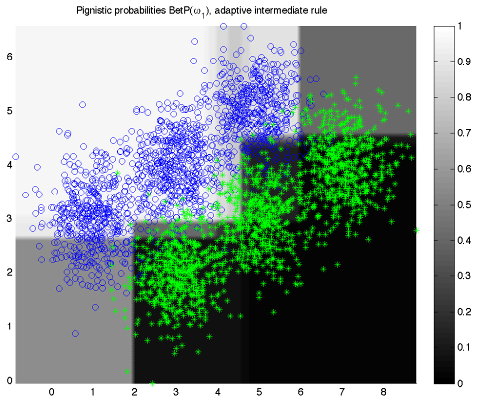

Figure 5: Pignistic probabilities $BetP(\omega_1)$ obtained using an adaptive rule.

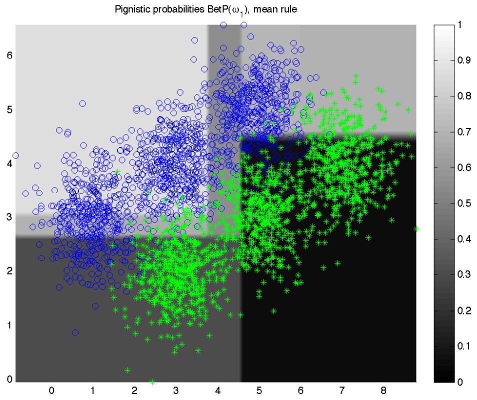

Figure 6: Decisions in favor of ω_1 obtained using the mean rule.

7. ACKNOWLEDGMENTS

The authors thank C. Richard and R. Lengellé from the University of Technology of Troyes for providing the EEG data, and the Foundation for Applied Neuroscience Research in Psychiatry (CHS de Rouffach, 68250 Rouffach, France) for authorizing their use.

8. REFERENCES

[1] Y. Bi, J. Guan, and D. Bell. The combination of multiple classifiers using an evidential reasoning approach. *Artificial Intelligence*, 172:1731–1751, 2008.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984.

[3] E. Côme, L. Oukhellou, T. Denœux, and P. Aknin. Learning from partially supervised data using mixture models and belief functions. *Pattern Recognition*, 42:334–348, 2009.

[4] T. Denœux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Transactions on System, Man and Cybernetics*, 25:804–813, 1995.

[5] T. Denœux. Conjunctive and disjunctive combination of belief functions induced by non-distinct bodies of evidence. *Artificial Intelligence*, 172:234–264, 2008.

[6] T. Denœux and M. Skarstein-Bjanger. Induction of decision trees from partially classified data using belief functions. In *Proceedings of SMC'2000*, Nashville, TN, 2000.

[7] T. Denœux and L. M. Zouhal. Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Sets and Systems*, 122:47–62, 2001.

[8] T. Dietterich. Approximate statistical tests for comparing supervised classification algorithms. *Neural Computation*, 10:1895–1923, 1998.

[9] D. Dubois and H. Prade. The principle of minimum specificity as a basis for evidential reasoning. *Uncertainty in Knowledge-based Systems*, pages 75–84, 1987.

[10] D. Dubois, H. Prade, and P. Smets. A definition of subjective possibility. *International Journal of Approximate Reasoning*, 48:352–364, 2008.

[11] J. François, Y. Grandvalet, T. Denoeux, and J.-M. Roger. Resample and combine: An approach to improving uncertainty representation in evidential pattern classification. *Information Fusion*, 4:75–85, 2003.

[12] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[13] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[14] E. P. Klement, R. Mesiar, and E. Pap. *Triangular norms.* Kluwer Academic Publishers, Dordrecht, 2000.

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

[16] R. Quinlan. *C4.5: Programs for machine learning.* Morgan Kaufman Publishers, 1993.

[17] B. Quost, T. Denœux, and M.-H. Masson. Pairwise classifier combination using belief functions. *Pattern Recognition Letters*, 28:644–653, April 2007.

[18] B. Quost, T. Denœux, and M.-H. Masson. Adapting a combination rule to non-independent information sources. In L. Magdalena, M. Ojeda-Aciego, and J. Verdegay, editors, *Proceedings of the 12th IPMU Conference*, pages 448–455, Málaga, Spain, 2008.

[19] B. Quost, M.-H. Masson, and T. Denœux. Refined classifier combination using belief functions. In *Proceedings of the 10th International Conference on Information Fusion*, pages 776–782, Cologne, Germany, 2008.

[20] M. Reformat and R. Yager. Building ensemble classifiers using belief functions and owa operators. *Soft Computing*, 12:543–558, 2008.

[21] C. Richard. *Une méthodologie pour la détection à structure imposée. Applications au plan temps-fréquence.* PhD thesis, Université de Technologie de Compiègne, 1998.

[22] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of statistics*, 26(5):1651–1686, 1998.

[23] G. Shafer. *A mathematical theory of evidence.* Princeton University Press, Princeton, NJ, 1976.

[24] P. Smets. Combining non-distinct evidences. In *Proceedings of the International Conference of the North American Fuzzy Information Processing Society (NAFIPS '86)*, pages 544–549, New Orleans, USA, 1986.

[25] P. Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:447–458, 1990.

[26] P. Smets. Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning*, 9:1–35, 1993.

[27] P. Smets. Data fusion in the transferable belief model. In *Proceedings of the 3rd International Conference on Information Fusion*, pages 21–33, Paris, France, 2000.

[28] P. Smets and R. Kennes. The transferable belief model. *Artificial Intelligence*, 66:191–234, 1994.

[29] P. Vannoorenberghe and T. Denœux. Handling uncertain labels in multiclass problems using belief decision trees. In *Proceedings of the 5th IPMU Conference*, pages 1919–1926, Annecy, France, 2002.

[30] L. M. Zouhal and T. Denœux. An evidence-theoretic k-nn rule with parameter optimization. *IEEE Transactions on System, Man and Cybernetics*, 28:263–271, 1998.