Different paradigms for choosing sequential reweighting algorithms

Gilles Blanchard*

Abstract

Analyses of the success of ensemble methods in classification have pointed out the important role played by the "margin" distribution function on the training and test sets. While it is acknowledged that one should generally try to achieve high margins on the training set, the more precise shape of the empirical margin distribution function one should favor in practice is subject to different approaches.

We first present two concurrent philosophies for choosing the empirical margin profile, one we call "minimax margin paradigm" and the other "mean and variance paradigm". The best known representative of the first paradigm is the AdaBoost algorithm, and this philosophy has been shown by several other authors to be closely related to the principle of the SVM. On the other hand, we show that the second paradigm is very close in spirit to Fisher's linear discriminant (in a feature space).

We construct two boosting-type algorithms, very similar in their form, dedicated to one or the other philosophy. We consequently derive by interpolation a very simple family of iterative reweighting algorithms that can be understood as different tradeoffs between the two above paradigms, and argue from experiments that this can allow for a suitable adaptivity to different classification problems, particularly in the presence of noise and/or excessive complexity of the base classifiers.

Keywords: ensemble methods, large margin classification, boosting.

*Département de mathématiques, Université Paris-Sud, Orsay, France; and Fraunhofer FIRST, Kekuléstr. 7, 12489 Berlin, Germany. E-mail: blanchar@first.fhg.de

1 Introduction

1.1 Motivation and previous work

In classification problems, recent work has given important attention to various procedures consisting in building several different classifiers of the same type (such as e.g. decision trees or RBF networks), usually obtained by perturbations of the training set, and then combining these together to form an aggregated classifier, usually by some (perhaps weighted) voting process.

The popularity of this type of method comes from their reported good performance in very different practical problems (see e.g. Dietterich (2000); Amit and Geman (1997); Viola and Jones (2001), and the review of Meir and Rätsch (2003)): usually aggregated classifiers outperform significantly, and sometimes in a spectacular way, single classifiers of the same type.

The theoretical approaches trying to explain the success of these methods have focused on an important quantity called the "margin" (representing, for a given example, the proportion of votes in favor of the true class vs. the wrong one). To this regard, the paper of Schapire, Freund, Bartlett and Lee (1998) should be considered as a milestone, by giving a qualitative (albeit not really practical) bound of the generalization error by quantities involving the empirical cumulative distribution function of the margins and the complexity of the base space.

However, while most authors seem to emphasize that the margins of the training set should be used in some way for guiding the construction of the aggregated classifier, it is not necessarily clear precisely what criterion should be used. The present paper is aimed at investigating this issue and its goal is twofold:

1. Identify within a common framework two concurrent existent paradigms (or philosophies) concerning the choice of this criterion and their relation to classical statistical principles. We emphasize that these criteria favor different shapes of the empirical margin distribution. This allows for a unifying point of view.

2. Derive a boosting-type (i.e. based on iterative reweighting) algorithm able to interpolate between these two paradigms, and therefore to adapt to different situations where one or the other (or some intermediate choice) is more suitable.

The organization of the paper is as follows. Sections 2 and 3 are devoted to a presentation of the two paradigms. The first paradigm (presented in

section 2) is linked to the "Large Margin Classifiers" philosophy and it is usually argued that it is the underlying principle of the popular AdaBoost algorithm. The second paradigm (presented in section 3) may seem a little less familiar. It is based on analyses put forward by Breiman (2001) and Amit and Blanchard (2001), and we actually show its similarity to Fisher's linear discriminant. (When considered in a linear feature space, the large margin paradigm and the Fisher discriminant paradigm have been compared by Mika (2002).)

For each these two paradigms, we put forward an iterative reweighting ensemble algorithm dedicated to finding a good empirical margin distribution according to the associated paradigm. Strikingly, these two algorithms share a very similar form, namely: the shape of the reweighting function (as a function of the current margin) is the same in the two algorithms (it is formed of a constant piece and a linear piece), only the choice of two parameters differs (the abscissa of the change-point and the height of the constant part).

In section 4, we put forward an algorithm that we understand as a heuristic interpolation between these two paradigms, allowing to sample a family of candidate empirical margin distributions among which one is finally picked by cross-validation. This algorithm is tested on benchmark datasets with two types of base classifiers (classification trees and RBF networks), and compared to other popular ensemble methods (Random Forests (Breiman, 2001) and AdaBoost-Reg, a regularized AdaBoost method (Rätsch, Onoda and Müller, 2001)).

We emphasize that this algorithm is *simple* and *cheap* from a computational point of view (apart from the computational burden for the weak learner itself – the latter is of course variable since the weak learner can be chosen arbitrarily), and that by its very nature, it can be applied to any weak learner. These two features (simplicity and applicability to any weak learner) are at the root of the success of AdaBoost, and we argue that the algorithm proposed here retains this two features while exhibiting improved results in those cases where AdaBoost would be overfitting. The experimental results also show clearly improved performance with respect to Random Forests when RBF networks are used as base classifiers. Finally the algorithm is on par with AdaBoost-Reg while being noticeably simpler.

The approach in this paper is mainly heuristic. Our belief is that, while theoretical bounds on the generalization error are an extremely important tool to give general guidelines and a qualitative appraisal of the behavior of algorithms, they are, as far as statistical learning is concerned, seldom accurate enough to give a quantitatively directly usable tool for model selection. A study of theoretical bounds related to the problems studied here

was presented elsewhere (see Blanchard (2001) and Blanchard (2003), improving the results of Schapire et al. (1998) and Breiman (1998b)), but we believe in practical applications it is also always necessary to additionally adopt heuristic guidelines (what we call "paradigms") — the latter of course based on sound arguments. In this regard, we think in particular that the theoretical bounds of Schapire et al. (1998) are in a sense "agnostic" as to what shape of empirical margin distribution should be preferred in practice, although the authors of that paper seem to favor what we call here the minimax paradigm. Our suggestion in the present paper is finally to try maintaining an agnostic attitude, and find a means (preferably as easy as possible from an algorithmic point of view) to sample several different good candidate margin "profiles" and pick among them by cross-validation, as is argued in section 4.

1.2 General framework and notations

We will consider a two-class classification problem where $X \in \mathcal{X}$ is the observed variable (in all the numerical simulations presented here $\mathcal{X}$ will be a real vector space of finite dimension), and $Y \in \{-1, 1\}$ is the class of the observation. The set of classifiers $\mathcal{F}$ is the set of (measurable) functions f from $\mathcal{X}$ to $\{-1, 1\}$. We assume that couples (X, Y) are drawn according to some underlying probability distribution P, and for any $f \in \mathcal{F}$, the generalization error of $f \in \mathcal{F}$ is defined by

$$\mathcal{E}(f) = E_P[\mathbf{1}\{f(X) \neq Y\}].$$

The training set is a finite set of observations $(X_i, Y_i)_{i=1\ldots N}$ used to build classifiers. We suppose that we have at hand an automated learning algorithm $\mathcal{W}$ which, given the training set and a set of nonnegative weights $(\omega_i)_{i=1\ldots N}$, outputs a classifier f. This algorithm is generally called the *weak learner* and is considered as a "black box" on which we have no influence. It is usually assumed that the outputs of the weak learner belong to some subset $\mathcal{H} \subset \mathcal{F}$ of the set of all possible classifiers, and that $\mathcal{H}$ is of small "size" in terms of complexity (for instance, it has finite VC dimension). The set $\mathcal{H}$ is called the "set of base classifiers". The weak learner $\mathcal{W}$ can be for example any classical learning algorithm[1], like a neural net, a decision tree,

[1]In the case the original algorithm is not well-suited for dealing with weighted data, one generally takes the weights into account by first randomly re-sampling from the training set using the prescribed weights; we will consider this as part of the weak learning procedure.

a linear discriminant... Ideally, the algorithm $\mathcal{W}$ can be thought as a procedure minimizing the *weighted* training error over the base classifier space $\mathcal{H}$. It is often a reasonable approximation although almost never exactly true in practice.

1.3 Voting methods and sequential reweighting schemes

"Aggregation", "voting" or "ensemble" methods are different names for a general family of procedures whose goal is to improve the weak learner's generalization error by combining several of the base classifiers. Informally speaking, such a combination is obtained by building (according to some protocol to be made precise) a finite set of base classifiers $f_1, \ldots, f_T$, along with some associated family of positive coefficients $(\alpha_i)_{i=1\ldots T}$.

The aggregated function $F_{\overline{\alpha}} : \mathcal{X} \to \mathbb{R}$ is then defined by

$$F_{\overline{\alpha}}(x) = \sum_{i=1}^{T} \alpha_i f_i(x)$$

and the associated classification function $\overline{F}_{\overline{\alpha}}$ is obtained by taking the sign of $F_{\overline{\alpha}}$ (if $F_{\overline{\alpha}}(x) = 0$, the class is determined arbitrarily), which corresponds to a (weighted) majority vote among the considered classifiers. We also define the normalized aggregated function $G_{\overline{\alpha}} : \mathcal{X} \to [-1, 1]$ by

$$G_{\overline{\alpha}}(x) = \left(\sum_{i=1}^{T} \alpha_i \right)^{-1} F_{\overline{\alpha}}(x).$$

In the sequel, to lighten notations we will conceive $\overline{\alpha}$ as a positive measure with finite support on the set $\mathcal{F}$, thus implying that $\overline{\alpha}$ alone entirely determines the aggregated functions above. We will refer to $\overline{\alpha}$ as a "combination" (of classifiers) and denote $|\overline{\alpha}| = \sum_{f \in \mathcal{F}} \alpha_f$.

Among ensemble methods, a wide subfamily is given by *sequential reweighting schemes* (SRS, also called "ARCing" methods (Breiman, 1998a)). Sequential reweighting schemes build aggregate classifiers according to the following general principle: call the weak learner sequentially, and iteratively update the weights (ω_i) at each iteration, depending on past errors. The coefficient (α_t) of classifier f_t at iteration t is also computed on the spot depending on past errors.

1.4 Margins

The literature devoted to ensemble methods has given particular attention to the study of "margins" (see e.g. Schapire et al., 1998; Koltchinkskii and

Panchenko, 2002; Breiman, 1998b; Smola, Bartlett, Schölkopf and Schuurmans, 2000; Blanchard, 2003; Meir and Rätsch, 2003), which will also be the main object of interest in the present paper.

The (normalized) margin function $M_{\overline{\alpha}} : \mathcal{X} \times \{-1, 1\} \to [-1, 1]$ associated to a combination $\overline{\alpha}$ is defined as

$$M_{\overline{\alpha}}(x, y) = y G_{\overline{\alpha}}(x) = y|\overline{\alpha}|^{-1} \sum_{f \in \mathcal{F}} \alpha_f f(x).$$

Basically, for a given example (x, y), the margin function tells us with what confidence the "vote" obtained with combination $\overline{\alpha}$ is in favor of the correct class. The example is misclassified iff the margin is nonpositive; for a positive, but close to zero margin, the example is correctly classified, but it is a "close call".

It is intuitively clear that when building a combination of base classifiers belonging to $\mathcal{H}$, one would be happy to have margins as high as possible on the training examples: this should bring more stability to the decision function. Theoretical bounds based on margins give a qualitative support to this intuition (see references cited earlier). However, if $\mathcal{H}$ is of limited complexity (which surely is necessary to prevent overfitting), it is not possible to increase simultaneously the margins on all examples, as we expect that no base classifier achieves perfect classification. Thus, increasing the margin for some training examples will also mean decreasing the margin of some others. The main problem is then, what kind of tradeoff is one willing to make in this situation. Our focus on this paper will concern the choice of classifier combination based on different "philosophies" concerning the shape the margin distribution of the training set should have.

2 The minimax margin philosophy

2.1 The AdaBoost algorithm and the Support Vector Machine

The best known SRS, which, thanks to its excellent practical performance, has renewed the interest in voting methods, is probably the AdaBoost algorithm (Freund and Schapire, 1996a). We will not describe this algorithm in all detail here, but we will review a few important facts (coming mainly from Schapire et al. (1998)) to understand its behavior. Let $\overline{\alpha}_t$ denote the combination obtained up to iteration t. It can be shown that the AdaBoost algorithm is such that the weight $\omega_{i,t+1}$ of example i at step $t+1$ is proportional to $\exp(-|\overline{\alpha}_t| M_{\overline{\alpha}_t}(X_i, Y_i))$. We write it this way to emphasize that the

unnormalized margin is used inside the exponential. It means that iteration $t+1$ will mainly concentrate on those examples that have the lowest margins and maybe almost "forget" the examples which have a relatively higher margin, all the more when t becomes larger since $|\overline{\alpha}_t|$ is growing with t (see also the related analysis of Onoda, Rätsch and Müller (1998)). This remark, along with other arguments, has led Schapire et al. (1998) to claim that AdaBoost is iteratively trying to find a combination $\overline{\alpha}$ for which the minimum margin on the training set is maximum, that is, approximately solves the optimization problem

$$\max_{\overline{\alpha}} \min_{i} M_{\overline{\alpha}}(X_i, Y_i). \tag{1}$$

We call this way of choosing the combination $\overline{\alpha}$ the "minimax margin" philosophy. Schapire et al. (1998) (see also e.g. Rätsch, Mika, Schölkopf and Müller, 2002) note also that there is a close connection between this philosophy and the principle underlying the (hard margin) Support Vector Machine. For a hard margin SVM, a linear classifier is chosen so as to maximize the minimum geometrical margin over the training examples (see Figure 1). Thus the general philosophy for choosing the classifier is comparable in these two methods.[2] This analogy also has led other authors to adapt the "soft margin" SVM principle to the Boosting case to avoid overfitting (Rätsch et al., 2001).

2.2 Theory of games and Blackwell's strategy

The minimax margin principle also has a natural interpretation in the framework of the theory of games. Namely, consider a "game" where the first player chooses an example (X_i, Y_i) of the training set, the second player chooses a classifier f from the set $\mathcal{H}$, and the second player wins (and the first player loses) 1 unit if the example is correctly classified and loses 1 unit otherwise, that is, player 2 wins $f(X_i)Y_i$. Then, the equilibrium mixture strategy $\overline{\alpha}^*$ of player 2 exactly corresponds to the solution of (1), and the value of the game is the minimax margin. This point of view has actually been one of the initial motivations for boosting methods (Freund and Schapire, 1996b; Breiman, 1998b).

It has several interesting consequences, like the following fact which is an easy consequence of the minimax theorem: if, for any set of weights (ω_i) on the training sample, there exists a classifier $f \in \mathcal{H}$ with weighted error

[2] One of the main differences between the two is that the SVM margin is defined with a Euclidian metric whereas the boosting margin is measured in terms of a L_1 metric.

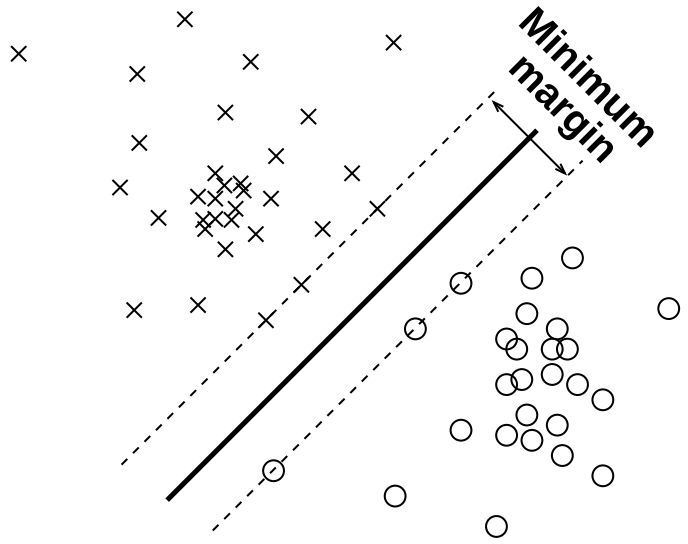

Figure 1: A pictorial idealization of the minimax margin paradigm for a Support Vector Machine.

strictly less than 0.5, then there exists an aggregated classifier with zero training error.

If γ denotes the value of the game, then it can be shown (Schapire et al., 1998) that the AdaBoost algorithm will asymptotically produce a combination $\overline{\alpha}$ whose minimum margin on the training set is at least $\gamma/2$ (as a consequence, if $\gamma > 0$, AdaBoost will asymptotically produce a classifier with zero training error). However, it is up to our knowledge unknown if the AdaBoost algorithm will, or not, asymptotically attain the minimax margin γ.

If we want to follow the minimax margin philosophy to the end, we would like to have an algorithm for which we are sure that the minimax margin will be asymptotically attained. Various iterative methods have been proposed by different authors to this end (Breiman, 1998b; Rätsch and Warmuth, 2001). However, there also exists an algorithm known in game theory as Blackwell's strategy which is guaranteed to attain the minimax margin under certain hypotheses (see Blackwell (1956) and some recent related results (Hart and Mas-Colell, 2001)). This algorithm is also an SRS which we describe in Figure 2.

This algorithm is extremely simple to describe: a "regret" vector R_t is kept along which keeps track, for each example, of the difference between

1. For $i = 1, \ldots, N$, initialize weights $\omega_{i,1} = 1/N$ and the "regret vector" $R_{i,1} = 0$.

2. Iteration t: call the weak learner $\mathcal{W}$ with the weights $(\omega_{i,t})$, resulting in classifier f_t. Let ε_t denote the *weighted* error of f_t.

3. For $i = 1, \ldots, N$, update regret vector the following way: $R_{i,t+1} = R_{i,t} + \mathbf{1}\{f_t(X_i) \neq Y_i\} - \varepsilon_t$.

4. Update weights the following way: for $i = 1, \ldots, N$, $\omega_{i,t+1} \propto (R_{i,t+1})_+$, where $(\cdot)_+$ is the positive part; then renormalize so that the weights sum to 1.

5. If $t < T_{max}$, proceed to next iteration and point (2). If iteration T_{max} is reached, take for the aggregated classifier the simple uniform average (vote) of the f_t's, $t = 1, \ldots, T$.

Figure 2: Blackwell's strategy applied in our setting.

the number of times it has been misclassified and the sum of the weighted errors. For the next iteration, the weights are taken proportional to the positive part of R_t. After reaching a fixed number of iterations, just take the uniform average of all the base classifiers thus obtained.

For this procedure, the following theorem (essentially coming from Hart and Mas-Colell (2001)) holds in the case where $\mathcal{H}$ is finite:

Theorem 1. *Let γ denote the minimax margin and ε_t denote the weighted error at step t.*

Assume that $\mathcal{H}$ is finite and that the weak learning algorithm $\mathcal{W}$ outputs the classifier $f \in \mathcal{H}$ having the least weighted error when it is called. Then the following assertions holds true:

$$(i) \forall i = 1, \ldots, N \qquad \limsup_{t \to \infty} \frac{1}{t} R_{i,t} \leq 0;$$

$$(ii) \forall i = 1, \ldots, N \qquad \liminf_{t \to \infty} \frac{1}{t} \sum_{k=1}^{t} f_k(X_i) Y_i \geq \gamma.$$

Proof. Point (i): see Hart and Mas-Colell (2001), section 3. Note that this point remains true if the weak learner $\mathcal{W}$ satisfies the weaker hypothesis that it always outputs a classifier having weighted error strictly less than 0.5. Point (ii) is then straightforward (see proof in appendix). □

Note that $\frac{1}{t}\sum_{k=1}^{t} f_t(X_i)Y_i$ is exactly the margin of the aggregated classifier at step t on example i. Hence under the conditions of the theorem the algorithm will converge to a solution of the minimax problem (1).

As such, Blackwell's strategy may seem of little interest, since there exists other and probably faster methods to reach the minimax margin. However, besides the advantage of its simplicity, it has the property (which will be of interest to us in the sequel) that the weights at step t are proportional to a 2-parameter piecewise linear function $\Phi_{a,b}$ of the margin of the current combination $\overline{\alpha}_t$; more precisely

$$\Phi_{a,b}(x) \doteq (x-a)_- + b, \tag{2}$$

where $(\cdot)_-$ denotes the negative part ($(x)_- = -x$ if $x < 0$ and is zero otherwise). The choice of parameters at step t leading to Blackwell's strategy is $a_t = 1 - \frac{2}{t-1}\sum_{k=1}^{t-1}\varepsilon_k, b_t = 0$ (where ε_k is the weighted error of classifier f_k). This is easily seen from the description of the algorithm on Fig. 2 and using the fact that $\mathbf{1}\{f(x) \neq y\} = \frac{1}{2}(1 - f(x)y)$ for a binary classifier function f (taking values in $\{-1, 1\}$).

3 The mean-and-variance philosophy

3.1 Chebyshev's inequality and Fisher's linear discriminant

Another way of understanding the performance of voting methods, proposed e.g. by Amit and Geman (1997); Amit and Blanchard (2001) and Breiman (2001), relies on looking at the two first moments of the margin function and a simple Chebyshev's inequality. The main idea of this analysis can be summed up in the following inequality:

Theorem 2. *Assume that combination $\overline{\alpha}$ is such that $E[M_{\overline{\alpha}}(X,Y)] > 0$. Then the following inequality holds:*

$$\mathcal{E}(\overline{F}_{\overline{\alpha}}) \leq \frac{Var[M_{\overline{\alpha}}(X,Y)]}{E[M_{\overline{\alpha}}(X,Y)]^2} \tag{3}$$

Proof. The proof is immediate:

$$\begin{aligned}
\mathcal{E}(\overline{F}_{\overline{\alpha}}) &\leq P[M_{\overline{\alpha}}(X,Y) \leq 0] \\
&= P[M_{\overline{\alpha}}(X,Y) - E[M_{\overline{\alpha}}(X,Y)] \leq -E[M_{\overline{\alpha}}(X,Y)]] \\
&\leq P[(M_{\overline{\alpha}}(X,Y) - E[M_{\overline{\alpha}}(X,Y)])^2 \geq E[M_{\overline{\alpha}}(X,Y)]^2] \\
&\leq Var[M_{\overline{\alpha}}(X,Y)]/E[M_{\overline{\alpha}}(X,Y)]^2,
\end{aligned}$$

where the last line results from Markov's inequality. □

Note that there exists various refinements and variants of this inequality. Amit and Geman (1997) and Amit and Blanchard (2001) derive a slightly different analysis for multiclass problems, and it is shown that the variance factor can mainly be interpreted as the average covariance, *conditional on class*, of two base classifiers in the aggregate. In (Devroye, Györfi and Lugosi, 1996, p. 41), a somewhat tighter bound based on the same quantities can be found.

What is the significance of this inequality? One the one hand, it is certainly true that Chebyshev's inequality is but very coarse and that the above bound cannot be very tight. On the other hand, one can argue the following: we expect that the average and variance of the margin function are two elementary statistical quantities which should be able to be estimated from their empirical counterparts on the training set without excessive error (for a theoretical study supporting this point, see Blanchard (2001)).

It is therefore more relevant to understand this inequality as an indication pointing towards what we should try to do in practice: find some compromise between mean and variance of the margin function (we want high mean but low variance), based on their empirical estimations.

Furthermore, another argument supporting the interest of this approach comes from the following useful comparison with Fisher's discriminant. It is indeed interesting to note that, while the minimax margin philosophy was comparable to a SVM classifier in a linear classification framework, the mean-and-variance philosophy is very comparable to the principle underlying Fisher's linear discriminant classifier. Namely, the following simple theorem makes the link apparent by showing that, when we apply the same line of reasoning (based on Chebyshev's inequality) to a Euclidian classification setup, we naturally find Fisher's discriminant function.

Theorem 3. *Consider a classification in a Euclidian space F; let m_1 and m_{-1} be the average values of class 1 and -1. For $a \in F$, let f_a be the linear classifier defined by $f_a(x) = sign(\langle a, (x-(m_1+m_{-1})/2)\rangle)$. Define the margin function $M_a(x,y) = y\langle a, (x-(m_1+m_{-1})/2)\rangle$. Then the following inequality holds for any a such that $\langle a, m_1\rangle > \langle a, m_{-1}\rangle$:*

$$\mathcal{E}(f_a) \leq \frac{Var[M_a]}{E[M_a]^2} = 4\frac{p_1 s_1^2 + p_{-1} s_{-1}^2}{\langle a, (m_1 - m_{-1})\rangle^2}, \tag{4}$$

where $p_i = P(Y = i)$ and $s_i^2 = Var[\langle a, X\rangle | Y = i], i = -1, 1$.

Proof. The first inequality results from Chebyshev's inequality similarly to theorem 2. The second equality is just a little calculation: for $i = -1, 1$ we

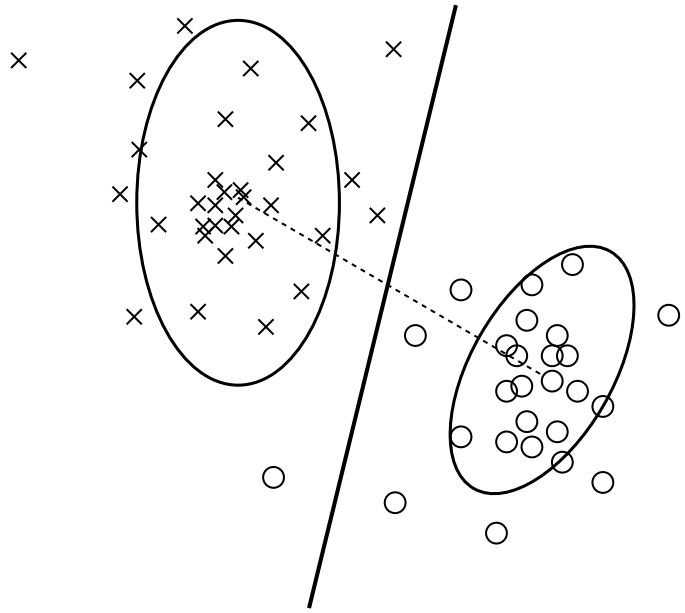

Figure 3: A pictorial idealization of the mean-and-variance paradigm for Fisher's linear discriminant. Compare with fig. 1 (the data points are the same).

note that:

$$E[M_a|Y=i] = \langle a, (m_1 - m_{-1})/2 \rangle,$$

and thus

$$E[M_a] = p_1 E[M_a|Y=1] + p_{-1} E[M_a|Y=-1] = \langle a, (m_1 - m_{-1})/2 \rangle,$$

and

$$Var[M_a] = E[Var[M_a|Y]] + Var[E[M_a|Y]] = p_1 s_1^2 + p_{-1} s_{-1}^2 + 0.$$

□

Thus, a very similar mean-variance analysis performed in a linear classification setup leads us to inequality (4), where the right-hand side is exactly the inverse of Fisher's discriminant function, used precisely to choose the optimal direction a of the linear classifier (see Fig. 3). Note that Fisher's linear discriminant is traditionally justified by considering the idealized situation where the classes have a Gaussian distribution with identical covariance matrix, but this is rarely the case in practice; hence, it can be seen in more generic situations as essentially based on the mean/variance approach.

The point of that philosophy is to try to be suitable in situations where the two classes form two overlapping clusters, i.e. when there exist noisy regions, whereas the "minimax" philosophy is more suited for situations where the two classes are well-separated, although maybe by an irregular boundary. A related analysis appears in Mika (2002), to compare the principles of SVMs and Fisher discriminant in a linear feature space.

3.2 The sequential lower-variance minimization

In this section we discuss how to implement in practice the method suggested by the mean-and-variance philosophy. First, if we believe that Chebyshev's inequality and the resulting inequality (3) is a good indication towards what should be done in practice, then an immediate remark is that we can replace in this inequality the variance of the margin by what we call its "lower-variance":

$$Var_{\_}[X] \doteq E[((X - E(X))_{\_})^2],$$

where $(\cdot)_{\_}$ denotes the negative part (i.e. $(x)_{\_} = -x$ when $x < 0$ and $(x)_{\_} = 0$ otherwise). This makes sense since we only want to take into account the lower deviations of the margin in order to bound the probability that it becomes negative.

Now, the second point is that we would like to find an SRS (see 1.3) corresponding to the mean-and-variance philosophy. It has been noted by several authors (Breiman, 1998b; Frean and Downs, 1998; Friedman, Hastie and Tibshirani, 2000, between others) that the AdaBoost algorithm can be interpreted as a gradient descent with an exponential cost function (this has led to proposing other SRSs based on different cost functions). We propose to follow this principle as applied to a cost function which reflects our present philosophy. A first naive choice would just be to choose the right hand side of (3). Two arguments lead to choose another solution. First, we expect inequality (3) to be too coarse to capture in all generality the optimal tradeoff between mean and variance. We therefore suggest to introduce a one-parameter family of cost functions corresponding to different possible tradeoffs (and the parameter will be selected by cross-validation). Second, our first experiments showed that a cost function defined as a ratio between mean and variance yielded quite unstable results and we prefer an additive cost function using this two quantities. Finally, we choose the following cost function depending on the parameter $\beta > 0$:

$$C_\beta(G_{\overline{\alpha}}) = |\overline{\alpha}|^{-1} \left(\sqrt{Var_{\_}[M_{\overline{\alpha}}]} - \beta E[M_{\overline{\alpha}}] \right) \tag{5}$$

(the square root is for homogeneity). The normalization by $|\overline{\alpha}|^{-1}$ is necessary if we want a scale invariant cost (the scale does not change the final classification function). Note that this is very similar to a cost function earlier suggested by us (Amit and Blanchard, 2001).

Now, what SRS will we derive from that cost function if we apply a gradient descent to it? First, we note that this cost function is convex in the simplex $\{|\overline{\alpha}| = 1\}$ (see appendix for a proof), so that gradient descent is a legitimate method to minimize it. We mainly need to compute the gradient of the cost along the direction corresponding to adding a new classifier to the mixture. Denote by $G_0 = G_{\overline{\alpha}}$ the normalized aggregated function corresponding to our current mixture $\overline{\alpha}$; consider adding classifier f to the mixture with a small weight, resulting in the normalized aggregated function $G_t = (1-t)G_0 + tf$ for some small t. We thus want to compute $\partial C_\beta(G_t)/\partial t|_{t=0}$ and minimize this quantity as a function of f. We then have the following property (see proof in Appendix):

Theorem 4. *Denote $\mathcal{E}_f(x,y) = \mathbf{1}\{f(x) \neq y\}$, $w(x,y) = (M_{\overline{\alpha}} - E[M_{\overline{\alpha}}])_-$. Then the classifier f minimizes $\partial C_\beta(G_t)/\partial t|_{t=0}$ iff it minimizes the quantity*

$$J(\overline{\alpha}, f) = E[w(X,Y)\mathcal{E}_f(X,Y)] + \left(\beta\sqrt{E[w(X,Y)^2]} - E[w(X,Y)]\right) E[\mathcal{E}_f(X,Y)]. \tag{6}$$

Keep in mind that the goal of the algorithm is to minimize the empirical cost function, i.e., when the variance and expectation operators in equation (5) are taken under the empirical distribution. Similarly, in the above definition of $J(\overline{\alpha}, f)$ the expectation is to be understood under the empirical distribution. Now, the quantity $J(\overline{\alpha}, f)$ can be interpreted as a weighted error: it can be put under the form $J(\overline{\alpha}, f) = E[(w(X,Y) + C)\mathcal{E}_f(X,Y)]$, with a constant C depending on $\overline{\alpha}$ but not on (X,Y). Therefore minimizing $J(\overline{\alpha}, f)$ with respect to f amounts to minimizing the weighted error of f when example (X,Y) is given a weight proportional to $w(x,y) + C$.

It can actually happen, if β is small, that some weights are negative because of the negative part in the constant term C. This is not a problem in practice, since we can then flip the label of the corresponding example and take the absolute value of the weight for the next training set. However, if one chooses $\beta \geq 1$, then obviously the weights are always nonnegative by Jensen's inequality.

As a consequence, the sequential lower-variance minimization algorithm goes as follows: call at each step the weak classifier with the weights corresponding to (6), and add the output f to the current combination with

coefficient 1. Note that in the AdaBoost algorithm, the coefficient given to the classifier is again the result of an optimization in the direction of f. But since this is not necessary to perform a simple gradient descent, we prefer this simpler uniform average here.

4 An algorithmic interpolation of the two "philosophies"

4.1 The interpolated algorithm

Now that we have explored the two paradigms about the margin distribution, we notice that the two algorithms built in the previous sections (Blackwell's strategy applied to classification and Sequential Lower-Variance Minimization, SLVM for short) have the interesting common point that the weights given to the training examples at each step t are in both cases of the form $\omega_{t,i} \propto \Phi_{a_t,b_t}(M_{\overline{\alpha}_{t-1}}(X_i, Y_i))$, where $\overline{\alpha}_{t-1}$ is the current combination at the beginning of step t and $\Phi_{a,b}(x) = (x-a)_- + b$ (see Figure 4). Moreover, both methods only use a simple vote among the base classifiers built (which corresponds to a coefficient of 1 given to each base classifier in the combination). For Blackwell's strategy, $b_t = 0$ and $a_t = 1 - \frac{2}{t-1}\sum_{k=1}^{t-1} \varepsilon_k$, where ε_k is the weighted error of base classifier f_k at step k; in other words, a_t is the average of the mean weighted margins of the base classifiers forming the combination up to the present step.

For the SLVM, $a_t = E[M_{\overline{\alpha}_{t-1}}]$ is the (empirical) average margin of the current combination, in other words, the sum of the mean (unweighted) margins of the base classifiers built up to the present step. If e_k denotes the (unweighted) empirical error of base classifier f_k, this is also equivalent to $a_t = 1 - \frac{2}{t-1}\sum_{k=1}^{t-1} e_k$; notice that, comparing with Blackwell's algorithm, we have just replaced the *weighted* error by the *unweighted* error in a_t. Additionally, $b_t = \beta\sqrt{E[w(X,Y)^2]} - E[w(X,Y)]$ (where w is defined as in Theorem 4) is a quantity depending on the parameter β.

Now, as noticed earlier, the minimax philosophy is more suited to low noise problems, and the mean-variance philosophy to cases were there are noisy regions. This is in accordance with experimental results reporting that AdaBoost's performance can become extremely bad in the presence of noise on the labeling of training examples (Dietterich, 2000), and more generally that AdaBoost can actually exhibit overfitting (Grove and Schuurmans, 1998; Rätsch et al., 2001). Therefore we would like to have an algorithm that is able to behave according to the one or the other philosophy according to the

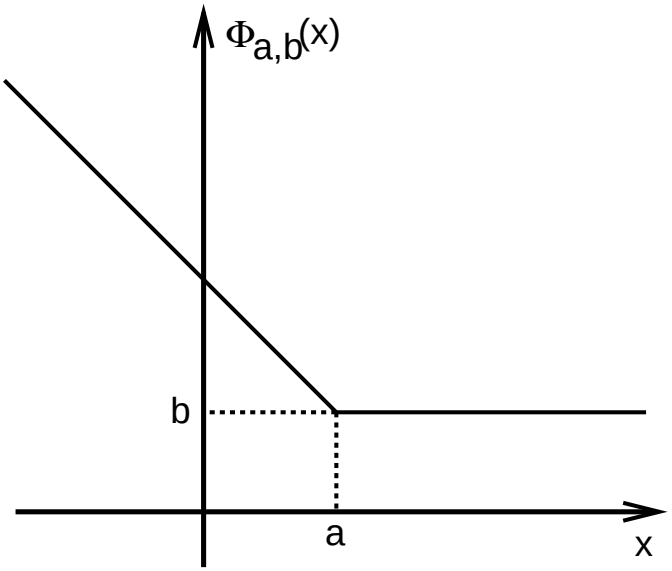

Figure 4: The weighting function $\Phi_{a,b}$.

situation. More precisely, we want to derive a parametrized family of SRSs able to interpolate between the two.

In view of the similar form taken by the reweighting in the two above algorithms, we propose the following simple solution: reweight the examples using a function Φ_{a_t,b_t} of the margins, were (a_t, b_t) are just given by some weighted mean with coefficients $(1-\delta, \delta)$ of the corresponding values for our two initial algorithms. This gives rise to the following choice for some $\delta \in [0,1]$: recalling the notation $w(x,y) = (M_{\overline{\alpha}_{t-1}}(x,y) - E[M_{\overline{\alpha}_{t-1}}])_-$,

$$\begin{cases} a_t &= 1 - \frac{2}{t-1}\sum_{k=1}^{t-1}\left((1-\delta)\varepsilon_k + \delta e_k\right), \\ b_t &= \delta(\beta\sqrt{E[w(X,Y)^2]} - E[w(X,Y)]), \end{cases} \tag{7}$$

The resulting algorithm is summed up in Figure 5 (where it was slightly extended to also accept values of δ greater than 1).

4.2 Experiments

In our tests of the algorithm, we fixed arbitrarily $\beta = 2$ in the interpolated algorithm in order to reduce to a single tuning constant $\delta > 0$, and also to ensure that the weights are always nonnegative. Note that the parameter β

1. For $i = 1, \ldots, N$, initialize weights $\omega_{i,1} = 1/N$, vectors $R_{i,1} = 0, S_{i,1} = 0$. Put $\delta^* = \max(\delta, 1)$.

2. Iteration t: call the weak learner $\mathcal{W}$ with the weights $(\omega_{i,t})$, resulting in classifier f_t. Let e_t denote the empirical *unweighted* error of f_t and ε_t denote its *weighted* error.

3. For $i = 1, \ldots, N$, update vectors R, S the following way:

$$R_{i,t+1} = R_{i,t} + \mathbf{1}\{f_t(X_i) \neq Y_i\} - \varepsilon_t;$$

$$S_{i,t+1} = S_{i,t} + \mathbf{1}\{f_t(X_i) \neq Y_i\} - e_t.$$

4. Put $V_{t+1} = \left(\beta\sqrt{\frac{1}{N}\sum_{i=1}^{N}(S_{i,t+1})_+^2} - \frac{1}{N}\sum_{i=1}^{N}(S_{i,t+1})_+\right)$. Update weights the following way:

$$\text{for } i = 1, \ldots, N, \qquad \omega_{i,t+1} = ((1-\delta^*)R_{i,t+1} + \delta^* S_{i,t+1})_+ + \delta V_{t+1};$$

then renormalize so that the weights sum to 1.

5. If $t < T_{max}$, proceed to next iteration and point (2). If iteration T_{max} is reached, take for the aggregated classifier the simple uniform average (vote) of the f_t's, $t = 1, \ldots, T$.

Figure 5: The interpolated algorithm, depending on constants $\beta > 0$ and $\delta > 0$.

could also be chosen by cross-validation in a reasonable range; however in our experience it resulted in little difference.

We performed two series of experiments: the first with classification trees of limited depth, and the second with RBF networks. The first series aims at investigating the qualitative properties of the algorithm and to compare it to AdaBoost, in particular in the presence of (labeling) noise. For this first series, we chose classification trees and stumps because they have been used as examples in other works on ensemble methods, and because they are fast to compute, allowing us to perform large sets of experiments. On the other hand, since classification trees are not excellent classifiers taken individually, the goal is not to achieve record performance.

In particular, it appears that the Random Forest (RF) algorithm has often better performance than ensemble of trees obtained with our algorithm, however one should keep in mind that RF is completely dedicated to classification trees: contrarily to AdaBoost and other reweighting schemes, it is not divided clearly into a weak learner and an ensemble scheme. In particular, it is difficult to state exactly what the "weak learner" would be for RF, since the randomization is part of the tree building procedure itself (a random subset of features is selected at each node, and the "ensemble" part merely consists in repeating the base procedure with a bootstrap). This makes it difficult to make a fair comparison with other ensemble schemes that use a "black box" weak learner. In any case, the fact that AdaBoost and the interpolated algorithm presented here can be applied to any weak learner proves a decisive advantage: in the second series of experiments, using small-size RBF networks as base classifiers (which are generally better classifiers than single classification trees), the test error rates of the reweighting schemes is very clearly better than RF for an important majority of the datasets.

4.2.1 Experiments with classification trees and stumps

For this first set of experiments we tested the algorithm for 7 benchmark datasets, 6 of which have been used by Rätsch et al. (2001)[3], the last one ('Breast-c') coming from the UCI repository[4].

Our main point in this series of experiments is to compare the behavior of the interpolated algorithm with plain AdaBoost for an arbitrary weak learner, so that we preferred to use a fast, but not very accurate, weak learner.

[3]made available at `http://ida.first.gmd.de/~raetsch/data/benchmarks.htm`

[4]available at `http://www.ics.uci.edu/~mlearn/MLRepository.html`

We tried two procedures: decision trees of depth 1 (stumps) and 3, split using the Gini criterion, with no pruning and a coarse stopping rule. For each of the datasets, we tried the learning algorithms with different labeling noise levels of 0%,5%,10% and 20% respectively; these noise levels correspond to the proportion of training examples whose labels are flipped before learning (the labels of the test samples used to estimate the error remain unchanged). For the interpolated algorithm, the constant δ was determined at each round by a 5-fold cross-validation on the training set, choosing among an arbitrary set of 9 values for δ: $\{0, .05, .1, .15, .2, .5, .75, 1, 1.5\}$ (this probably could be improved). For each of these situations, the error rates were estimated with 100 different training and test sets (the same training and test sets are used with the different algorithms). Finally, for each of the SRSs used, we performed $T_{max} = 500$ iterations at each round.

First, we show in Figure 6 different empirical margins "profiles" (cumulative distribution functions) of the training set, obtained with different values of the parameter on the 'Waveform' dataset. It is interesting to note that these profiles follow basically what we should have been expecting from the construction of the algorithm, namely, that for low δ, the distribution has almost a steep jump near what should be the minimax margin; for higher values of δ, the distribution of the margins is more spread out but also has a higher mean. On the figure it is noticeable that the AdaBoost algorithm does a better job than Blackwell's strategy (corresponding to $\delta = 0$) at pulling the minimum training margin higher; this is probably because, since the AdaBoost algorithm is based on an exponential cost function, it converges very quickly (this nice property of AdaBoost has been pointed out several times in the literature). By comparison, Blackwell's strategy only uses a linear function for its weighting and therefore needs more iterations to converge, and the 500 iterations performed here are probably not enough to reach the asymptotic regime (although the classification rates are very close). On the other hand, Fig. 6 shows that the proposed algorithm achieves precisely its initial purpose, which was to be able to sample different candidate ensemble classifiers that achieve qualitatively different tradeoffs concerning the shape (profile) of the margin c.d.f., this tradeoff being basically between the proportion of examples having a high margin, and the average margin of all the examples. The test set margin distribution also reported in Fig. 6 show that the qualitative differences appearing on the training set produce test profiles following the same qualitative patterns, so that varying the parameter indeed makes a difference for test sample classification. (On the example showed on the figure, it is a case where AdaBoost actually offers the best solution, as is seen at the test c.d.f. at the point 0, which represents the test error. This

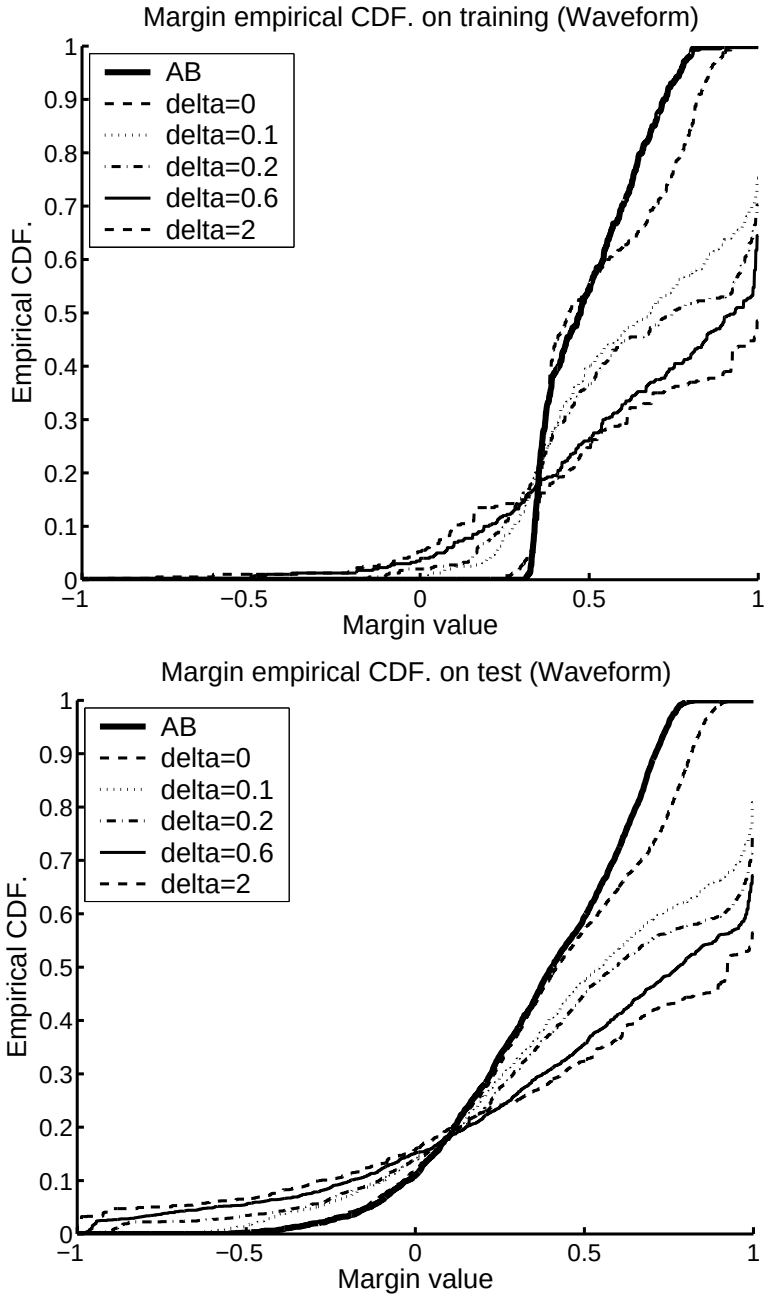

Figure 6: Margin "profiles" (top: on training set; bottom: on test set) obtained on the 'waveform' dataset, with depth 3 decision trees, with AdaBoost and the interpolated algorithm for different values of δ.

matches the results shown in Table 2.)

The classification errors obtained on the different datasets are given on Tables 1 and 2. The interpolated algorithm does noticeably better than AdaBoost in general. It is interesting to note that the classification rates are generally worse (for both algorithms) using depth-3 trees than using stumps (except for the 'Waveform' and 'Banana' datasets).

This must indicate that the depth-3 trees must be overfitting in most of the cases, which has important consequences in terms of the performance of the aggregated classifier. The interpolated algorithm outperforms AdaBoost the most clearly for depth-3 trees on the one hand, and for the higher labeling noise levels on the other, which indicates that the interpolated algorithm is much more resistant to noise and overfitting. The 'German' and 'Diabetes' datasets are particularly instructive: when we compare the classification errors obtained with stumps and depth-3 trees, we see clearly that AdaBoost's performances are severely degraded, while the interpolated algorithm only suffers a small increase in classification errors.

The AdaBoost algorithm outperforms the interpolated algorithm with stumps in some cases on the datasets 'Waveform', 'German' and 'Fsolar', but when we look at the numbers, the two algorithm actually appear mostly on par (only in three cases is the difference significant in the sense of a 95% T-Test). One can sum up the results saying that over all the cases considered there are 32 wins of the interpolated algorithm vs. AdaBoost, 3 losses and 21 ties.

As far as classification trees are concerned, it should be noted that for a majority of these datasets, the Random Forest algorithm actually outperforms the interpolated algorithm (compare to Table 3 in next section). However, as pointed out earlier it is not obvious how to identify a "weak learner" in the RF algorithm, that could be used by the other ensemble algorithms for fair comparison. Moreover, the reweighting schemes can be applied to other weak learners, yielding in most cases better final classification rates than RF, as shown in the next section.

4.2.2 Experiments with RBF networks

For this second set of experiments we used RBF networks as weak learners. In contrast to decision trees, RBF networks are quite accurate weak learners; they are also slower to train, so that in that case we made a more limited set of experiments, only considering data without labelling noise. The goal of this series of experiments is also to provide a performance comparison with the algorithm AdaBoost-Reg (Rätsch et al., 2001), which is an alter-

Lab. Noise %	0%		5%		10%		20%	
Method	AB	Interp.	AB	Interp.	AB	Interp.	AB	Interp.
Breast-c.	3.9 ± 2.1	$\mathbf{3.7 \pm 1.8}$ =	5.1 ± 2.4	$\mathbf{4.1 \pm 2.2}$ +	5.5 ± 2.4	$\mathbf{4.4 \pm 2.0}$ +	7.3 ± 3.2	$\mathbf{5.1 \pm 2.5}$ +
Banana	27.8 ± 1.6	$\mathbf{27.5 \pm 1.7}$ =	28.2 ± 1.8	$\mathbf{27.9 \pm 1.8}$ =	29.0 ± 2.2	$\mathbf{28.5 \pm 2.1}$ =	30.4 ± 2.8	$\mathbf{29.8 \pm 2.8}$ =
German	$\mathbf{24.0 \pm 2.3}$	24.3 ± 2.2 =	$\mathbf{25.0 \pm 2.3}$	$\mathbf{25.0 \pm 2.2}$ =	$\mathbf{25.9 \pm 2.4}$	$\mathbf{25.9 \pm 2.5}$ =	27.9 ± 2.7	$\mathbf{27.6 \pm 2.8}$ =
F.Solar	$\mathbf{33.0 \pm 2.0}$	33.1 ± 2.0 =	$\mathbf{33.3 \pm 1.6}$	33.5 ± 2.0 =	$\mathbf{33.6 \pm 1.9}$	33.9 ± 2.3 =	$\mathbf{34.0 \pm 2.4}$	34.9 ± 2.6 -
Heart	21.7 ± 4.0	$\mathbf{18.6 \pm 3.9}$ +	23.8 ± 3.8	$\mathbf{19.2 \pm 4.3}$ +	26.1 ± 4.8	$\mathbf{21.1 \pm 4.6}$ +	29.9 ± 5.2	$\mathbf{25.4 \pm 5.0}$ +
Diabetes	$\mathbf{24.7 \pm 1.7}$	$\mathbf{24.7 \pm 1.8}$ =	25.8 ± 1.9	$\mathbf{25.0 \pm 1.9}$ +	26.1 ± 2.1	$\mathbf{24.9 \pm 1.8}$ +	28.8 ± 2.6	$\mathbf{26.6 \pm 2.9}$ +
Waveform	12.5 ± 0.6	$\mathbf{12.4 \pm 0.7}$ =	15.4 ± 1.1	$\mathbf{14.0 \pm 1.2}$ +	17.7 ± 1.3	$\mathbf{14.9 \pm 1.5}$ +	22.5 ± 2.0	$\mathbf{17.6 \pm 2.0}$ +

Table 1: Results with stumps as base classifiers (100 training and test sets, 500 iterations of the ensemble algorithms). The sign after the interpolated algorithm results indicate the result of a 95% 2-sided T-test of equality with the AdaBoost results: (+) or (-) indicates that the equality is rejected, (=) indicates that it is not.

Lab. Noise %	0%		5%		10%		20%	
Method	AB	Interp.	AB	Interp.	AB	Interp.	AB	Interp.
Breast-c.	**3.2 ± 1.9**	3.7 ± 2.0 =	5.1 ± 2.4	**4.4 ± 2.6** +	6.8 ± 2.9	**4.4 ± 2.3** +	9.9 ± 4.2	**5.5 ± 2.9** +
Banana	**13.9 ± 0.7**	**13.9 ± 1.2** =	16.6 ± 1.4	**15.8 ± 1.5** +	19.6 ± 1.6	**17.9 ± 1.8** +	25.8 ± 2.1	**23.1 ± 2.2** +
German	25.1 ± 2.3	**24.3 ± 2.1** +	27.6 ± 2.3	**25.6 ± 2.4** +	29.2 ± 2.7	**26.0 ± 2.4** +	33.0 ± 2.8	**27.9 ± 2.7** +
F.Solar	34.6 ± 1.8	**33.8 ± 1.9** +	34.6 ± 1.7	**34.1 ± 2.1** =	35.2 ± 2.0	**34.8 ± 2.2** =	**35.6 ± 2.2**	35.8 ± 2.3 =
Heart	22.2 ± 4.2	**21.3 ± 4.4** =	24.3 ± 4.4	**23.0 ± 4.8** +	26.2 ± 4.3	**24.1 ± 4.7** +	31.6 ± 5.3	**28.0 ± 5.7** +
Diabetes	27.3 ± 2.0	**25.0 ± 1.9** +	28.6 ± 2.2	**25.6 ± 1.7** +	30.0 ± 2.3	**25.5 ± 2.3** +	34.0 ± 2.9	**27.9 ± 3.5** +
Waveform	**11.7 ± 0.6**	12.3 ± 1.0 -	**12.9 ± 0.8**	13.5 ± 1.0 -	**14.7 ± 1.0**	**14.7 ± 1.1** =	20.2 ± 2.0	**18.4 ± 2.1** +

Table 2: Results with (non-pruned) depth 3 tree classifiers, same conditions as in Table 1.

Dataset	Random Forest	AB. RBF-net	AB_{Reg} RBF-net	Interp. RBF-net	vs. AB_R	vs. RF
Banana	12.5 ± 0.8	12.0 ± 0.6	11.0 ± 0.6	$\mathbf{10.7 \pm 0.5}$	+	+
German	$\mathbf{23.0 \pm 2.1}$	26.6 ± 2.5	24.4 ± 2.3	23.9 ± 2.3	=	-
F.Solar	34.2 ± 1.7	35.5 ± 1.8	$\mathbf{33.1 \pm 1.9}$	34.7 ± 1.7	-	-
Heart	18.8 ± 4.0	20.5 ± 3.0	$\mathbf{16.9 \pm 3.9}$	17.6 ± 2.9	=	+
Diabetes	24.7 ± 1.7	26.9 ± 2.2	23.9 ± 1.9	$\mathbf{23.7 \pm 2.0}$	=	+
Waveform	11.1 ± 0.7	11.0 ± 0.6	$\mathbf{9.8 \pm 0.4}$	9.9 ± 0.4	=	+
Titanic	23.0 ± 2.2	22.6 ± 1.2	$\mathbf{22.4 \pm 1.1}$	23.6 ± 2.1	-	-
Breast-c. (2)	$\mathbf{26.6 \pm 4.4}$	30.6 ± 4.8	$\mathbf{26.6} \pm 4.7$	27.3 ± 4.6	=	=
Ringnorm	3.6 ± 0.3	5.4 ± 2.3	$\mathbf{1.6 \pm 0.1}$	1.8 ± 0.2	-	+
Twonorm	3.5 ± 0.5	4.7 ± 1.6	3.5 ± 0.4	$\mathbf{2.9 \pm 0.3}$	+	+

Table 3: Performance comparison chart between Random Forest, AdaBoost, AdaBoost-Reg and the interpolated algorithm (applied to RBF-nets). The two last columns show the results of 95% confidence two-sided T-test of the interpolated algorithm vs. Adaboost-Reg and Random Forest, respectively.

native regularized version of AdaBoost based on different principles. For a fair comparison, we followed the protocol used in the latter reference; more precisely, we used the data and the RBF parameters provided on G.Rätsch's repository; the interpolation parameter was estimated by cross-validation for the first five realizations of each dataset, and the median of these five values was used for the other realizations (which is exactly the protocol followed by Rätsch et al. (2001): this is rendered necessary to limit the computation time needed). We also used a bigger number of benchmark datasets as compared to the first series of experiments (note that the Breast-Cancer dataset here is not the same as in the previous series).

The results for this experiment are reported in Table 3. It is first interesting to note that the best ensemble method for RBF networks is *never* AdaBoost. This shows clearly that AdaBoost largely overfits when the base classifiers are too complex. Individually, the interpolated algorithm applied to RBF networks is almost always better than AdaBoost (only one loss), outperforms Random Forests in a clear majority of cases (6 victories, 3 losses, 1 tie) and is on par with AdaBoost-Reg (5 ties, 3 losses, 2 victories). (An algorithm wins against another whenever the corresponding 2-sided T-test for equality of the mean error rates is rejected). The advantage of the interpolated algorithm with respect to the latter is that it is computationally simpler (in particular, AdaBoost-Reg involves a line search at each step to determine the coefficient of the classifier). The raw results suggest that AdaBoost-Reg may have a slight edge over the interpolated algorithm, but a down-to-earth view of the results taking into account the confidence intervals on test classification leads to the conclusion that the algorithms have

essentially equivalent performance.

5 Conclusion

Numerous theoretical works in the past few years providing bounds on classification error rates of ensemble methods have supported the intuition that one should generally try to obtain high margins on the training set. Once this principle is posed, there still are some partially heuristic choices to be made regarding the tradeoffs to be made and the desired shape of the margin distribution.

We have pointed out two different philosophies that can be considered as different strategies to achieve this goal, and that we showed or recalled to be linked with standard statistical learning procedures used in more classical frameworks (Support Vector Machines and Fisher's linear discriminant), thus trying to provide a unifying point of view over different types of ensemble methods.

Based on this analysis, and observing the remarkably similar form of two specific algorithms designed to achieve the aims of each paradigm, we built a very simple family of algorithms corresponding to interpolated parameter values between the two initial methods. In our opinion, the main virtue of this algorithm is to propose a *simple* method able to *sample* a variety of good candidate empirical margin profiles. This different type of profiles correspond basically to different tradeoffs between the proportion of examples having a high enough margin, and the average margin of the examples. Figure 6 shows clearly on an example how this goal is attained. It is then quite simple to pick among the sampled profiles by cross-validation to choose the tradeoff best suited to the data. This proved, on benchmark datasets, to be an efficient method to improve over the performances of AdaBoost, more noticeably in noisy situations or when the base classifiers tend to be too complex. This algorithm outperforms Random Forests when used with good base classifiers (small-size RBF networks) and exhibits the same level of performance as the Regularized AdaBoost method, while being noticeably simpler.

Acknowledgements

This work finds some of its roots in a joined research project with Pr. Y. Amit whom the author would like to thank. The author would also like to thank K-R. Müller, G. Rätsch and S. Mika for stimulating discussions about this work.

References

Amit, Y. and Blanchard, G. (2001). Multiple randomized classifiers: MRCL, *Technical report*, University of Chicago.
*http://galton.uchicago.edu/~amit/Papers/mrcl.ps.gz

Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees, *Neural Computation* **9**: 1545–1588.

Blackwell, D. (1956). An analog of the minmax theorem for vector payoffs, *Pacific Journal of Mathermatics* **6**: 1–8.

Blanchard, G. (2001). *Mixture and aggregation of estimators for pattern recognition. Application to decision trees.*, PhD thesis, Université Paris-Nord. (In English, with an introductory part in French).
*http://www.math.u-psud.fr/~blanchard/publi/these.ps.gz

Blanchard, G. (2003). Generalization error bounds for aggregate classifiers, *in* D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick and B. Yu (eds), *Nonlinear Estimation and Classification*, Vol. 171 of *Lecture Notes in Statistics*, Springer.

Breiman, L. (1998a). Arcing classifiers, *The annals of statistics* **26**(3): 801–849.

Breiman, L. (1998b). Prediction games and arcing algorithms, *Technical report*, Statistics department, University of California at Berkeley.
*ftp://ftp.stat.berkeley.edu/pub/users/breiman/games.ps.Z

Breiman, L. (2001). Random forests, *Machine Learning* **45**: 5–32.

Devroye, L., Györfi, L. and Lugosi, G. (1996). *A probabilistic theory of pattern recognition*, Vol. 31 of *Applications of Mathematics*, Springer.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization, *Machine Learning* **40**.

Frean, M. and Downs, T. (1998). A simple cost function for boosting, *Technical report*, Dep. of Computer Science and Electrical Engineering, University of Queensland.
*http://www.boosting.org/papers/FreDow98.ps.gz

Freund, Y. and Schapire, R. (1996a). Experiments on a new boosting algorithm, *Machine Learning: proceedings of the 13th international conference*, pp. 148–156.

Freund, Y. and Schapire, R. E. (1996b). Game theory, on-line prediciton and Boosting, *Proceedings of the 9th annual conference on computational learning theory.*

Friedman, J., Hastie, T. and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting, *The Annals of Statistics* **28**: 337–374.

Grove, A. and Schuurmans, D. (1998). Boosting in the limit: Maximizing the margin of learned ensembles, *Proceedings of the Fifteenth National Conference on Artifical Intelligence.*
*http://www.boosting.org/papers/GroSch98.ps.gz

Hart, S. and Mas-Colell, A. (2001). A general class of adaptive strategies, *Journal of Economic Theory* **98**: 26–54.

Koltchinkskii, V. and Panchenko, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers, *Annals of Statistics* **30**(1).

Meir, R. and Rätsch, G. (2003). An introduction to boosting and leveraging, *in* S. Mendelson and A. Smola (eds), *Advanced Lectures on Machine Learning*, LNCS, Springer, pp. 119–184. In press. Copyright by Springer Verlag.
*http://www.boosting.org/papers/MeiRae03.ps.gz

Mika, S. (2002). *Kernel Fisher Discriminants*, PhD thesis, University of Technology, Berlin.

Onoda, T., Rätsch, G. and Müller, K.-R. (1998). An asymptotic analysis of AdaBoost in the binary classification case, *in* L. Niklasson, M. Bodén and T. Ziemke (eds), *Proc. of the Int. Conf. on Artificial Neural Networks (ICANN'98)*, pp. 195–200.
*http://www.boosting.org/papers/ICANN98.ps.gz

Rätsch, G., Onoda, T. and Müller, K.-R. (2001). Soft margins for adaboost, *Machine Learning* **3**(42): 287–320.

Rätsch, G. and Warmuth, M. (2001). Marginal boosting, *Technical report*, Royal Holloway College, London.

Rätsch, G., Mika, S., Schölkopf, B. and Müller, K.-R. (2002). Constructing boosting algorithms from SVMs: an application to one-class classification, *IEEE P.A.M.I.* **24**(9).

Schapire, R. E., Freund, Y., Bartlett, P. and Lee, W. S. (1998). Boosting the margins: a new explanation for the effectiveness of voting methods, *Annals of Statistics* **26**(5): 1651–1686.

Smola, A. J., Bartlett, P. L., Schölkopf, B. and Schuurmans, D. (eds) (2000). *Advances in Large Margin Classifiers*, MIT Press.

Viola, P. and Jones, M. J. (2001). Robust real-time object detection, *Technical Report CRL2001/01*, COMPAQ Cambridge research laboratory.

Appendix

Proof of theorem 1. Point (ii) is a direct consequence of (i): by the minimax theorem, the minimax margin γ is such that for any collection of weights (ω_i), there exists a classifier $f \in \mathcal{H}$ having average weighted margin higher than γ or, in other words, weighted error less than $(1-\gamma)/2$. Since $\mathcal{W}$ finds a classifier with minimum weighted error, we must have $\varepsilon_t \leq (1-\gamma)/2$ for all t.

Now, we have for all $i = 1, \ldots, N$

$$R_{i,t} = \sum_{k=1}^{t} (\mathbf{1}\{f_k(X_i)Y_i \leq 0\} - \varepsilon_k) = \sum_{k=1}^{t} \left(\frac{1}{2}(1 - f_k(X_i)Y_i) - \varepsilon_k \right),$$

and (i) hence implies that for any for any $\delta > 0$, for t big enough we have for all i

$$\frac{1}{t}\sum_{k=1}^{t} f_k(X_i)Y_i \geq 1 - \frac{2}{t}\sum_{k=1}^{t} \varepsilon_k - \delta \geq \gamma - \delta,$$

which proves (ii). □

Proof of the convexity of cost function $C_\beta(G_{\overline{\alpha}})$. We want to prove that $C_\beta(G_{\overline{\alpha}})$ given by (5) is convex on the simplex $\mathcal{S} = \{|\overline{\alpha}| = 1\}$. Since the margin function $M_{\overline{\alpha}}$ is a linear function of $|\overline{\alpha}|$, this thus amounts to proving that the

function $\sqrt{Var_-[M_{\overline{\alpha}}]}$ is convex. Let $\overline{\alpha}_1, \overline{\alpha}_2 \in \mathcal{S}$, $\eta \in [0,1]$ and consider

$$\begin{aligned}
A &= Var_- \left[M_{\eta\overline{\alpha}_1+(1-\eta)\overline{\alpha}_2}\right] \\
&= E\left[\left(\eta\left(M_{\overline{\alpha}_1} - E\left[M_{\overline{\alpha}_1}\right]\right) + (1-\eta)\left(M_{\overline{\alpha}_2} - E\left[M_{\overline{\alpha}_2}\right]\right)\right)_-^2\right] \\
&\leq E\left[\left(\eta\left(M_{\overline{\alpha}_1} - E\left[M_{\overline{\alpha}_1}\right]\right)_- + (1-\eta)\left(M_{\overline{\alpha}_2} - E\left[M_{\overline{\alpha}_2}\right]\right)_-\right)^2\right] \\
&= \eta^2 E\left[\left(M_{\overline{\alpha}_1} - E\left[M_{\overline{\alpha}_1}\right]\right)_-^2\right] + (1-\eta)^2 E\left[\left(M_{\overline{\alpha}_2} - E\left[M_{\overline{\alpha}_2}\right]\right)_-^2\right] \\
&\qquad + 2\eta(1-\eta)E\left[\left(M_{\overline{\alpha}_1} - E\left[M_{\overline{\alpha}_1}\right]\right)_-\left(M_{\overline{\alpha}_2} - E\left[M_{\overline{\alpha}_2}\right]\right)_-\right],
\end{aligned}$$

where at the third line we have used the convexity of the negative part $(\cdot)_-$; and

$$\begin{aligned}
B &= \left(\eta Var_-\left[M_{\overline{\alpha}_1}\right]^{1/2} + (1-\eta)Var_-\left[M_{\overline{\alpha}_2}\right]^{1/2}\right)^2 \\
&= \eta^2 E\left[\left(M_{\overline{\alpha}_1} - E\left[M_{\overline{\alpha}_1}\right]\right)_-^2\right] + (1-\eta)^2 E\left[\left(M_{\overline{\alpha}_2} - E\left[M_{\overline{\alpha}_2}\right]\right)_-^2\right] \\
&\qquad + 2\eta(1-\eta)\left(E\left[\left(M_{\overline{\alpha}_1} - E\left[M_{\overline{\alpha}_1}\right]\right)_-^2\right]\right)^{1/2}\left(E\left[\left(M_{\overline{\alpha}_2} - E\left[M_{\overline{\alpha}_2}\right]\right)_-^2\right]\right)^{1/2};
\end{aligned}$$

we have $A \leq B$ by the Cauchy-Schwartz inequality, hence the result. □

Proof of theorem 4. Let us start with recalling the following simple fact: if $h(x)$ is a differentiable function on $\mathbb{R}$, then it is easy to check that

$$\left.\frac{d(h(x))_-^2}{dx}\right|_{x=x_0} = -2(h(x_0))_- h'(x_0).$$

Now, let us assume without loss of generality that $|\overline{\alpha}| = 1$ and put $M_f(x,y) = f(x)y$ (so that, since $f(x) \in \{-1,1\}$, we have $M_f(x,y) = 1 - 2\mathcal{E}_f(x,y)$), then

$$Var_-[(1-t)M_{\overline{\alpha}} + tM_f] = E[(M_{\overline{\alpha}} - E[M_{\overline{\alpha}}] + t(M_f - M_{\overline{\alpha}} - E[M_f - M_{\overline{\alpha}}]))_-^2],$$

so that using the first remark,

$$\begin{aligned}
\left.\frac{\partial}{\partial t}\right|_{t=0} Var_-[(1-t)M_{\overline{\alpha}} + tM_f] &= -2E[(M_{\overline{\alpha}} - E[M_{\overline{\alpha}}])_-(M_f - M_{\overline{\alpha}} - E[M_f - M_{\overline{\alpha}}])] \\
&= 4E[w(X,Y)\mathcal{E}_f(X,Y)] - 4E[w(X,Y)]E[\mathcal{E}_f] + C_{\overline{\alpha}},
\end{aligned}$$

where $C_{\overline{\alpha}}$ is independent of f; similarly

$$\left.\frac{\partial}{\partial t}\right|_{t=0} E[(1-t)M_{\overline{\alpha}} + tM_f] = E[M_f - M_{\overline{\alpha}}] = -2E[\mathcal{E}_f] + C'_{\overline{\alpha}},$$

so that finally

$$\left.\frac{\partial}{\partial t}\right|_{t=0} C_\beta(G_t) = 2\left(\frac{E[w(X,Y)\mathcal{E}_f(X,Y)] - E[w(X,Y)]E[\mathcal{E}_f]}{\sqrt{Var_-[M_{\overline{\alpha}}]}} + \beta E[\mathcal{E}_f]\right) + C''_{\overline{\alpha}};$$

the result follows by translation and multiplication by constants independent of f. □