Machine Learning, 42, 287–320, 2001

Soft Margins for AdaBoost

G. RÄTSCH raetsch@first.gmd.de
GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany*

T. ONODA onoda@criepi.denken.or.jp
CRIEPI,† Komae-shi, 2-11-1 Iwado Kita, Tokyo, Japan

K.-R. MÜLLER klaus@first.gmd.de
GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany; University of Potsdam, Neues Palais 10, 14469 Potsdam, Germany*

Editor: Robert Schapire

Abstract. Recently ensemble methods like ADABOOST have been applied successfully in many problems, while seemingly defying the problems of overfitting.

ADABOOST rarely overfits in the low noise regime, however, we show that it clearly does so for higher noise levels. Central to the understanding of this fact is the margin distribution. ADABOOST can be viewed as a constraint gradient descent in an error function with respect to the margin. We find that ADABOOST asymptotically achieves a *hard margin* distribution, i.e. the algorithm concentrates its resources on a few hard-to-learn patterns that are interestingly very similar to Support Vectors. A hard margin is clearly a sub-optimal strategy in the noisy case, and regularization, in our case a "mistrust" in the data, must be introduced in the algorithm to alleviate the distortions that single difficult patterns (e.g. outliers) can cause to the margin distribution. We propose several regularization methods and generalizations of the original ADABOOST algorithm to achieve a *soft margin*. In particular we suggest (1) regularized $\text{ADABOOST}_{\text{REG}}$ where the gradient decent is done directly with respect to the soft margin and (2) regularized linear and quadratic programming (LP/QP-) ADABOOST, where the soft margin is attained by introducing slack variables.

Extensive simulations demonstrate that the proposed regularized ADABOOST-type algorithms are useful and yield competitive results for noisy data.

Keywords: ADABOOST, arcing, large margin, soft margin, classification, support vectors

1. Introduction

Boosting and other ensemble[1] learning methods have been recently used with great success in applications like OCR (Schwenk & Bengio, 1997; LeCun et al., 1995). But so far the reduction of the generalization error by Boosting algorithms has not been fully understood.

For **low noise** cases Boosting algorithms are performing well for good reasons (Schapire et al., 1997; Breiman, 1998). However, recent studies with **highly noisy** patterns (Quinlan, 1996; Grove & Schuurmans, 1998; Rätsch et al., 1998) showed that it is clearly a myth that Boosting methods do not overfit.

*www.first.gmd.de
†criepi.denken.or.jp
*www.uni-potsdam.de

In this work, we try to gain insight into these seemingly contradictory results for the low and high noise regime and we propose improvements of ADABOOST that help to achieve noise robustness.

Due to their similarity, we will refer in the following to ADABOOST (Freund & Schapire, 1994) and *unnormalized Arcing* (Breiman, 1997b) (with exponential function) as *ADABOOST-type algorithms* (ATA). In Section 2 we give an asymptotical analysis of ATAs. We find that the error function of ATAs can be expressed in terms of the margin and that in every iteration ADABOOST tries to minimize this error by a stepwise maximization of the margin (see also Breiman, 1997a; Frean & Downs, 1998; Friedman, Hastie, & Tibshirani, 1998; Onoda, Rätsch, & Müller, 1998; Rätsch, 1998). As a result of the asymptotical analysis of this error function, we introduce the *hard margin* concept and show connections to Support Vector (SV) learning (Boser, Guyon, & Vapnik, 1992) and to linear programming (LP). Bounds on the size of the margin are also given.

In Section 3 we explain why an ATA that enforces a hard margin in training will overfit for noisy data or overlapping class distributions. So far, we only know what a margin distribution to achieve optimal classification in the no-noise case should look like: a large hard margin is clearly a good choice (Vapnik, 1995). However, for noisy data there is always the tradeoff between "believing" in the data or "mistrusting" it, as the very data point could be mislabeled or an outlier. So we propose to relax the hard margin and to regularize by allowing for misclassifications (*soft margin*). In Section 4 we introduce such a regularization strategy to ADABOOST and subsequently extend the LP-ADABOOST algorithm of Grove and Schuurmans (1998) by slack variables to achieve soft margins. Furthermore, we propose a quadratic programming ADABOOST algorithm (QP-ADABOOST) and show its connections to SUPPORT VECTOR MACHINES (SVMs).

Finally, in Section 5 numerical experiments on several artificial and real-world data sets show the validity and competitiveness of our regularized Boosting algorithms. The paper concludes with a brief discussion.

2. Analysis of ADABOOST's learning process

2.1. Algorithm

Let $\{h_t(\mathbf{x}) : t = 1, \ldots, T\}$ be an ensemble of T hypotheses defined on an input vector $\mathbf{x} \in X$ and let $\mathbf{c} = [c_1 \cdots c_T]$ be their weights satisfying $c_t \geq 0$ and $\sum_{t=1}^{T} c_t = 1$. We will consider only the binary classification case in this work, i.e. $h_t(\mathbf{x}) = \pm 1$; most results, however, can be transfered easily to classification with more than two classes (e.g. Schapire, 1999; Schapire & Singer, 1998; Breiman, 1997b).

The ensemble generates the label $\tilde{f}(\mathbf{x}) \equiv \tilde{f}_T(\mathbf{x})$ which is the weighted majority of the votes, where

$$f_T(\mathbf{x}) := \sum_{t=1}^{T} c_t h_t(\mathbf{x}) \quad \text{and} \quad \tilde{f}_T(\mathbf{x}) := \operatorname{sign}(f_T(\mathbf{x})).$$

In order to train the ensemble, i.e. to find T appropriate hypotheses $\{h_t(\mathbf{x})\}$ and the

weights $\mathbf{c}$ for the convex combination, several algorithms have been proposed: popular ones are WINDOWING (Quinlan, 1992), BAGGING (Breiman, 1996), ADABOOST (Freund & Schapire, 1994), ARC-X4 (Breiman, 1998) and ARC-GV (Breiman, 1997b). In the sequel analysis, we will focus on ADABOOST-type algorithms and give their pseudo-code in figure 1 (further details can be found in e.g. Freund & Schapire, 1994; Breiman, 1997b).

In the binary classification case, we define the *margin* for an input-output pair $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ as

$$\rho(\mathbf{z}_i, \mathbf{c}) = y_i f(\mathbf{x}_i) = y_i \sum_{t=1}^{T} c_t h_r(\mathbf{x}_i), \tag{1}$$

Algorithm ATA$(\mathbf{Z}, T, \phi)$

Input: l examples $\mathbf{Z} = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l) \rangle$

Initialize: $w_1(\mathbf{z}_i) = 1/l$ for all $i = 1 \ldots l$

Do for $t = 1, \ldots, T$,

1. Train classifier with respect to the weighted sample set $\{\mathbf{Z}, \mathbf{w}_t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto \{\pm 1\}$
2. Calculate the training error ϵ_t of h_t:

$$\epsilon_t = \sum_{i=1}^{l} w_t(\mathbf{z}_i) \mathrm{I}(h_t(\mathbf{x}_i) \neq y_i) \ , \tag{F1.1}$$

   abort if $\epsilon_t = 0$ or $\epsilon_t \geq \phi - \Delta$, where Δ is a small constant

3. Set

$$b_t = \log \frac{\phi(1 - \epsilon_t)}{\epsilon_t(1 - \phi)}. \tag{F1.2}$$

4. Update weights $\mathbf{w}_t$:

$$w_{t+1}(\mathbf{z}_i) = w_t(\mathbf{z}_i) \exp\left\{-b_t \mathrm{I}(h_t(\mathbf{x}_i) = y_j)\right\} / Z_t \ , \tag{F1.3}$$

   where Z_t is a normalization constant, such that $\sum_{i=1}^{l} w_{t+1}(\mathbf{z}_i) = 1$.

Output: Final hypothesis

$$f(\mathbf{x}) = \sum_{t=1}^{T} c_t h_t(\mathbf{x}), \quad \text{where} \quad c_t := \frac{b_t}{\sum_{t=1}^{T} |b_t|} \tag{F1.4}$$

Figure 1. The ADABOOST-type algorithm (ATA). For $\phi = \frac{1}{2}$, we retrieve the original ADABOOST algorithm (Freund & Schapire, 1994). The ATA is a specialization of unnormalized Arcing (Breiman, 1997a) (with exponential function).

where $i = 1, \ldots, l$ and l denotes the number of training patterns. The margin at $\mathbf{z}$ is positive if the correct class label of the pattern is predicted. As the positivity of the margin value increases, the decision stability becomes larger. Moreover, if $|\mathbf{c}| := \sum_{t=1}^{T} c_t = 1$, then $\rho(\mathbf{z}_i, \mathbf{c}) \in [-1, 1]$. We will sometimes for convenience also use a margin definition with $\mathbf{b}$ (instead of $\mathbf{c}$) which denotes simply an unnormalized version of $\mathbf{c}$, i.e. usually $|\mathbf{b}| \neq 1$ (cf. (F1.2) and (F1.4) in figure 1). Note that the *edge* (cf. Breiman, 1997b) is just an affine transformation of the margin.

The margin $\varrho(\mathbf{c})$ of a classifier (instance) is defined as the smallest margin of a pattern over the training set, i.e.

$$\varrho(\mathbf{c}) = \min_{i=1,\ldots,l} \rho(\mathbf{z}_i, \mathbf{c}).$$

Figure 2 illustrates the functioning of ADABOOST. Patterns that are misclassified get higher weights in the next iteration. The patterns near the decision boundary are usually harder to classify and therefore get high weights after a few iterations.

2.2. *Error function of ADABOOST*

An important question in the analysis of ATAs is: *what kind of error function is optimized*? From the algorithmic formulation (cf. figure 1), it is not straight forward to see what the

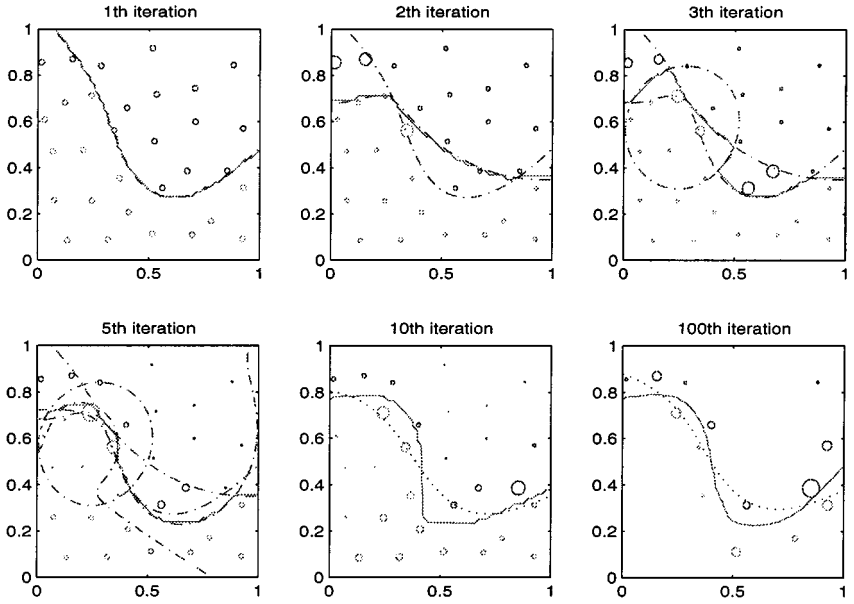

Figure 2. Illustration of ADABOOST on a 2D toy data set: The diameter of the points (the brightness gives the class label) is proportional to the weight that the pattern gets in the first, second, third, 5th, 10th and 100th iteration. The dash-dotted lines show the decision boundaries of the single classifiers (up to the 5th iteration). The solid line shows the decision line of the combined classifier. In the last two plots the decision line of BAGGING is plotted for a comparison (dotted).

aim of this algorithm is. So to gain a better understanding why one should use the weights of the hypotheses c_t and of the patterns $w_t(\mathbf{z}_i)$ in the manner of Eqs. (F1.2) and (F1.3), let us study the following three statements

1. The weights $w_t(\mathbf{z}_i)$ in the t-th iteration are chosen such that the previous hypothesis has exactly a weighted training error ϵ of $1/2$ (Schapire et al., 1997).
2. The weight b_t (and c_t) of a hypothesis is chosen such that it minimizes a functional G first introduced by Breiman (1997b) (see also Rätsch, 1998; Mason, Bartlett, & Baxter, 2000a; Onoda et al., 1998; Friedman et al., 1998; Frean & Downs, 1998). This functional depends on the margins of all patterns and is defined by

$$G(b_t, \mathbf{b}^{t-1}) = \sum_{i=1}^{l} \exp\left\{-\rho(\mathbf{z}_i, \mathbf{b}^t) + |\mathbf{b}^t|\left(\frac{1}{2} - \phi\right)\right\}, \tag{2}$$

   where ϕ is a constant (cf. figure 1). This functional can be minimized analytically (Breiman, 1997b) and one gets the explicit form of Eq. (F1.2) as a solution of $\frac{\partial G(b_t, \mathbf{b}^{t-1})}{\partial b_t} = 0$.
3. To train the t-th hypothesis (step 1 in figure 1) we can either use bootstrap replicates of the training set (sampled according to $\mathbf{w}_t$) or minimize a weighted error function for the base learning algorithm. We observed that the convergence of the ATA is faster if a weighted error function is used.

Taking a closer look at the definition of G, one finds that the computation of the sample distribution $\mathbf{w}_t$ (cf. Eq. (F1.3)) can be derived directly from G. Assuming that G is the error function which is minimized by the ATA, then G essentially defines a loss function over margin distributions, which depends on the value of $|\mathbf{b}|$. The larger the margins $\rho(\mathbf{z}_i)$, the smaller will be the value of G.

So, the gradient $\frac{\partial G}{\partial \rho(\mathbf{z}_i)}$ gives an answer to the question, which pattern should increase its margin most strongly in order to decrease G maximally (gradient descent). This information can then be used to compute a re-weighting of the sample distribution $\mathbf{w}_t$ for training the next hypothesis h_t. If it is important to increase the margin of a pattern $\mathbf{z}_i$, then its weight $w_t(\mathbf{z}_i)$ should be high—otherwise low (because the distribution $\mathbf{w}_t$ sums to one). Interestingly, this is exactly what ATAs are doing and we arrive at the following lemma (Breiman, 1997b; Rätsch, 1998):

Lemma 1. *The computation of the pattern distribution $\mathbf{w}_{t+1}$ in the t-th iteration is equivalent to normalizing the gradient of $G(b_{t+1}, \mathbf{b}^t)$ with respect to $\rho(\mathbf{z}_i, \mathbf{b}^t)$, i.e.*

$$w_{t+1}(\mathbf{z}_i) = \frac{\partial G(b_{t+1}, \mathbf{b}^t)}{\partial \rho(\mathbf{z}_i, \mathbf{b}^t)} \Bigg/ \sum_{j=1}^{l} \frac{\partial G(b_{t+1}, \mathbf{b}^t)}{\partial \rho(\mathbf{z}_j, \mathbf{b}^t)}. \tag{3}$$

The proof can found in Appendix A.

From Lemma 1, the analogy to a gradient descent method is (almost) complete. In a gradient descent method, the first step is to compute the gradient of the error function with

respect to the parameters which are to be optimized: this corresponds to computing the gradient of G with respect to the margins. The second step is to determine the step size in gradient direction (usually done by a line-search): this is analogous to the minimization of G with respect to b_t (cf. point 2).

Therefore, ATAs can be related to a gradient descent method in a hypothesis (or function) space $\mathcal{H}$ which is determined by the structure of the base learning algorithm, i.e. ATAs aim to minimize the functional G by constructing an ensemble of classifiers (Onoda et al., 1998; Rätsch, 1998; Mason et al., 2000b; Friedman et al., 1998; Friedman, 1999). This also explains point 1 in the list above, as in a standard gradient descent method, a new search direction is usually chosen perpendicular to the previous one.

In the ADABOOST-type algorithm, the gradients are found by changing the weights of the training patterns, and there are essentially two ways of incorporating the re-weighting into the boosting procedure. The first is to create bootstrap replicates sampled according to the pattern weightings, which usually induces strong random effects that hide the "true" information contained in the pattern weightings. The second and more direct way is to weight the error function and use weighted minimization (Breiman, 1997b). Clearly, weighted minimization is more efficient in terms of the number of boosting iterations than the bootstrap approach.[2] In fact, it can be shown that employing weighted minimization (Breiman, 1997b) for finding the next hypothesis in each iteration leads to the best (single) hypothesis for minimizing G (Mason et al., 2000a), i.e. adding the hypothesis with smallest weighted training error ϵ_t will lead to the smallest value of G and therefore to a fast convergence. This reasoning explains the third statement.

2.3. *ADABOOST as an annealing process*

From the definition of G and $\rho(\mathbf{z}_i, \mathbf{b}^t)$, Eq. (3) can be rewritten as

$$w_{t+1}(\mathbf{z}_i) = \frac{\exp\left(-\frac{1}{2}\rho(\mathbf{z}_i, \mathbf{c}^t)\right)^{|\mathbf{b}^t|}}{\sum_{j=1}^{l} \exp\left(-\frac{1}{2}\rho(\mathbf{z}_j, \mathbf{c}^t)\right)^{|\mathbf{b}^t|}}, \tag{4}$$

where we emphasize that $|\mathbf{b}^t|$ can be written in the exponent. Inspecting this equation more closely, we see that ATA uses a soft-max function (e.g. Bishop, 1995) with parameter $|\mathbf{b}^t|$ that we would like to interpret as an *annealing parameter* (Onoda et al., 1998; Onoda, Rätsch, & Müller, 2000). In the beginning $|\mathbf{b}^t|$ is small and all patterns have almost the same weights (if $|\mathbf{b}^t| = 0$ then all weights are the same). As $|\mathbf{b}^t|$ increases, the patterns with smallest margin will get higher and higher weights. In the limit of large $|\mathbf{b}^t|$, we arrive at the maximum function: Only the pattern(s) with the smallest margin will be taken into account for learning and get a non-zero weight.

Note that in the limit for $|\mathbf{b}^t| \to \infty$, a simple rescaling of the error function $G(\mathbf{b}^t)$ gives the minimum margin $\varrho(\mathbf{z}_i, \mathbf{c}^t)$, i.e. $\varrho(\mathbf{z}_i, \mathbf{c}^t) = -\lim_{|\mathbf{b}^t|\to\infty} \frac{1}{|\mathbf{b}^t|} \log G(\mathbf{b}^t)$.

The following lemma shows that under usual circumstances, the length of the hypothesis weight vector $|\mathbf{b}^t|$ increases at least linearly with the number of iterations.

Lemma 2. *If, in the learning process of an ATA with* $0 < \phi < 1$, *all weighted training errors* ϵ_t *are bounded by* $\epsilon_t \leq \phi - \Delta$ $(0 < \Delta < \phi)$, *then* $|\mathbf{b}|$ *increases at least linearly with the number of iterations* t.

Proof: With (F1.2), the smallest value for b_t is achieved, if $\epsilon_t = \phi - \Delta$. Then we have $b_t = \log \frac{q}{q-\Delta}$, where $q := \phi(1 - \phi + \Delta)$. We find $q > \Delta > 0$ and because of $\phi(1 - \phi) > \Delta(1 - \phi)$ we get $\phi(1 - \phi + \Delta) > \Delta$ and hence also $q - \Delta > 0$. Therefore, the smallest value of b_t is $\log \frac{q}{q-\Delta}$ is always larger than a constant γ, which only depends on ϕ and Δ. Thus, we have $|\mathbf{b}^t| > t\gamma$. □

If the annealing speed is low, the achieved solution should have larger margins than for a high speed annealing strategy. This holds for similar reasons as for a standard annealing process (Kirkpatrick, 1984): in the error surface, a better local minimum (if exist) can be obtained locally, if the annealing speed is slow enough. From Eq. (F1.2), we observe that if the training error ϵ_t takes a small value, b_t becomes large. So, strong learners can reduce their training errors strongly and will make $|\mathbf{b}|$ large after only a few ATA iterations, i.e. the asymptotics, where the addition of a new ensemble member does not change the result, is reached faster. To reduce the annealing speed either ϕ or the complexity of the base hypotheses has to be decreased (with the constraint $\epsilon_t < \phi - \Delta$; cf. Onoda et al., 1998).

In figure 3 (left), the ADABOOST Error ($\phi = \frac{1}{2}$), the Squared Error $(y - f(x))^2$ and the Kullback-Leibler Error $\ln \rho(\mathbf{z})/\ln 2$ are plotted. Interestingly, the Squared and Kullback-Leibler Error are very similar to the error function of ADABOOST for $|\mathbf{b}| = 3$. As $|\mathbf{b}|$ increases,[3] the ATA error function approximates a $0/\infty$ loss (patterns with margin smaller than $1 - 2\phi$ get loss ∞; all others have loss 0) and bounds the $0/1$ loss at $1 - 2\phi$ from above.

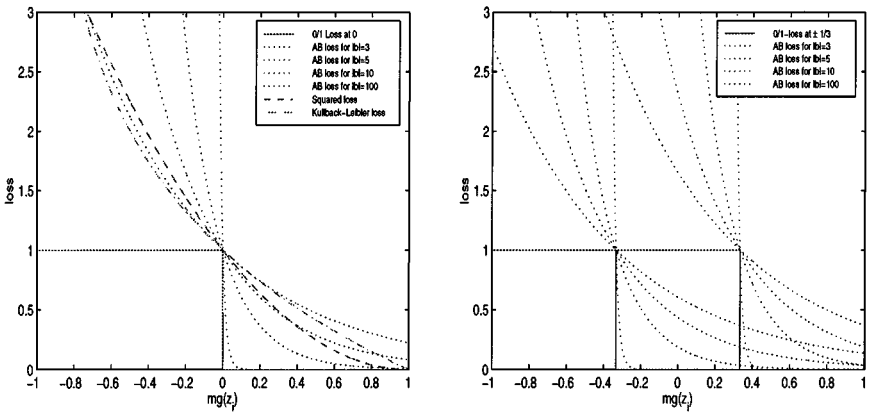

Figure 3. Different loss functions for classification (see text). The abscissa shows the margin $yf(\mathbf{x})$ of a pattern and the y-coordinate shows the monotone loss for that pattern: $0/1$-Loss, Squared Error, Kullback-Leibler Error and ADABOOST Error (cf. Eq. (2)), where $|\mathbf{b}|$ is either 3, 5, 10 or 100 (in reading order). On the left panel is $\phi = 1/2$ and on the right plot ϕ is either $1/3$ or $2/3$. The step position of the $0/\infty$ loss, which is approximated for $|\mathbf{b}| \to \infty$, is determined by ϕ.

Figure 3 (right) shows the different offsets of the step seen in the ATA Error ($|\mathbf{b}| \to \infty$). They result from different values of ϕ (here ϕ is either $1/3$ or $2/3$).

2.4. Asymptotic analysis

2.4.1. How large is the margin? ATA's good generalization performance can be explained in terms of the size of the (hard) margin that can be achieved (Schapire et al., 1997; Breiman, 1997b): for low noise, the hypothesis with the largest margin will have the best generalization performance (Vapnik, 1995; Schapire et al., 1997). Thus, it is interesting to understand what the margin size depends on.

Generalizing Theorem 5 of Freund and Schapire (1994) to the case $\phi \neq \frac{1}{2}$ we get

Theorem 3. *Assume, $\epsilon_1, \ldots, \epsilon_T$ are the weighted classification errors of $h_1, \ldots, h_T$ that are generated by running an ATA and $1 > \phi > \max_{t=1,\ldots,T} \epsilon_t$. Then the following inequality holds for all $\theta \in [-1, 1]$:*

$$\frac{1}{l}\sum_{i=1}^{l} I\left(y_i f(\mathbf{x}_i) \leq \theta\right) \leq \left(\varphi^{\frac{1+\theta}{2}} + \varphi^{-\frac{1-\theta}{2}}\right)^T \prod_{t=1}^{T} \sqrt{\epsilon_t^{1-\theta}(1-\epsilon_t)^{1+\theta}}, \tag{5}$$

where f is the final hypothesis and $\varphi = \frac{\phi}{1-\phi}$, where I is the indicator function.

The proof can be found in Appendix B.

Corollary 4. *An ADABOOST-type algorithm will asymptotically ($t \to \infty$) generate margin distributions with a margin ϱ, which is bounded from below by*

$$\varrho \geq \frac{\ln(\phi\epsilon^{-1}) + \ln((1-\phi)(1-\epsilon)^{-1})}{\ln(\phi\epsilon^{-1}) - \ln((1-\phi)(1-\epsilon)^{-1})}, \tag{6}$$

where $\epsilon = \max_t \epsilon_t$, if $\epsilon \leq (1-\varrho)/2$ is satisfied.

Proof: The maximum of $\epsilon_t^{1-\theta}(1-\epsilon_t)^{1+\theta}$ with respect to ϵ_t is obtained for $\frac{1}{2}(1-\theta)$ and for $0 \leq \epsilon_t \leq \frac{1}{2}(1-\theta)$ it is increasing monotonically in ϵ_t. Therefore, we can replace ϵ_t by ϵ in Eq. (5) for $\theta \leq \varrho$:

$$\mathbf{P}_{(\mathbf{x},y)\sim\mathbf{Z}}[y f(\mathbf{x}) \leq \theta] \leq \left(\left(\varphi^{\frac{1+\theta}{2}} + \varphi^{-\frac{1-\theta}{2}}\right)\epsilon^{\frac{1-\theta}{2}}(1-\epsilon)^{\frac{1+\theta}{2}}\right)^T.$$

If the basis on the right hand side is smaller than 1, then asymptotically we have $\mathbf{P}_{(\mathbf{x},y)\sim\mathbf{Z}}[y f(\mathbf{x}) \leq \theta] = 0$; this means that asymptotically, there is no example that has a smaller margin than θ, for any $\theta < 1$. The supremum over all θ such that the basis is less than 1, $\theta_{\max}$ is described by

$$\left(\varphi^{\frac{1+\theta_{\max}}{2}} + \varphi^{-\frac{1-\theta_{\max}}{2}}\right)\epsilon^{\frac{1-\theta_{\max}}{2}}(1-\epsilon)^{\frac{1+\theta_{\max}}{2}} = 1.$$

We can solve this equation to obtain $\theta_{\max}$

$$\theta_{\max} = \frac{\ln(\phi\epsilon^{-1}) + \ln((1-\phi)(1-\epsilon)^{-1})}{\ln(\phi\epsilon^{-1}) - \ln((1-\phi)(1-\epsilon)^{-1})}.$$

We get the assertion because ϱ is always larger or equal $\theta_{\max}$. □

From Eq. (6) we can see the interaction between ϕ and ϵ: if the difference between ϵ and ϕ is small, then the right hand side of (6) is small. The smaller ϕ is, the more important this difference is. From Theorem 7.2 of Breiman (1997b) we also have the weaker bound $\varrho \geq 1 - 2\phi$ and so, if ϕ is small then ϱ must be large, i.e. choosing a small ϕ results in a larger margin on the training patterns. On the other hand, an increase of the complexity of the basis algorithm leads to an increased ϱ, because the error ϵ_t will decrease.

2.4.2. Support patterns. A decrease in the functional $G(\mathbf{c}, |\mathbf{b}|) := G(\mathbf{b})$ (with $\mathbf{c} = \mathbf{b}/|\mathbf{b}|$) is predominantly achieved by improvements of the margin $\rho(\mathbf{z}_i, \mathbf{c})$. If the margin $\rho(\mathbf{z}_i, \mathbf{c})$ is negative, then the error $G(\mathbf{c}, |\mathbf{b}|)$ clearly takes a large value, amplified by $|\mathbf{b}|$ in the exponent. So, ATA tries to decrease the negative margins most efficiently in order to improve the error $G(\mathbf{c}, |\mathbf{b}|)$.

Now let us consider the asymptotic case, where the number of iterations and therefore $|\mathbf{b}|$ take large values (cf. Lemma 2). Here, the margins $\rho(\mathbf{z}_i, \mathbf{c})$ of all patterns $\mathbf{z}_1, \ldots, \mathbf{z}_l$, are almost the same and small differences are amplified strongly in $G(\mathbf{c}, |\mathbf{b}|)$. For example, given two margins $\rho(\mathbf{z}_i, \mathbf{c}) = 0.2$ and $\rho(\mathbf{z}_j, \mathbf{c}) = 0.3$ and $|\mathbf{b}| = 100$, then this difference is amplified exponentially $\exp\{-\frac{100\times 0.2}{2}\} = e^{-10}$ and $\exp\{-\frac{100\times 0.3}{2}\} = e^{-15}$ in $G(\mathbf{c}, |\mathbf{b}|)$, i.e. by a factor of $e^5 \approx 150$. From Eq. (4) we see that as soon as the annealing parameter $|\mathbf{b}|$ is large, the ATA learning becomes a hard competition case: only the patterns with smallest margin will get high weights, other patterns are effectively neglected in the learning process. We get the following interesting lemma.

Lemma 5. *During the ATA learning process, the smallest margin of the (training) patterns of each class will asymptotically converge to the same value, i.e.*

$$\lim_{t\to\infty} \min_{i:y_i=1} \rho(\mathbf{z}_i, \mathbf{c}^t) = \lim_{t\to\infty} \min_{i:y_i=-1} \rho(\mathbf{z}_i, \mathbf{c}^t), \tag{7}$$

if the following assumptions are fulfilled:

1. *the weight of each hypothesis is bounded from below and above by*

$$0 < \gamma < b_t < \Gamma < \infty, \text{ and} \tag{8}$$

2. *the learning algorithm must (in principle) be able to classify all patterns to one class $c \in \{\pm 1\}$, if the sum over the weights of patterns of class c is larger than a constant δ, i.e.*

$$\sum_{i:y_i=c} w(\mathbf{z}_i) > \delta \Rightarrow h(\mathbf{x}_i) = c \quad (i = 1, \ldots, l). \tag{9}$$

The proof can be found in Appendix C.

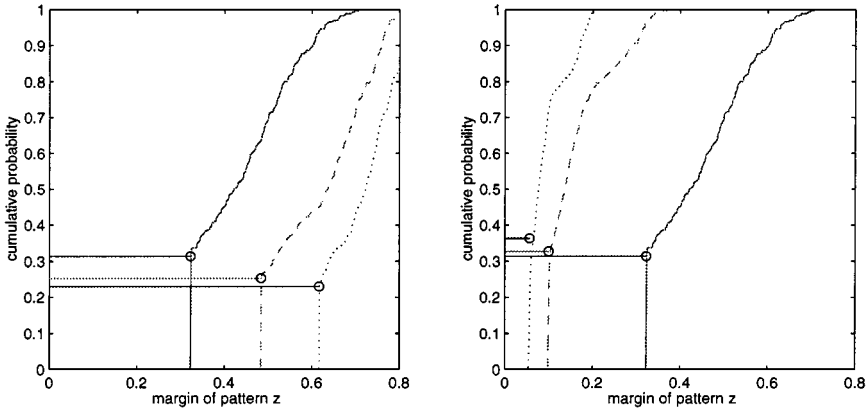

Figure 4. Margin distributions of ADABOOST for different noise levels: $\sigma^2 = 0\%$ (dotted), 9% (dashed), 16% (solid) with RBF nets (13 centers) as base hypotheses (left); and with 7 (dotted), 13 (dashed), 30 (solid) centers in the base hypotheses for data with $\sigma^2 = 16\%$ (right) after 10^4 ADABOOST iterations ($\phi = 1/2$). These graphs are an experimental confirmation of the trends expected from Eq. (6).

Note 6. Assumption 2 of Lemma 5 ensures, that the classifier is in principle able to misclassify every single pattern. Effectively, this assumption introduces something like a bias term b that is automatically adjusted if the smallest margins of one class are significantly different from the other class. In SVMs for the separable case (Boser et al., 1992), b is directly computed such that the smallest margins of both classes are the same.

Therefore, the ATA learning process converges, under rather mild assumptions, to a solution where a subset of the training patterns has asymptotically the same smallest margin. We call these patterns *Support Patterns* (SPs) (cf. figure 4).

To validate our theoretical analysis we performed numerical simulations on toy data with an asymptotic number (10^4) of boosting steps. The training data was generated from several (non-linearly transformed) Gaussian and uniform blobs,[4] which were additionally distorted by uniformly distributed noise $U(0.0, \sigma^2)$. In our simulations, we used 300 patterns and σ^2 is either 0%, 9%, or 16%.

In all simulations, radial basis function (RBF) networks with adaptive centers are used as base learners (cf. Appendix D or Müller et al., 1998 for a detailed description). Figure 4 shows the margin distributions after 10^4 boosting iterations at different noise levels σ^2 (left) and for different strengths of the base hypotheses (right). From these figures, it becomes apparent that the margin distribution asymptotically makes a step at a specific margin size and that some subset of the training patterns all have similar margins that correspond to the minimal margin discussed above. If the noise level is high or the complexity of the base hypothesis is low, one gets higher training errors ϵ_t and therefore a smaller value of ϱ. These numerical results support our theoretical asymptotic analysis. Interestingly, the margin distributions of ATAs resembles the ones of SUPPORT VECTOR MACHINES for the separable case (Boser et al., 1992; Cortes & Vapnik, 1995; Vapnik, 1995, cf. figure 5). In our toy example (cf. figure 6) we show the decision lines of SVMs and ATAs. We note a very high overlap between the patterns that become support vectors (SVs) (cf. figure 6

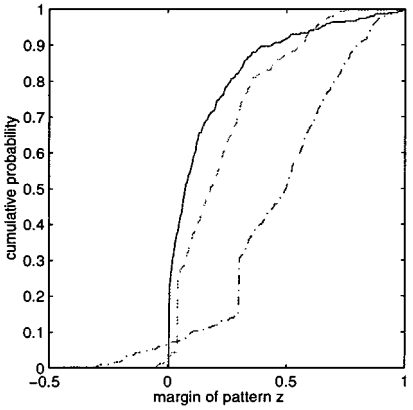

Figure 5. Typical margin distribution graphs (normalized) of a SVM with hard margin (solid) and soft margin with $C = 10^{-3}$ (dashed) and $C = 10^{-1}$ (dash-dotted). Here for the same toy example a RBF kernel (width $= 0.3$) is used. The generalization error of the SVM with hard margin is more than two times larger as with $C = 10^{-1}$.

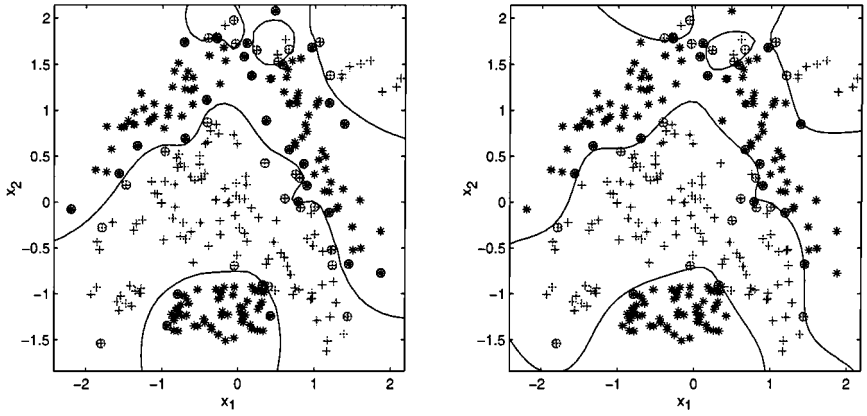

Figure 6. Training patterns with decision lines for ADABOOST (left) with RBF nets (13 centers) and SVM (right) for a low noise case with similar generalization errors. The positive and negative training patterns are shown as '$+$' and '$*$' respectively, the support patterns and support vectors are marked with 'o'.

right) and the patterns that lie within the step part of the margin distribution for ATA (cf. figure 4 left).

So, the ADABOOST-type algorithm achieves asymptotically a decision with *hard margin*, very similar to the one of SVMs for the separable case. Intuitively this is clear: the most difficult patterns are emphasized strongly and become support patterns or support vectors asymptotically. The degree of overlap between the support vectors and support patterns depends on the kernel (SVM) and on the base hypothesis (ATA) being used. For the SUPPORT VECTOR MACHINE with RBF kernel the highest overlap was achieved, when the average widths of the RBF networks was used as kernel width for the SUPPORT VECTOR MACHINE

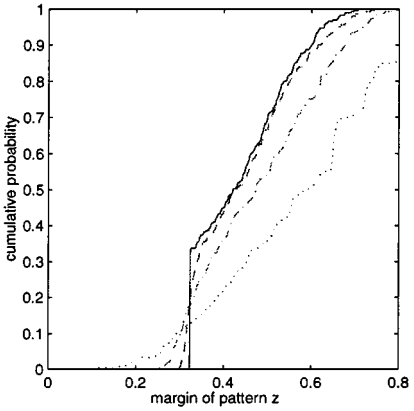

Figure 7. Typical margin distribution graphs of (original) ADABOOST after 20 (dotted), 70 (dash-dotted), 200 (dashed) and 10^4 (solid) iterations. Here, the toy example (300 patterns, $\sigma = 16\%$) and RBF networks with 30 centers are used. After already 200 iterations the asymptotical convergence is almost reached.

(Rätsch, 1998). We have also observed this striking similarity of SPs in ADABOOST and SVs of the SVM in several other non-toy applications.

In the sequel, we can often assume the asymptotic case, where a hard margin is achieved (the more hypotheses we combine, the better is this approximation). Experimentally we find that the hard margin approximation is valid (cf. Eq. (4)) already for e.g. $|\mathbf{b}| > 100$. This is illustrated by figure 7, which shows some typical ATA margin distributions after 20, 70, 200 and 10^4 iterations.
To recapitulate our findings of this section:

1. ADABOOST-type algorithms aim to minimize a functional, which depends on the margin distribution. The minimization is done by means of a constraint gradient descent with respect to the margin.
2. Some training patterns, which are in the area of the decision boundary, have asymptotically (for a large number of boosting steps) the same margin. We call these patterns Support Patterns. They have a large overlap to the SVs found by a SVM.
3. Asymptotically, ATAs reach a hard margin comparable to the one obtained by the original SVM approach (Boser et al., 1992).
4. Larger hard margins can be achieved, if ϵ_t (more complex base hypotheses) and/or ϕ are small (cf. Corollary 4). For the low noise case, a choice of $\theta \neq \frac{1}{2}$ can lead to a better generalization performance, as shown for e.g. OCR benchmark data in Onoda et al. (1998).

3. Hard margin and overfitting

In this section, we give reasons why the ATA is *not noise robust* and exhibits suboptimal generalization ability in the presence of noise. According to our understanding, noisy data has at least one of the following properties: (a) overlapping class probability distributions,

(b) outliers and (c) mislabeled patterns. All three types of noise appear very often in data analysis. Therefore the development of a noise robust version of ADABOOST is very important.

The first theoretical analysis of ADABOOST in connection with margin distributions was done by Schapire et al. (1997). Their main result is a bound on the generalization error $\mathbf{P}_{\mathbf{z}\sim\mathcal{D}}[\rho(\mathbf{z}) \leq 0]$ depending on the VC-dimension d of the base hypotheses class and on the margin distribution on the training set. With probability at least $1-\delta$

$$\mathbf{P}_{\mathbf{z}\sim\mathcal{D}}[\rho(\mathbf{z}) \leq 0] \leq \mathbf{P}_{\mathbf{z}\sim\mathbf{Z}}[\rho(\mathbf{z}) \leq \theta] + \mathcal{O}\left(\frac{1}{\sqrt{l}}\left(\frac{d\log^2(l/d)}{\theta^2} + \log(1/\delta)\right)\right) \tag{10}$$

is satisfied, where $\theta > 0$ and l denotes the number of patterns. It was stated that the reason for the success of ADABOOST, compared to other ensemble learning methods (e.g. BAGGING), is the maximization of the margin. The authors observed experimentally that ADABOOST maximizes the margin of patterns which are most difficult, i.e. have the smallest margin and that on the other hand by increasing the minimum margin of a few patterns, the margin of the rest of the other patterns is also reduced.

In Breiman (1997b), the connection between maximizing the smallest margin and a good generalization error was analyzed experimentally and could not be confirmed for noisy data.

In Grove and Schuurmans (1998) the Linear Programming (LP) approach of Freund and Schapire (1996) and Breiman (1997b) was extended and used to maximize the smallest margin of an existing ensemble of classifiers. Several experiments with LP-ADABOOST on UCI benchmarks (often noisy data) were made and it was unexpectedly observed that LP-ADABOOST performs in almost all cases worse than the original ADABOOST algorithm, even though the smallest observed margins were larger.

Our experiments have shown that as the margin increases, the generalization performance becomes better on data sets with almost no noise (e.g. OCR, cf. Onoda et al., 1998), however, we also observe that ADABOOST overfits on noisy data (for a moderate number of combined hypotheses).

As an example for overlapping classes, figure 8 (left) shows a typical overfitting behavior in the generalization error for ADABOOST on the same data as in Section 2. Here, already after only 80 boosting iterations the best generalization performance is achieved. From Eq. (6) we see that ADABOOST will asymptotically achieve a positive margin (for $\phi < \frac{1}{2}$) and all training patterns are classified according to their possibly wrong labels (cf. figure 8 (right)). However, this is at the expense that the complexity of the combined hypotheses increases and the decision surface becomes clearly less smooth. The achieved decision line is far away from the Bayes optimal line (cf. dashed line in figure 8 (right)).

To discuss the generally bad performance of hard margin classifiers in the presence of outliers and mislabeled patterns, we analyze the toy example in figure 9. Let us first consider the case without noise (left). Here, we can estimate the optimal separating hyperplane correctly. In figure 9 (middle) we have one outlier, which corrupts the estimation. The ADABOOST-type algorithm will certainly concentrate its weights on this outlier and spoil the good estimate that we would get without outlier. Next, let us consider more complex decision lines. Here the overfitting problem gets even more distinct, if we can generate

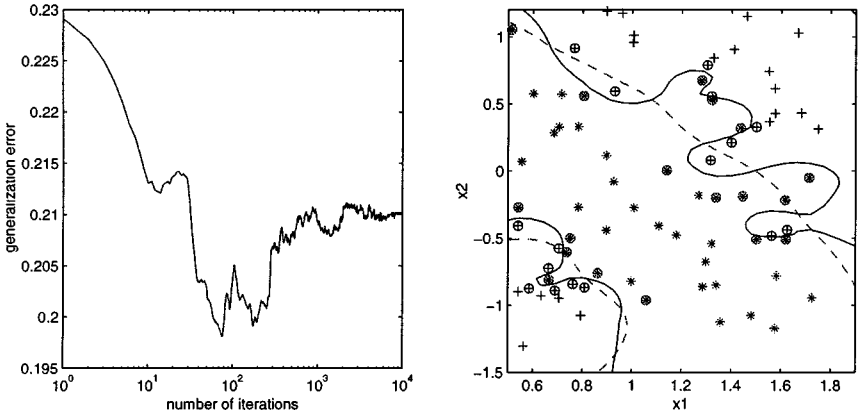

Figure 8. Typical overfitting behavior in the generalization error (smoothed) as a function of the number of iterations (left; log scale) and a typical decision line (right) generated by ADABOOST (10^4 iterations) using RBF networks (30 centers) in the case of noisy data (300 patterns, $\sigma^2 = 16\%$). The positive and negative training patterns are shown as '$+$' and '$*$' respectively, the support patterns are marked with 'o'. An approximation to the Bayes decision line is plotted dashed.

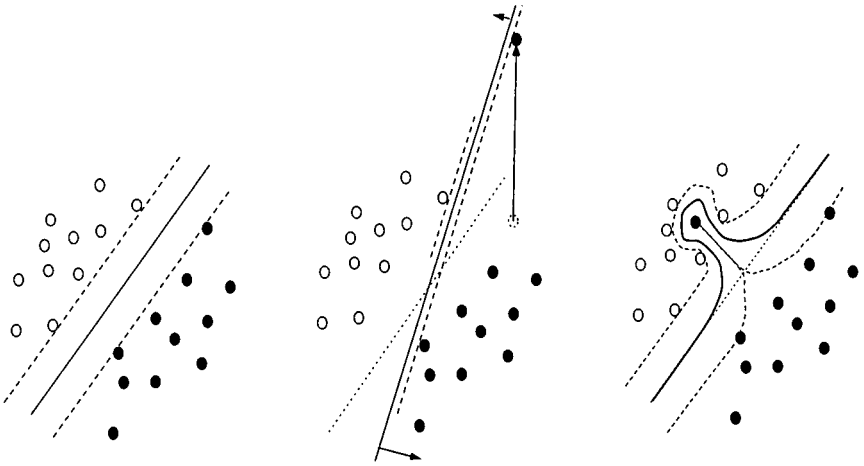

Figure 9. The problem of finding a maximum margin "hyper-plane" on reliable data (left), data with outlier (middle) and with a mislabeled pattern (right). The solid line shows the resulting decision line, whereas the dashed line marks the margin area. In the middle and on the left the original decision line is plotted with dots. The hard margin implies noise sensitivity, because only one pattern can spoil the whole estimation of the decision line.

more and more complexity by combining a lot of hypotheses. Then all training patterns (even mislabeled ones or outliers) can be classified correctly. In figure 8 (right) and figure 9 (right) we see that the decision surface is rather rough and gives bad generalization.

From these cartoons, it becomes apparent that ATA is noise sensitive and maximizing the smallest margin in the case of noisy data can (and will) lead to bad results. Therefore, we need to relax the hard margin and allow for a possibility of mistrusting the data.

From the bound (10) it is indeed not obvious that we should maximize the smallest margin: the first term on the right hand side of Eq. (10) takes the whole margin distribution into account. If we would allow a non-zero training error in the settings of figure 9, then the first term of the right hand side of (10) becomes non-zero ($\theta > 0$). But then θ can be larger, such that the second term is much smaller. In Mason et al. (2000a) and Mason et al. (2000b) similar bounds were used to optimize the margin distribution (a piecewise linear approximation) directly. This approach, similar in spirit than ours, is more successful on noisy data than the simple maximization of the smallest margin.

In the following we introduce several possibilities to mistrust parts of the data, which leads to the *soft margin* concept.

4. Improvements using a soft margin

Since the original SVM algorithm (Boser et al., 1992) assumed separable classes and pursued a hard margin strategy, it had similarly poor generalization performance on noisy data as the ATAs. Only the introduction of soft margins for SUPPORT VECTOR MACHINES (Cortes & Vapnik, 1995) allowed them to achieve much better generalization results (cf. figure 5).

We will now show how to use the soft margin idea for ATAs. In Section 4.1 we modify the error function from Eq. (2) by introducing a new term, which controls the importance of a pattern in the reweighting scheme. In Section 4.2 we demonstrate that the soft margin idea can be directly build into the LP-ADABOOST algorithm and in Section 4.3 we show an extension to quadratic programming—QP-ADABOOST—with its connections to the support vector approach.

In the following subsections and also in the experimental section we will only consider the case $\phi = \frac{1}{2}$. Generalizing to other values of ϕ is straightforward.

4.1. Margin vs. influence of a pattern

First, we propose an improvement of the original ADABOOST by using a regularization term in (2) in analogy to weight decay in neural networks and to the soft margin approach of SVMs.

From Corollary 4 and Theorem 2 of Breiman (1997b), all training patterns will get a margin $\rho(\mathbf{z}_i)$ larger than or equal to $1-2\phi$ after asymptotically many iterations (cf. figure 3 and discussion in Section 2). From Eq. (2) we can see that $G(\mathbf{b})$ is minimized as ϱ is maximized, where

$$\rho(\mathbf{z}_i, \mathbf{c}) \geq \varrho \quad \text{for all } i = 1, \ldots, l. \tag{11}$$

After many iterations, these inequalities are satisfied for a ϱ that is larger or equal than the margin given in Corollary 4. If $\varrho > 0$ (cf. Corollary 4), then all patterns are classified according to their possibly wrong labels, which leads to overfitting in the presence of noise. Therefore, any modification that improves ADABOOST on noisy data, must not force all margins beyond 0. Especially those patterns that are mislabeled and usually more difficult to classify, should be able to attain margins smaller than 0.

If we knew beforehand which patterns were unreliable we could just remove them from the training set or, alternatively, we could not require that they have a large margin (cf. also the interesting approach for regression in Breiman (1999)). Suppose we have defined a non-negative quantity $\zeta(\mathbf{z}_i)$, which expresses our "mistrust" in a pattern $\mathbf{z}_i$. For instance, this could be a probability that the label of a pattern is incorrect. Then we can relax (11) and get

$$\rho(\mathbf{z}_i, \mathbf{c}) \geq \varrho - C\zeta(\mathbf{z}_i), \tag{12}$$

where C is an a priori chosen constant. Furthermore, we can define the *soft margin* $\tilde{\rho}(\mathbf{z}_i)$ of a pattern $\mathbf{z}_i$ as a tradeoff between the margin and $\zeta(\mathbf{z}_i)$

$$\tilde{\rho}(\mathbf{z}_i, \mathbf{c}) := \rho(\mathbf{z}_i, \mathbf{c}) + C\zeta(\mathbf{z}_i) \tag{13}$$

and from Eq. (12) we obtain

$$\tilde{\rho}(\mathbf{z}_i, \mathbf{c}) \geq \varrho. \tag{14}$$

Now we can again simply maximize the smallest *soft* margin (i.e. maximize ϱ) and we expect to observe less overfitting. The problem now is, how to define $\zeta(\mathbf{z}_i)$. We restrict ourselves here by presenting only one definition of ζ based on the *influence* of a pattern on the combined hypotheses h_r

$$\mu_t(\mathbf{z}_i) = \sum_{r=1}^{t} c_r w_r(\mathbf{z}_i),$$

which is the average weight of a pattern computed during the ATA learning process (cf. pseudo-code in figure 1). The rationale is: a pattern which is very often misclassified (i.e. difficult to classify) will have a high average weight, i.e. a high influence.[5] Interestingly, in the noisy case there is (usually) a high overlap between patterns with high influence and mislabeled patterns (or other patterns very near to or just beyond the decision line).

In the SVM approach it turns out that introducing slack variables to the quadratic optimization problem (Cortes & Vapnik, 1995) is the same as introducing a upper bound on the Lagrange multipliers of the patterns (Cortes & Vapnik, 1995; Vapnik, 1995; Schölkopf, 1997). Empirical evidence shows that the influence of a pattern $\mu_t(\mathbf{z}_i)$ is very similar to a Lagrange multiplier in LP-ADABOOST (Grove & Schuurmans, 1998), since it indicates how much the pattern contributes to the decision. Lagrange multipliers of patterns that are not support patterns in the linear program will be 0 and the influence of a non support pattern will also converge asymptotically to 0 (for $t \to \infty$). Furthermore, we found experimentally that both numerical values coincide within a small range (details on this connection can be found in Rätsch et al., 2000).

From this discussion it becomes apparent that it makes sense to mistrust patterns with high influences in the noisy case. From this we define ζ by

$$\zeta(\mathbf{z}_i) \equiv \mu_t(\mathbf{z}_i)^p, \tag{15}$$

such that the influence of a pattern is penalized, where p is an exponent chosen a priori (for example choose $p = 1$ or 2).[6] If a training pattern has high weights $\zeta(\mathbf{z}_i)$, then also the soft

margin is increasing. If we now maximize the smallest soft margin, we do not force outliers to be classified according to their possibly wrong labels, but we allow for some errors. Our prior for the choice (15) is to weight all patterns equally. This counterbalances the tendency of ATAs to overweight certain patterns. So we tradeoff between margin and influence.

Note 7. If we choose $C = 0$ in Eq. (12), the original ADABOOST algorithm is retrieved. If C is chosen high, then each single data point is "not taken very seriously" and we observe empirically that the number of support patterns increases. For $C \to \infty$ we (almost) retrieve the BAGGING algorithm (Breiman, 1996) (in this case, the pattern weighting will be always uniform).

Of course, other functional forms of ζ are also possible (see also Rätsch et al., 2000), for instance $\zeta^t(\mathbf{z}_i) = \mathbf{P} f^t(\mathbf{x}_i)$, where $\mathbf{P}$ is an arbitrary regularization operator. With $\mathbf{P}$ it is possible to incorporate (other) prior knowledge about the problem into ATAs like smoothness of the decision surface much in the spirit of Tikhonov regularizers (e.g. Tikhonov & Arsenin, 1977; Smola, Schölkopf, & Müller, 1998; Rokui & Shimodaira, 1998).

Now we can reformulate the ATA optimization process in terms of soft margins. From Eq. (14) and the definition in (15) we can easily derive the new error function (cf. Eq. (2)), which aims to maximize the soft margin (we assume $\phi = \frac{1}{2}$):

$$\begin{aligned} G_{Reg}(\mathbf{b}^t) &= \sum_{i=1}^{l} \exp\left\{-\frac{1}{2}\tilde{\rho}(\mathbf{z}_i, \mathbf{b}^t)\right\} \\ &= \sum_{i=1}^{l} \exp\left\{-\frac{1}{2}|\mathbf{b}^t|[\rho(\mathbf{z}_i, \mathbf{c}^t) + C\mu_t(\mathbf{z}_i)^p]\right\}. \end{aligned} \tag{16}$$

The weight $w_{t+1}(\mathbf{z}_i)$ of a pattern is computed as the derivative of Eq. (16) with respect to $\tilde{\rho}(\mathbf{z}_i, \mathbf{b}^t)$ (cf. Lemma 1)

$$w_{t+1}(\mathbf{z}_i) = \frac{1}{Z_t}\frac{\partial G_{Reg}(\mathbf{b}^t)}{\partial \tilde{\rho}(\mathbf{z}_i, \mathbf{b}^t)} = \frac{\exp\{-\tilde{\rho}(\mathbf{z}_i, \mathbf{b}^t)/2\}}{\sum_{j=1}^{l} \exp\{-\tilde{\rho}(\mathbf{z}_j, \mathbf{b}^t)/2\}}, \tag{17}$$

where Z_t is a normalization constant such that $\sum_{i=1}^{l} w_{t+1}(\mathbf{z}_i) = 1$. For $p = 1$ we get the update rule for the weight of a training pattern in the t-th iteration (for details cf. Rätsch, 1998) as

$$w_{t+1}(\mathbf{z}_i) = \frac{w_t(\mathbf{z}_i)}{Z_t}\exp\{b_t \mathrm{I}(y_i \neq h_t(\mathbf{x}_i)) - C\zeta^t(\mathbf{z}_i)|\mathbf{b}^t|\}, \tag{18}$$

and for $p = 2$ we obtain

$$w_{t+1}(\mathbf{z}_i) = \frac{w_t(\mathbf{z}_i)}{Z_t}\exp\{b_t \mathrm{I}(y_i \neq h_t(\mathbf{x})) - C\zeta^t(\mathbf{z}_i)|\mathbf{b}^t| + C\zeta^{t-1}(\mathbf{z}_i)|\mathbf{b}^{t-1}|\}, \tag{19}$$

where Z_t is again a normalization constant. It is more difficult to compute the weight b_t of the t-th hypothesis analytically. However, we can get b_t efficiently by a line search

Algorithm $\text{ADABOOST}_{\text{REG}}(\mathbf{Z}, T, C, p)$

Input: l examples $\mathbf{Z} = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l) \rangle$

Initialize: $w_1(\mathbf{z}_i) = 1/l$ for all $i = 1 \ldots l$

Do for $t = 1, \ldots, T$,

1. Train classifier with respect to the weighted sample set $\{\mathbf{Z}, \mathbf{w}_t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1 \ldots 1]$
2. Find the weight of the hypothesis b_t by minimizing (16):

$$b_t = \operatorname*{argmin}_{b_t \geq 0} \sum_{i=1}^{l} \exp\left\{ -\frac{1}{2} \left[\rho(\mathbf{z}_i, \mathbf{b}^t) + C|\mathbf{b}^t|\mu_t(\mathbf{z}_i)^p \right] \right\} \quad \text{(F11.1)}$$

abort if

$$b_t = 0 \quad \text{or} \quad b_t \geq \Gamma \ , \quad \text{(F11.2)}$$

where Γ is a large constant

3. Update weights $\mathbf{w}_t$:

$$w_{t+1}(\mathbf{z}_i) = \frac{1}{Z_t} \exp\left\{ -\frac{1}{2} \left[\rho(\mathbf{z}_i, \mathbf{b}^t) + C|\mathbf{b}^t|\mu_t(\mathbf{z}_i)^p \right] \right\} \quad \text{(F11.3)}$$

where Z_t is a normalization constant, such that $\sum_{i=1}^{l} w_{t+1}(\mathbf{z}_i) = 1$ (cf. Equations (18) and (19)).

Output: Final hypothesis with weights $\mathbf{b}$ as in Fig. 1

Figure 10. The $\text{ADABOOST}_{\text{REG}}$ (ABR) algorithm (Rätsch, 1998; Rätsch, Onoda, & Müller, 1999), where C is a regularization constant and p is a parameter that changes the regularization characteristics. In all simulations in Section 5 we used $p = 2$. An implementation can be downloaded from `http://ida.first.gmd.de/~raetsch`.

procedure (e.g. Press et al., 1992) minimizing (16), which has a unique solution because $\frac{\partial}{\partial b_t} G_{Reg}(\mathbf{b}^t) > 0$ is satisfied for $b_t > 0$. An algorithmic formulation can be found in figure 10.

We can interpret this approach as regularization analogous to weight decay. Our prior is that some patterns are likely not to be reliable, so in the noisy case we prefer hypotheses which do not rely on only a few patterns with high weights.[7] Instead, we are looking for hypotheses with smaller values of $\zeta(\mathbf{z}_i)$. So by this regularization, ADABOOST is not changed for easily classifiable patterns, but only for the most difficult ones.

The variables $\zeta(\mathbf{z}_i)$ in Eq. (12) can also be interpreted as slack-variables (cf. SVM approach and next section), which are non-linearly involved in the error function. Large values of $\zeta(\mathbf{z}_i)$ for some patterns allow for a larger (soft-) margin ϱ. For a comparison of the soft margin distributions of a single RBF classifier and $\text{ADABOOST}_{\text{REG}}$ see figure 11.

Summarizing, our modification of ADABOOST constructs a soft margin to avoid overfitting.

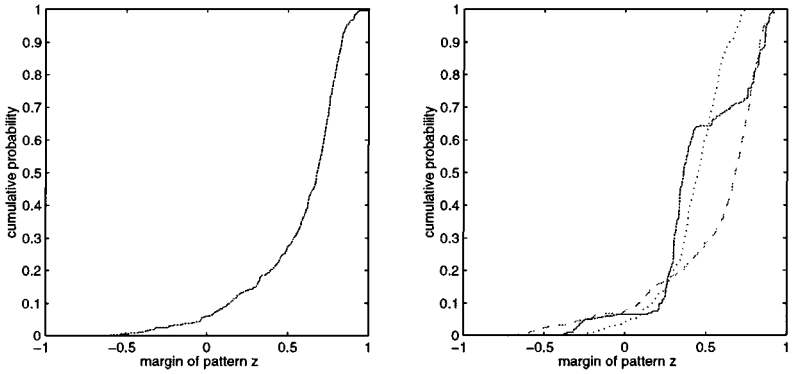

Figure 11. Margin distribution graphs of the RBF base hypothesis (scaled) trained with Squared Error (left) and ADABOOST$_{\text{REG}}$ (right) with different values of C for the toy data set after 1000 iterations. Note that for some values for C the graphs of ADABOOST$_{\text{REG}}$ are quite similar to the graphs of the single RBF net.

4.2. Linear programming with slack variables

Grove and Schuurmans (1998) showed how to use linear programming to maximize the smallest margin for a given ensemble and proposed LP-ADABOOST (cf. Eq. (21)). In their approach, they first compute a margin (or gain) matrix $M \in \{\pm 1\}^{l \times T}$ for the given hypotheses set, which is defined by

$$M_{i,t} = y_i h_t(\mathbf{x}_i). \tag{20}$$

M defines which hypothesis contributes a positive (or negative) part to the margin of a pattern and is used to formulate the following max-min problem: find a weight vector $\mathbf{c} \in \mathbb{R}^T$ for hypotheses $\{h_t\}_{t=1}^T$, which maximizes the smallest margin $\varrho := \min_{i=1,\ldots,l} \rho(\mathbf{z}_i)$. This problem can be solved by linear programming (e.g. Mangasarian, 1965):

Maximize ϱ subject to

$$\begin{aligned} \sum_{t=1}^{T} M_{i,t} c_t &\geq \varrho \quad i = 1, \ldots, l \\ c_t &\geq 0 \quad t = 1, \ldots, T \\ \sum_{t=1}^{T} c_t &= 1. \end{aligned} \tag{21}$$

This LP-ADABOOST algorithm achieves a larger hard margin than the original ADABOOST algorithm, however in this form it cannot hope to generalize well on noisy data (see our discussion in Section 3). Therefore we also define a soft-margin for a pattern $\tilde{\rho}'(\mathbf{z}_i) = \rho(\mathbf{z}_i) + \xi_i$, which is technically equivalent to the introduction of slack variables ξ_i and we arrive at the algorithm LP$_{\text{REG}}$-ADABOOST (Rätsch, 1998; Rätsch et al., 1999; Rätsch, Onoda, & Müller, 1998). To avoid large values of the slack variables, while solving

Algorithm $\mathrm{LP}_{\mathrm{REG}}$-ADABOOST$(\mathbf{Z}, T, C)$

Input: l examples $\mathbf{Z} = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\rangle$

1. Run ADABOOST and obtain T hypothesis $h_1, \ldots, h_T$
2. Compute margin matrix M according to Eq. (20)
3. Compute $\mathbf{c}$ by maximizing $\varrho - \frac{C}{l}\sum_{i=1}^{l}\xi_i$ subject to

$$\begin{aligned} \sum_{t=1}^{T} c_t M_{i,t} &\geq \varrho - \xi_i, & i &= 1, \ldots, l \\ \xi_i &\geq 0, & i &= 1, \ldots, l \\ c_t &\geq 0, & t &= 1, \ldots, T \\ \sum_{t=1}^{T} c_t &= 1 \end{aligned}$$

Output: Final hypothesis with weights $\mathbf{c}$ as in Fig. 1.

Figure 12. The $\mathrm{LP}_{\mathrm{REG}}$-ADABOOST algorithm, where C is a regularization constant.

the linear program with slack variables, the sum of all ξ_i is penalized in the objective function (cf. pseudo-code in figure 12). This modification allows that some patterns have smaller margins than ϱ. There is a tradeoff controlled by the constant C: (a) make all margins larger than ϱ and (b) maximize $\varrho - \frac{C}{l}\sum_i \xi_i$.

Our algorithm is related to the LP-SVM approach (Schölkopf, Smola, & Williamson, 2000). As in the original SVM approach, the Lagrange multipliers will be sparse and again we get support vector/patterns. Interestingly, it turns out in both approaches (asymptotically, i.e. with the number of patterns) that $\nu := \frac{1}{C} \in [0 \ldots 1]$ is an upper bound on the fraction of misclassified samples and a lower bound on the fraction of support vectors (Rätsch et al., 2000).

4.3. Quadratic programming and the connection to support vector machines

In the following section, we extend the $\mathrm{LP}_{\mathrm{REG}}$-ADABOOST (LPR) algorithm to quadratic programming by using similar techniques as in SUPPORT VECTOR MACHINES (Boser et al., 1992; Cortes & Vapnik, 1995; Mangasarian, 1965). This gives interesting insights to the connection between SUPPORT VECTOR MACHINES and ADABOOST.

We start by transforming the $\mathrm{LP}_{\mathrm{REG}}$-ADABOOST algorithm, which maximizes ϱ, while $|\mathbf{c}|$ is kept fixed, to a linear program in which ϱ is fixed (to e.g. 1) and $|\mathbf{b}|$ is minimized. Unfortunately, there is no equivalent linear program because of the slack variables. But we can use a Taylor expansion[8] to get the following linear program (compare with linear programming approaches related to SV learning e.g. Bennett & Mangasarian, 1992; Weston

et al., 1997; Frieß & Harrison, 1998; Bennett, 1998):

$$\text{Minimize } \|\mathbf{b}\|_1 + C\sum_i \xi_i \text{ subject to}$$

$$\begin{aligned} \sum_{t=1}^{T} b_t M_{i,t} &\geq 1 - \xi_i, \quad t = 1, \ldots, T, \\ b_t &\geq 0, \quad t = 1, \ldots, T, \\ \xi_i &\geq 0, \quad i = 1, \ldots, l. \end{aligned} \tag{22}$$

Essentially, this is the same algorithm as in figure 12: for a different value of C problem, (22) is equivalent to the one in figure 12 (cf. Smola, 1998 and Lemma 3 in Rätsch et al., 2000). Instead of using the ℓ_1-norm in the optimization objective of (22), we can also use the ℓ_p-norm. Clearly, each p will imply its own soft margin characteristics. Using $p = 2$ leads to an algorithm similar to the SVM (cf. figure 14).

The optimization objective of a SVM is to find a function $h^{\mathbf{w}}$ which minimizes a functional of the form (Vapnik, 1995)

$$E = \|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \xi_i, \tag{23}$$

subject to the constraints

$$y_i h(\mathbf{x}_i) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0, \text{ for } i = 1, \ldots, l.$$

Here, the variables ξ_i are the slack-variables responsible for obtaining a soft margin. The norm of the parameter vector $\mathbf{w}$ defines a system of nested subsets in the combined hypothesis space and can be regarded as a measure of the complexity (as also the size of the margin

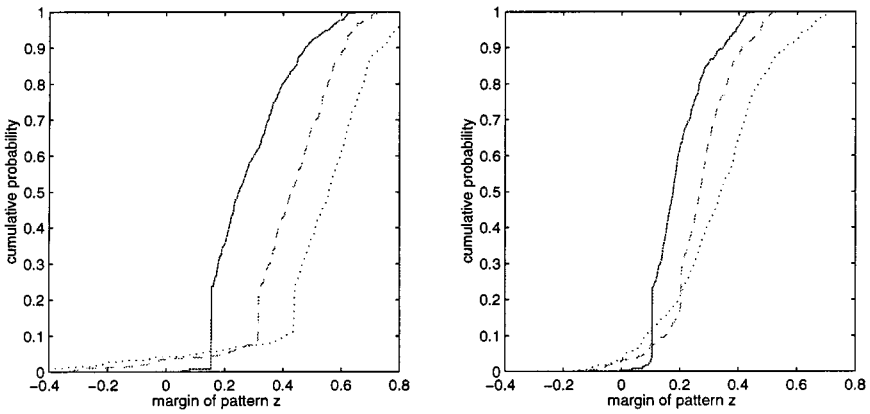

Figure 13. Margin distribution graphs of LP_{REG}-ADABOOST (left) and QP_{REG}-ADABOOST (right) for different values of C for the toy data set after 1000 iterations. LP_{REG}-ADABOOST sometimes generates margins on the training set, which are either 1 or -1 (step in the distribution).

of hypothesis $h^{\mathbf{w}}$) (Vapnik, 1995). With functional (23), we get a tradeoff (controlled by C) between the complexity of the hypothesis ($\|\mathbf{w}\|^2$) and the degree how much the classification may differ from the labels of the training patterns ($\sum_i \xi_i$).

For ensemble learning, so far we do not have such a measure of complexity. Empirically, we observed that the more different the weights are for the hypotheses, the higher the complexity of the ensemble. With this in mind, we can use the ℓ_p norm ($p > 1$) of the hypotheses weight vector $\|\mathbf{b}\|_p$ as a complexity measure. Let us assume, for example that we have $|\mathbf{b}| = 1$, then $\|\mathbf{b}\|_p$ this is a small value, as the elements of $\mathbf{b}$ are approximately equal (analogous to BAGGING). If $\|\mathbf{b}\|_p$ has high values, then there are some strongly emphasized hypotheses (far away from BAGGING).[9] Hence, we can apply the optimization principles of SVMs to ADABOOST and get a quadratic optimization problem in $\mathbf{b}$:

$$\text{Minimize } \|\mathbf{b}\|^2 + C \sum_i \xi_i,$$

with the constraints given in Eq. (22). We call this algorithm QP_{REG}-ADABOOST (QPR) since it was motivated by the connection to LP_{REG}-ADABOOST (cf. algorithm (22)) and by the analogy to the support vector algorithm (for pseudocode see figure 14). We expect a similar performance of QP_{REG} and LP_{REG}-ADABOOST with subtle differences on specific data sets due to the different "types" of soft margins. Furthermore, they should exhibit superior performance compared to the original ADABOOST algorithm on noisy data. For an overall comparison of the margin distributions of original ADABOOST, SVM, $\text{ADABOOST}_{\text{REG}}$ and LP/QP-ADABOOST see figures 5, 7, 11 and 13.

Summarizing, we introduced in this section the soft margin concept to ADABOOST by (a) regularizing the objective function (2), (b) LP_{REG}-ADABOOST, which uses slack variables and (c) QP_{REG}-ADABOOST, which exhibits an interesting connection to SVMs.

Algorithm QP_{REG}-ADABOOST$(\mathbf{Z}, T, C)$

Input: l examples $\mathbf{Z} = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\rangle$

1. Run ADABOOST and obtain T hypothesis $h_1, \ldots, h_T$
2. Compute margin matrix M according to Eq. (20)
3. Compute $\mathbf{b}$ by minimizing $\|\mathbf{b}\|^2 + \frac{C}{l}\sum_{i=1}^{l} \xi_i$,

$$\begin{aligned} \sum_{t=1}^{T} b_t M_{i,t} &\geq 1 - \xi_i, & t &= 1, \ldots, T, \\ b_t &\geq 0, & t &= 1, \ldots, T, \\ \xi_i &\geq 0, & i &= 1, \ldots, l\ . \end{aligned}$$

Output: Final hypothesis with weights $\mathbf{b}$ as in Eq. (F1.4).

Figure 14. The QP_{REG}-ADABOOST algorithm, where C is a regularization constant.

5. Experiments

In order to evaluate the performance of our new algorithms, we perform large scale simulations and compare the single RBF classifier, the original ADABOOST algorithm, $\text{ADABOOST}_{\text{REG}}$, L/QP_{REG}-ADABOOST and a SUPPORT VECTOR MACHINE (with RBF kernel).

5.1. Experimental setup

For this, we use 13 artificial and real world data sets from the UCI, DELVE and STATLOG benchmark repositories[10]: banana (toy data set used in the previous sections), breast cancer,[11] diabetes, german, heart, image segment, ringnorm, flare solar, splice, new-thyroid, titanic, twonorm, waveform. Some of the problems are originally not binary classification problems, hence a random partition into two classes is used.[12] At first we generate 100 partitions into training and test set (mostly $\approx 60\% : 40\%$). On each partition we train a classifier and then compute its test set error.

In all experiments, we combine 200 hypotheses. Clearly, this number of hypotheses is somewhat arbitrary and may not be optimal. However we checked that original ADABOOST with early stopping is most of the time worse than any of the proposed soft margin algorithms (cf. an earlier study Rätsch, 1998). However, we use a fixed number of iterations for all algorithms, therefore this comparison should be fair.

As base hypotheses we use RBF nets with adaptive centers as described in Appendix D. On each of the 13 data sets we employ cross validation to find the best base hypothesis model, which is then used in the ensemble learning algorithms. For selecting the best RBF model we optimize the number of centers (parameter K, cf. figure 15) and the number of iteration steps for adapting the RBF centers and widths (parameter O). The parameter λ was fixed to 10^{-6}.

The parameter C of the regularized versions of ADABOOST and the parameters (C, σ) of the SVM (C is the regularization constant and σ is the width of the RBF-kernel being used) are optimized on the first five realizations of each data set. On each of these realizations, a 5-fold-cross validation procedure gives a good model.[13] Finally, the model parameters are computed as the median of the five estimations and used throughout the training on all 100 realization of that data set. This way of estimating the parameters is computationally highly expensive, but it will make our comparison more robust and the results more reliable.

Note, to perform the simulations in this setup we had to train more than 3×10^6 adaptive RBF nets and to solve more than 10^5 linear or quadratic programming problems—a task that would have taken altogether 2 years of computing time on a single Ultra-SPARC machine, if we had not distributed it over 30 computers.

5.2. Experimental results

In Table 1 the average generalization performance (with standard deviation) over the 100 partitions of the data sets is given. The second last line in Table 1 showing 'Mean%', is

Algorithm RBF-Net(K, λ, O)

Input:

Sequence of labeled training patterns $\mathbf{Z} = \langle (\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l) \rangle$

Number of RBF centers K

Regularization constant λ

Number of iterations O

Initialize:

Run K-means clustering to find initial values for μ_k and determine $\sigma_k, k = 1, \ldots, K$, as the distance between μ_k and the closest μ_i $(i \neq k)$.

Do for $o = 1 : O$,

1. Compute optimal output weights $\mathbf{w} = \left(G^\top G + 2\frac{\lambda}{l}\mathbf{I}\right)^{-1} G^\top \mathbf{y}$
2a. Compute gradients $\frac{\partial}{\partial \mu_k} E$ and $\frac{\partial}{\partial \sigma_k} E$ as in (28) and (27) with optimal $\mathbf{w}$ and form a gradient vector $\mathbf{v}$
2b. Estimate the conjugate direction $\overline{\mathbf{v}}$ with Fletcher-Reeves-Polak-Ribiere CG-Method (Press et al., 1992)
3a. Perform a line search to find the minimizing step size δ in direction $\overline{\mathbf{v}}$; in each evaluation of E recompute the optimal output weights $\mathbf{w}$ as in line 1
3b. Update μ_k and σ_k with $\overline{\mathbf{v}}$ and δ

Output: Optimized RBF net

Figure 15. Pseudo-code description of the RBF net algorithm, which is used as base learning algorithm in the simulations with ADABOOST.

computed as follows: For each data set the average error rates of all classifier types are divided by the minimum error rate for this data set and 1 is subtracted. These resulting numbers are averaged over the 13 data sets and the variance is computed. The last line gives the Laplacian probability (and variance) over 13 data sets whether a particular method wins on a particular realization of a data set, i.e. has the lowest generalization error. Our experiments on noisy data (cf. Table 1) show that:

– The results of ADABOOST are in almost all cases worse than the single classifier. This is clearly due to the overfitting of ADABOOST. If early stopping is used then the effect is less drastic but still clearly observable (Rätsch, 1998).
– The averaged results for ADABOOST$_{\text{REG}}$ are a bit better (Mean% and Winner%) than the results of the SVM, which is known to be an excellent classifier. In five (out of seven) cases ADABOOST$_{\text{REG}}$ is significant better than the SVM. Moreover, the single

Table 1. Comparison among the six methods: Single RBF classifier, ADABOOST (AB), ADABOOST$_{\text{REG}}$ (AB$_R$; $p = 2$), L/QP$_{\text{REG}}$-ADABOOST (L/QP$_R$-AB) and a SUPPORT VECTOR MACHINE: Estimation of generalization error in % on 13 data sets (best method in bold face, second emphasized). The columns S_1 and S_2 show the results of a significance test (95%-t-test) between AB/AB$_R$ and AB$_R$/SVM, respectively. ADABOOST$_{\text{REG}}$ gives the best overall performance.

	RBF	AB	S_1	AB$_R$	LP$_R$-AB	QP$_R$-AB	S_2	SVM
Banana	*10.8 ± 0.6*	12.3 ± 0.7	+	10.9 ± 0.4	**10.7 ± 0.4**	10.9 ± 0.5	+	11.5 ± 0.7
B. Cancer	27.6 ± 4.7	30.4 ± 4.7	+	26.5 ± 4.5	26.8 ± 6.1	**25.9 ± 4.6**		*26.0 ± 4.7*
Diabetes	24.3 ± 1.9	26.5 ± 2.3	+	*23.8 ± 1.8*	24.1 ± 1.9	25.4 ± 2.2	+	**23.5 ± 1.7**
German	24.7 ± 2.4	27.5 ± 2.5	+	*24.3 ± 2.1*	24.8 ± 2.2	25.3 ± 2.1	−	**23.6 ± 2.1**
Heart	17.6 ± 3.3	20.3 ± 3.4	+	*16.5 ± 3.5*	17.5 ± 3.5	17.2 ± 3.4		**16.0 ± 3.3**
Image	3.3 ± 0.6	**2.7 ± 0.7**		**2.7 ± 0.6**	*2.8 ± 0.6*	**2.7 ± 0.6**		3.0 ± 0.6
Ringnorm	*1.7 ± 0.2*	1.9 ± 0.3	+	**1.6 ± 0.1**	2.2 ± 0.5	1.9 ± 0.2	+	*1.7 ± 0.1*
F. Solar	34.4 ± 2.0	35.7 ± 1.8	+	*34.2 ± 2.2*	34.7 ± 2.0	36.2 ± 1.8	−	**32**.4 ± **1.8**
Splice	*10.0 ± 1.0*	10.1 ± 0.5	+	**9.5 ± 0.7**	10.2 ± 1.6	10.1 ± 0.5	+	10.9 ± 0.7
Thyroid	*4.5 ± 2.1*	**4.4 ± 2.2**	−	4.6 ± 2.2	4.6 ± 2.2	**4.4 ± 2.2**		4.8 ± 2.2
Titanic	23.3 ± 1.3	*22.6 ± 1.2*		*22.6 ± 1.2*	24.0 ± 4.4	22.7 ± 1.1		**22.4 ± 1.0**
Twonorm	*2.9 ± 0.3*	3.0 ± 0.3	+	**2.7 ± 0.2**	3.2 ± 0.4	3.0 ± 0.3	+	3.0 ± 0.2
Waveform	10.7 ± 1.1	10.8 ± 0.6	+	**9.8 ± 0.8**	10.5 ± 1.0	10.1 ± 0.5		*9.9 ± 0.4*
Mean%	6.6 ± 5.8	11.9 ± 7.9		1.7 ± 1.9	8.9 ± 10.8	5.8 ± 5.5		4.6 ± 5.4
Winner%	14.8 ± 8.5	7.2 ± 7.8		26.0 ± 12.4	14.4 ± 8.6	13.2 ± 7.6		23.5 ± 18.0

RBF classifier wins less often than the SVM (for a comparison in the regression case cf. Müller et al., 1998).

– L/QP$_{\text{REG}}$-ADABOOST improves the results of ADABOOST. This is due to the use of a soft margin. But the results are not as good as the results of ADABOOST$_{\text{REG}}$ and the SVM. One reason is that the hypotheses generated by ADABOOST (aimed to construct a hard margin) may not provide the appropriate basis to subsequently generate a good soft margin with linear and quadratic programming approaches.
– We can observe that quadratic programming gives slightly better results than linear programming. This may be due to the fact that the hypotheses coefficients generated by LP$_{\text{REG}}$-ADABOOST are more sparse (smaller ensemble) and larger ensembles may have a better generalization ability (Breiman, 1998). Furthermore, with QP-ADABOOST we prefer ensembles which have approximately equally weighted hypotheses. As stated in Section 4.3, this implies a lower complexity of the combined hypothesis, which can lead to a better generalization performance.
– The results of ADABOOST$_{\text{REG}}$ are in ten (out of 13) cases significantly better than the results of ADABOOST. Also, in ten cases ADABOOST$_{\text{REG}}$ performs better than the single RBF classifier.

Summarizing, ADABOOST$_{\text{REG}}$ wins most often and shows the best average performance. In most of the cases it performs significantly better than ADABOOST and it performs

slightly better than SUPPORT VECTOR MACHINES. This demonstrates the noise robustness of the proposed algorithm.

The slightly inferior performance of SVM compared to ADABOOST$_{\text{REG}}$ may be explained with the fixed σ of the RBF-kernel for SVM. By fixing σ we look at the data only at *one scale*, i.e. we are losing possible multiscale information that could be inherent of the data. Further causes could be the coarse model selection, and the error function of the SV algorithm, which is not adapted to the noise model in the data (see Smola et al., 1998).

So, the original ADABOOST algorithm is useful for *low* noise cases, where the classes are easily separable (as shown for OCR cf. Schwenk & Bengio, 1997; LeCun et al., 1995). L/QP$_{\text{REG}}$-ADABOOST can improve the ensemble structure through introducing a soft margin and the same hypotheses (just with another weighting) can result in a much better generalization performance. The hypotheses, which are used by L/QP$_{\text{REG}}$-ADABOOST may be sub-optimal, because they are not part of the L/QP optimization process that aims for a soft margin. ADABOOST$_{\text{REG}}$ does not have this problem: the hypotheses are generated such that they are appropriate to form the desired soft-margin. ADABOOST$_{\text{REG}}$ extends the applicability of Boosting/Arcing methods to non-separable cases and should be preferably applied if the data is noisy.

6. Conclusion

We have shown that ADABOOST performs a constrained gradient descent in an error function that optimizes the margin (cf. Eq. (2)). Asymptotically, all emphasis is concentrated on the difficult patterns with small margins, easy patterns effectively do not contribute to the error measure and are neglected in the training process (very much similar to support vectors). It was shown theoretically and experimentally that the cumulative margin distribution of the training patterns converges asymptotically to a step. Therefore, original ADABOOST achieves a *hard margin* classification asymptotically. The asymptotic margin distribution of ADABOOST and SVM (for the separable case) are very similar. Hence, the patterns lying in the step part (support patterns) of the margin distribution show a large overlap to the support vectors found by a SVM.

We discussed in detail that ATAs and hard margin classifiers are in general noise sensitive and prone to overfit. We introduced three regularization-based ADABOOST algorithms to alleviate this overfitting problem: (1) direct incorporation of the regularization term into the error function (ADABOOST$_{\text{REG}}$), use of (2) linear and (3) quadratic programming with slack variables to improve existing ensembles. The essence of our proposed algorithms is to achieve a soft margin (through regularization term and slack variables) in contrast to the hard margin classification used before. The soft-margin approach allows to control how much we trust the data, so we are permitted to ignore noisy patterns (e.g. outliers) which would otherwise spoile our classification. This generalization is very much in the spirit of SUPPORT VECTOR MACHINES that also tradeoff the maximization of the margin and the minimization of the classification errors by introducing slack variables. Note that we just gave one definition for the soft margin in ADABOOST$_{\text{REG}}$ other extensions that e.g. use regularization operators (e.g. Smola et al., 1998; Rokui & Shimodaira, 1998; Bishop, 1995) or that have other functional forms (cf. Rätsch et al., 2000) are also possible.

In our experiments on noisy data the proposed regularized versions of ADABOOST: ADABOOST$_{\text{REG}}$ and L/QP$_{\text{REG}}$-ADABOOST show a more robust behavior than the original ADABOOST algorithm. Furthermore, ADABOOST$_{\text{REG}}$ exhibits a better overall generalization performance than all other analyzed algorithms including the SUPPORT VECTOR MACHINES. We conjecture that this result is mostly due to the fact that SUPPORT VECTOR MACHINES can only use one σ, i.e. only one–fixed–kernel, and therefore loses multi-scaling information. ADABOOST does not have this limitation, since we use RBF nets with adaptive kernel widths as base hypotheses.

Our future work will concentrate on a continuing improvement of ADABOOST-type algorithms for noisy real world applications. Also, a further analysis of the relation between ADABOOST (in particular QP$_{\text{REG}}$-ADABOOST) and SUPPORT VECTOR MACHINES from the margin point of view seems promising, with particular focus on the question of what good margin distributions should look like. Moreover, it is interesting to see how the techniques established in this work can be applied to ADABOOST in a regression scenario (cf. Bertoni, Campadelli, & Parodi, 1997; Friedman, 1999; Rätsch et al., 2000).

Appendix

A. Proof of Lemma 1

Proof: We define $\pi_t(\mathbf{z}_i) := \prod_{r=1}^{t} \exp(-b_r \mathrm{I}(h_r(\mathbf{z}_i) = y_i))$ and from definition of G and d we get

$$\begin{aligned}\frac{\frac{\partial G}{\partial \rho(\mathbf{z}_i, \mathbf{b}^t)}}{\sum_{j=1}^{l} \frac{\partial G}{\partial \rho(\mathbf{z}_j, \mathbf{b}^t)}} &= \frac{\exp\left(-\frac{1}{2}\rho(\mathbf{z}_i, \mathbf{b}^t)\right)}{\sum_{j=1}^{l} \exp\left(-\frac{1}{2}\rho(\mathbf{z}_i, \mathbf{b}^t)\right)} \\ &= \frac{\pi_t(\mathbf{z}_i)}{\sum_{j=1}^{l} \pi_t(\mathbf{x}_j)} \\ &= \frac{\pi_t(\mathbf{z}_i)}{\tilde{Z}_t},\end{aligned}$$

where $\tilde{Z}_t := \sum_{i=1}^{l} \pi_t(\mathbf{z}_i)$. By definition, $\pi_t(\mathbf{z}_i) = \pi_{t-1}(\mathbf{z}_i) \exp(-b_t \mathrm{I}(h_t(\mathbf{z}_i) = y_i))$ and $\pi_1(\mathbf{z}_i) = 1/l$. Thus, we get

$$\begin{aligned}w_{t+1}(\mathbf{z}_i) = \frac{\pi_t(\mathbf{z}_i)}{\tilde{Z}_t} &= \frac{\pi_{t-1}(\mathbf{z}_i) \exp(-b_t \mathrm{I}(h_t(\mathbf{z}_i) = y_i))}{\tilde{Z}_t} \\ &= \frac{w_{t-1}(\mathbf{z}_i) \tilde{Z}_{t-1} \exp(-b_t \mathrm{I}(h_t(\mathbf{z}_i) = y_i))}{\tilde{Z}_t} \\ &= \frac{w_{t-1}(\mathbf{z}_i) \exp(-b_t \mathrm{I}(h_t(\mathbf{z}_i) = y_i))}{Z_t},\end{aligned}$$

where $Z_t = \tilde{Z}_t \tilde{Z}_{t-1}$ (cf. step 4 in figure 1). □

B. Proof of Theorem 3

The proof follows the one of Theorem 5 in (Schapire et al., 1997). Theorem 3 is a generalization for $\phi \neq \frac{1}{2}$.

Proof: If $yf(x) \leq \theta$, then we have

$$y \sum_{t=1}^{T} b_t h_t(x) \leq \theta \sum_{t=1}^{T} b_t,$$

and also

$$\exp\left\{-\frac{y}{2}\sum_{t=1}^{T} b_t h_t(x) + \frac{\theta}{2}\sum_{t=1}^{T} b_t\right\} \geq 1.$$

Thus,

$$\begin{aligned}\mathbf{P}_{(\mathbf{x},y)\sim\mathbf{Z}}[yf(\mathbf{x}) \leq \theta] &\leq \frac{1}{l}\sum_{i=1}^{l} \exp\left\{-\frac{y_i}{2}\sum_{t=1}^{T} b_t h_t(\mathbf{x}_i) + \frac{\theta}{2}\sum_{t=1}^{T} b_t\right\} \\ &= \frac{\exp\left(\frac{\theta}{2}\sum_{t=1}^{T} b_t\right)}{l}\sum_{i=1}^{l} \exp\left\{-\frac{y_i}{2}\sum_{t=1}^{T} b_t h_t(\mathbf{x}_i)\right\},\end{aligned}$$

where

$$\begin{aligned}&\sum_{i=1}^{l} \exp\left\{-\frac{y_i}{2}\sum_{t=1}^{T} b_t h_t(\mathbf{x}_i)\right\} \\ &\quad= \sum_{i=1}^{l} \exp\left\{-\frac{y_i}{2}\sum_{t=1}^{T-1} b_t h_t(\mathbf{x}_i)\right\}\exp\left\{-\frac{y_i}{2} b_T h_T(\mathbf{x}_i)\right\} \\ &\quad= \sum_{i:h_T(\mathbf{x}_i)=y_i} \exp\left\{-\frac{y_i}{2}\sum_{t=1}^{T-1} b_t h_t(\mathbf{x}_i)\right\} e^{-b_T/2} \\ &\qquad+ \sum_{i:h_T(\mathbf{x}_i)\neq y_i} \exp\left\{-\frac{y_i}{2}\sum_{t=1}^{T-1} b_t h_t(\mathbf{x}_i)\right\} e^{b_T/2} \\ &\quad= \left(\sum_{i=1}^{l} \exp\left\{-\frac{y_i}{2}\sum_{t=1}^{T-1} b_t h_t(\mathbf{x}_i)\right\}\right)((1-\epsilon_T)e^{-b_T/2} + \epsilon_T e^{b_T/2}),\end{aligned}$$

because

$$\epsilon_T = \frac{1}{\sum_{j=1}^{l} w_j^T} \sum_{i:h_T(\mathbf{x}_i)\neq y_i} w_i^T.$$

With $\sum_{i=1}^{l} \exp(0) = l$ (for $t = 1$), we get recursively

$$\mathbf{P}_{(\mathbf{x},y)\sim\mathbf{Z}}[yf(\mathbf{x}) \le \theta] \le \exp\left(\frac{\theta}{2}\sum_{t=1}^{T} b_t\right)\prod_{t=1}^{T}((1-\epsilon_t)e^{-b_t/2} + \epsilon_t e^{b_t/2}).$$

Plugging in the definition for b_t we get

$$\begin{aligned}
\mathbf{P}_{(\mathbf{x},y)\sim\mathbf{Z}}[yf(\mathbf{x}) \le \theta] &\le \left(\prod_{t=1}^{T}\frac{1-\epsilon_t}{\epsilon_t}\prod_{t=1}^{T}\frac{\phi}{1-\phi}\right)^{\theta/2} \\
&\quad \times \left(\sqrt{\frac{\phi}{1-\phi}} + \sqrt{\frac{1-\phi}{\phi}}\right)^{T}\prod_{t=1}^{T}\sqrt{(1-\epsilon_t)\,\epsilon_t} \\
&= \left(\left(\frac{\phi}{1-\phi}\right)^{\frac{1+\phi}{2}} + \left(\frac{1-\phi}{\phi}\right)^{\frac{1-\phi}{2}}\right)^{T}\prod_{t=1}^{T}\sqrt{(1-\epsilon_t)^{1+\theta}\epsilon_t^{1-\theta}} \\
&= \left(\varphi^{\frac{1+\theta}{2}} + \varphi^{-\frac{1-\theta}{2}}\right)^{T}\prod_{t=1}^{T}\sqrt{\epsilon_t^{1-\theta}(1-\epsilon_t)^{1+\theta}}.
\end{aligned}$$

□

C. Proof of Lemma 5

Proof: We have to show that $\lim_{t\to\infty}\varepsilon_t = 0$, where

$$\varepsilon_t := \left|\min_{i:y_i=1}\rho(\mathbf{z}_i, \mathbf{c}^t) - \min_{j:y_j=-1}\rho(\mathbf{z}_j, \mathbf{c}^t)\right|.$$

The set S_c^t is the set of support patterns at iteration t:

$$S_c^t = \left\{j \in \{1,\ldots,l\} : \rho(\mathbf{z}_j, \mathbf{c}^t) = \min_{i:y_i=c}\rho(\mathbf{z}_i, \mathbf{c}^t)\right\},$$

which clearly contains at least one element. Note that $S_1^\infty \cup S_{-1}^\infty$ is the set of support patterns, which we will get asymptotically.

Suppose we have $\varepsilon_t > \Delta/2 > 0$ and $s \in \{\pm 1\}$ is the class with the smallest margin. With (4), $q \in S_s^t$ and $Q := \exp(-\frac{1}{2}\rho(\mathbf{z}_q, \mathbf{c}^t))$ we get

$$\begin{aligned}
\sum_{i:y_i=s} w_t(\mathbf{z}_i) &\ge \frac{Q^{|\mathbf{b}^t|}}{Q^{|\mathbf{b}^t|} + \sum_{i:y_i\neq s}\exp\left(-\frac{1}{2}\rho(\mathbf{z}_i, \mathbf{c}^t)\right)^{|\mathbf{b}^t|}} \\
&> \frac{Q^{|\mathbf{b}^t|}}{Q^{|\mathbf{b}^t|} + (l-1)Q^{|\mathbf{b}^t|}\mathrm{e}^{-|\mathbf{b}^t|\Delta/4}}
\end{aligned}$$

With $|\mathbf{b}^t| > t\gamma$ from assumption (8) we get $\sum_{i:y_i=s} w_t(\mathbf{z}_i) \geq \delta$, for $t > t_1 := \frac{\log(l-1)-\log(1/\delta-1)}{\gamma\Delta/4}$. Hence, with assumption (9) we get: If $t \geq t_1$, then all patterns $\mathbf{z}_i$ of class s will be classified correctly and patterns of class $r = -s$ will be misclassified. Therefore, we obtain $\rho(\mathbf{z}_i, \mathbf{c}^{t+1}) = \frac{|\mathbf{b}^t|\rho(\mathbf{z}_i,\mathbf{c}^t)+b_{t+1}}{|\mathbf{b}^{t+1}|}$, for $i \in \{k : y_k = s\}$ (especially for $i \in S_s^t$). The patterns of class r will be misclassified and we have $\rho(\mathbf{z}_j, \mathbf{c}^{t+1}) = \frac{|\mathbf{b}^t|\rho(\mathbf{z}_j,\mathbf{c}^t)-b_{t+1}}{|\mathbf{b}^{t+1}|}$, for $j \in \{k : y_k = r\}$. It follows

$$\begin{aligned}\varepsilon_{t+1} &= \left| \min_{i:y_i=s} \frac{|\mathbf{b}^t|\rho(\mathbf{z}_i, \mathbf{c}^t) - b_{t+1}}{|\mathbf{b}^{t+1}|} - \min_{j:y_j=r} \frac{|\mathbf{b}^t|\rho(\mathbf{z}_j, \mathbf{c}^t) + b_{t+1}}{|\mathbf{b}^{t+1}|} \right| \\ &= \left| \frac{\varepsilon_t|\mathbf{b}^t| - 2b_{t+1}}{|\mathbf{b}^{t+1}|} \right|\end{aligned}$$

As long as $\varepsilon_t > \Delta/2$, all patterns of class r will be misclassified and patterns of class s will be classified correctly. If it becomes $(\varepsilon_t|\mathbf{b}^t| - 2b_{t+1})/|\mathbf{b}^{t+1}| < -\Delta/2$, then the same reasoning can be done interchanging the role of s and r. If furthermore $t > t_2 := \frac{4\Gamma}{\gamma\Delta}$, then $\varepsilon_t|\mathbf{b}^t| - 2b_{t+1} > 0$ and we have $\varepsilon_{t+1} < \varepsilon_t - \omega_t$, where ω_t is decreasing not too fast (i.e. it can be bounded by $\mathcal{O}(1/t)$). Hence, we will reach the case $\varepsilon_t < \Delta/2$ and get: If t is large enough ($t > t_3 := \frac{2\Gamma(2-\Delta)}{\Delta\gamma}$), then

$$-\Delta < \frac{\varepsilon_t|\mathbf{b}^t| - 2b_{t+1}}{|\mathbf{b}^{t+1}|} < \Delta,$$

i.e. adding a new hypothesis will not lead to $\varepsilon_{t+1} \geq \Delta$.

Therefore, after a finite number $\tilde{t}$ of subsequent steps, we have we will reach $\varepsilon_t < \Delta$. Furthermore, the discussion above shows that, if t is large enough, it is not possible to leave the Δ area around 0. Hence, for each $\Delta > 0$, we can provide an index $T = \max(t_1, t_2, t_3) + \tilde{t}$ (where $t_1, \ldots, t_3$ are the lower bounds on t used above), such that $\varepsilon_t < \Delta$ for all $t > T$. This implies the desired result. □

D. RBF nets with adaptive centers

The RBF nets used in the experiments are an extension of the method of Moody and Darken (1989), since centers and variances are also adapted (see also Bishop, 1995; Müller et al., 1998). The output of the network is computed as a linear superposition of K basis functions

$$f(\mathbf{x}) = \sum_{k=1}^{K} w_k g_k(\mathbf{x}), \tag{24}$$

where $w_k, k = 1, \ldots, K$, denotes the weights of the output layer. The Gaussian basis functions g_k are defined as

$$g_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mu_k\|^2}{2\sigma_k^2}\right), \tag{25}$$

where μ_k and σ_k^2 denote means and variances, respectively. In a first step, the means μ_k are initialized with K-means clustering and the variances σ_k are determined as the distance between μ_k and the closest $\mu_i (i \neq k, i \in \{1, \ldots, K\})$. Then in the following steps we perform a gradient descent in the regularized error function (weight decay)

$$E = \frac{1}{2} \sum_{i=1}^{l} (y_i - f(\mathbf{x}_i))^2 + \frac{\lambda}{2l} \sum_{k=1}^{K} w_k^2. \tag{26}$$

Taking the derivative of Eq. (26) with respect to RBF means μ_k and variances σ_k we obtain

$$\frac{\partial E}{\partial \mu_k} = \sum_{i=1}^{l} (f(\mathbf{x}_i) - y_i) \frac{\partial}{\partial \mu_k} f(\mathbf{x}_i), \tag{27}$$

with $\frac{\partial}{\partial \mu_k} f(\mathbf{x}_i) = w_k \frac{\mathbf{x}_i - \mu_k}{\sigma_k^2} g_k(\mathbf{x}_i)$ and

$$\frac{\partial E}{\partial \sigma_k} = \sum_{i=1}^{l} (f(\mathbf{x}_i) - y_i) \frac{\partial}{\partial \sigma_k} f(\mathbf{x}_i), \tag{28}$$

with $\frac{\partial}{\partial \sigma_k} f(\mathbf{x}_i) = w_k \frac{\|\mu_k - \mathbf{x}_i\|^2}{\sigma_k^3} g_k(\mathbf{x}_i)$. These two derivatives are employed in the minimization of Eq. (26) by a conjugate gradient descent with line search, where we always compute the optimal output weights in every evaluation of the error function during the line search. The optimal output weights $\mathbf{w} = [w_1, \ldots, w_K]^\top$ in matrix notation can be computed in closed form by

$$\mathbf{w} = \left(G^T G + 2 \frac{\lambda}{l} \mathbf{I} \right)^{-1} G^T \mathbf{y}, \text{ where } G_{ik} = g_k(\mathbf{x}_i) \tag{29}$$

and $\mathbf{y} = [y_1, \ldots, y_l]^\top$ denotes the output vector, and $\mathbf{I}$ an identity matrix. For $\lambda = 0$, this corresponds to the calculation of a pseudo-inverse of G.

So, we simultaneously adjust the output weights and the RBF centers and variances (see figure 15 for pseudo-code of this algorithm). In this way, the network fine-tunes itself to the data after the initial clustering step, yet, of course, overfitting has to be avoided by careful tuning of the regularization parameter, the number of centers K and the number of iterations (cf. Bishop, 1995). In our experiments we always used $\lambda = 10^{-6}$ and up to ten CG iterations.

Acknowledgments

We thank for valuable discussions with B. Schölkopf, A. Smola, T. Frieß, D. Schuurmans and B. Williamson. Partial funding from EC STORM project number 25387 is gratefully acknowledged. Furthermore, we acknowledge the referees for valuable comments.

Notes

1. An ensemble is a collection of neural networks or other types of classifiers (hypotheses) that are trained for the same task.
2. In Friedman et al. (1998) it was mentioned that sometimes the randomized version shows a better performance, than the version with weighted minimization. In connection with the discussion in Section 3 this becomes clearer, because the randomized version will show an overfitting effect (possibly much) later and overfitting may be not observed, whereas it was observed using the more efficient weighted minimization.
3. In our experiments we have often observed values for $|\mathbf{b}|$ that are larger than 100 after only 200 iterations.
4. A detailed description of the generation of the toy data used in the asymptotical simulations can be found in the Internet http://ida.first.gmd.de/~raetsch/data/banana.html.
5. The definition of the influence clearly depends on the base hypothesis space $\mathcal{H}$. If the hypothesis space changes, other patterns may be more difficult to classify.
6. Note that for $p = 1$ there is a connection to the leave-one-out-SVM approach of Weston (1999).
7. Interestingly, the (soft) SVM generates many more support vectors in the high noise case than in the low noise case (Vapnik, 1995).
8. From the algorithm in figure 12, it is straight forward to get the following linear program, which is equivalent for any fixed $S > 0$:

$$\text{Minimize } \frac{\varrho^+}{S} + \frac{C}{S}\sum_i \xi_i^+ \text{subject to } S\sum_{t=1}^{T} c_t M_{i,t} \geq \varrho^+ - \xi_i^+,$$

$$\text{where } \varrho^+ := S\varrho, \xi_i^+ := S\xi_i, b_t \geq 0, \xi_i^+ \geq 0, \sum_t b_t = S.$$

   In this problem we can set ϱ^+ to 1 and try to optimize S. To retrieve a linear program, we use the Taylor expansion around 1: $\frac{1}{S} = S + \mathcal{O}((S-1)^2)$ and $\sum_i \xi_i^+/S = \sum_i \xi_i^+ + \mathcal{O}(S-1)$. For $S = |\mathbf{b}|$ we get algorithm (22).
9. For $p = 2$, $\|\mathbf{b}\|_2$ corresponds to the Renyi entropy of the hypothesis vector and we are effectively trying to minimize this entropy while separating the data.
10. These data sets including a short description, the splits into the 100 realizations and the simulation results are available at http://ida.first.gmd.de/~raetsch/data/benchmarks.htm.
11. The breast cancer domain was obtained from the University Medical Center, Inst. of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data.
12. A random partition generates a mapping $\mathbf{m}$ of n to two classes. For this a random ± 1 vector $\mathbf{m}$ of length n is generated. The positive classes (and the negative respectively) are then concatenated.
13. The parameters selected by the cross validation are only near-optimal. Only 15–25 values for each parameter are tested in two stages: first a global search (i.e. over a wide range of the parameter space) was done to find a good guess of the parameter, which becomes more precise in the second stage.

References

Bennett, K. (1998). Combining support vector and mathematical programming methods for induction. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods—SV learning*. Cambridge, MA: MIT Press.

Bennett, K. & Mangasarian, O. (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software, 1*, 23–34.

Bertoni, A., Campadelli, P., & Parodi, M. (1997). A boosting algorithm for regression. In W. Gerstner, A. Germond, M. Hasler, & J.-D. Nicoud (Eds.), *LNCS, Vol. V: Proceedings ICANN'97: Int. Conf. on Artificial Neural Networks* (pp. 343–348). Berlin: Springer.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Boser, B., Guyon, I., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In D. Haussler (Ed.), *Proceedings COLT'92: Conference on Computational Learning Theory* (pp. 144–152). New York, NY: ACM Press.

Breiman, L. (1996). Bagging predictors. *Mechine Learning, 26*(2), 123–140.

Breiman, L. (1997a). Arcing the edge. Technical Report 486, Statistics Department, University of California.

Breiman, L. (1997b). Prediction games and arcing algorithms. Technical Report 504, Statistics Department, University of California.

Breiman, L. (1998). Arcing classifiers. *The Annals of Statistics, 26*(3), 801–849.

Breiman, L. (1999). Using adaptive bagging to debias regressions. Technical Report 547, Statistics Department, University of California.

Cortes, C. & Vapnik, V. (1995). Support vector networks. *Machine Learning, 20*, 273–297.

Frean, M. & Downs, T. (1998). A simple cost function for boosting. Technical Report, Department of Computer Science and Electrical Engineering, University of Queensland.

Freund, Y. & Schapire, R. (1994). A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings EuroCOLT'94: European Conference on Computational Learning Theory*. LNCS.

Freund, Y. & Schapire, R. (1996). Game theory, on-line prediction and boosting. In *Proceedings COLT'86: Conf. on Comput. Learning Theory* (pp. 325–332). New York, NY: ACM Press.

Friedman, J. (1999). Greedy function approximation. Technical Report, Department of Statistics, Stanford University.

Friedman, J., Hastie, T., & Tibshirani, R. (1998). Additive logistic regression: A statistical view of boosting. Technical Report, Department of Statistics, Sequoia Hall, Stanford University.

Frieß, T. & Harrison, R. (1998). Perceptrons in kernel feature space. Research Report RR-720, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK.

Grove, A. & Schuurmans, D. (1998). Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the Fifteenth National Conference on Artifical Intelligence*.

Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *J. Statistical Physics, 34*, 975–986.

LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Müller, U., Säckinger, E., Simard, P., & Vapnik, V. (1995). Learning algorithms for classification: A comparism on handwritten digit recognition. *Neural Networks*, 261–276.

Mangasarian, O. (1965). Linear and nonlinear separation of patterns by linear programming. *Operations Research, 13*, 444–452.

Mason, L., Bartlett, P. L., & Baxter, J. (2000a). Improved generalization through explicit optimization of margins. *Machine Learning 38*(3), 243–255.

Mason, L., Baxter, J., Bartlett, P. L., & Frean, M. (2000b). Functional gradient techniques for combining hypotheses. In A. J. Smola, P. Bartlett, B. Schölkopf, & C. Schuurmans (Eds.), *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press.

Moody, J. & Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation, 1*(2), 281–294.

Müller, K.-R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., & Vapnik, V. (1998). Using support vector machines for time series prediction. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in Kernel Methods—Support Vector Learning*. Cambridge, MA: MIT Press.

Onoda, T., Rätsch, G., & Müller, K.-R. (1998). An asymptotic analysis of ADABOOST in the binary classification case. In L. Niklasson, M. Bodén, & T. Ziemke (Eds.), *Proceedings ICANN'98: Int. Conf. on Artificial Neural Networks* (pp. 195–200).

Onoda, T., Rätsch, G., & Müller, K.-R. (2000). An asymptotical analysis and improvement of ADABOOST in the binary classification case. *Journal of Japanese Society for AI, 15*(2), 287–296 (in Japanese).

Press, W., Flannery, B., Teukolsky, S., & Vetterling, W. (1992). *Numerical Recipes in C* (2nd ed.). Cambridge: Cambridge University Press.

Quinlan, J. (1992). *C4.5: Programs for Machine Learning*. Los Altos, CA: Morgan Kaufmann.

Quinlan, J. (1996). Boosting first-order learning. In S. Arikawa & A. Sharma (Eds.), *LNAI, Vol. 1160: Proceedings of the 7th International Workshop on Algorithmic Learning Theory* (pp. 143–155). Berlin: Springer.

Rätsch, G. (1998). Ensemble learning methods for classification. Master's Thesis, Department of Computer Science, University of Potsdam, Germany (in German).

Rätsch, G., Onoda, T., & Müller, K.-R. (1998). Soft margins for ADABOOST. Technical Report NC-TR-1998-021, Department of Computer Science, Royal Holloway, University of London, Egham, UK.

Rätsch, G., Onoda, T., & Müller, K.-R. (1999). Regularizing ADABOOST. In M. Kearns, S. Solla, & D. Cohn (Eds.), *Advances in Neural Information Processing Systems 11* (pp. 564–570). Cambridge, MA: MIT Press.

Rätsch, G., Schölkopf, B., Smola, A., Mika, S., Onoda, T., & Müller, K.-R. (2000). Robust ensemble learning. In A. Smola, P. Bartlett, B. Schölkopf, & D. Schuurmans (Eds.), *Advances in Large Margin Classifiers* (pp. 207–219). Cambridge, MA: MIT Press.

Rätsch, G., Warmuth, M., Mika, S., Onoda, T., Lemm, S., & Müller, K.-R. (2000). Barrier boosting. In *Proceedings COLT'00: Conference on Computational Learning Theory* (pp. 170–179). Los Altos, CA: Morgan Kaufmann.

Rokui, J. & Shimodaira, H. (1998). Improving the generalization performance of the minimum classification error learning and its application to neural networks. In *Proc. of the Int. Conf. on Neural Information Processing (ICONIP)* (pp. 63–66). Japan, Kitakyushu.

Schapire, R. (1999). Theoretical views of boosting. In *Proceedings EuroCOLT'99: European Conference on Computational Learning Theory*.

Schapire, R., Freund, Y., Bartlett, P., & Lee, W. (1997). Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings ICML'97: International Conference on Machine Learning* (pp. 322–330). Los Altos, CA: Morgan Kaufmann.

Schapire, R. & Singer, Y. (1998). Improved boosting algorithms using confidence-rated predictions. In *Proceedings COLT'98: Conference on Computational Learning Theory* (pp. 80–91).

Schölkopf, B. (1997). *Support Vector Learning*. R. Oldenbourg Verlag, Berlin.

Schölkopf, B., Smola, A., & Williamson, R. (2000). New support vector algorithms. *Neural Computation*. also NeuroCOLT TR–31–89, 12:1083–1121.

Schwenk, H. & Bengio, Y. (1997). AdaBoosting neural networks. In W. Gerstner, A. Germond, M. Hasler, & J.-D. Nicoud (Eds.), *Proceedings ICANN'97: Int. Conf. on Artificial Neural Networks, Vol. 1327 of LNCS* (pp. 967–972). Berlin: Springer.

Smola, A. J. (1998). Learning with kernels. Ph.D. Thesis, Technische Universität Berlin.

Smola, A., Schölkopf, B., & Müller, K.-R. (1998). The connection between regularization operators and support vector kernels. *Neural Networks, 11*, 637–649.

Tikhonov, A. & Arsenin, V. (1977). *Solutions of Ill-Posed Problems*. Washington, D.C.: W.H. Winston.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Berlin: Springer.

Weston, J. (1999). LOO-support vector machines. In *Proceedings of IJCNN'99*.

Weston, J., Gammerman, A., Stitson, M. O., Vapnik, V., Vovk, V., & Watkins, C. (1997). Density estimation using SV machines. Technical Report CSD-TR-97-23, Royal Holloway, University of London, Egham, UK.

Received August 14, 1998
Revised June 14, 1999
Accepted October 27, 1999
Final manuscript September 18, 2000