Maximizing the Margin with Boosting*

Gunnar Rätsch[1] and Manfred K. Warmuth[2]

[1] RSISE, Australian National University
Canberra, ACT 0200, Australia
Gunnar.Raetsch@anu.edu.au
[2] University of California at Santa Cruz
Santa Cruz, CA 95060, USA
manfred@cse.ucsc.edu

Abstract. AdaBoost produces a linear combination of weak hypotheses. It has been observed that the generalization error of the algorithm continues to improve even after all examples are classified correctly by the current linear combination, i.e. by a *hyperplane* in feature space spanned by the weak hypotheses. The improvement is attributed to the experimental observation that the distances (margins) of the examples to the separating hyperplane are increasing even when the training error is already zero, that is all examples are on the correct side of the hyperplane. We give an iterative version of AdaBoost that explicitly maximizes the minimum margin of the examples. We bound the number of iterations and the number of hypotheses used in the final linear combination which approximates the maximum margin hyperplane with a certain precision. Our modified algorithm essentially retains the exponential convergence properties of AdaBoost and our result does not depend on the size of the hypothesis class.

1 Introduction

In the most common version of boosting the algorithm is given a fixed set of labeled training examples. In each stage the algorithm produces a probability weighting on the examples. It then is given a *weak* hypothesis whose error (probability of wrong classification) is slightly below 50%. The weak hypothesis is used to update the distribution. Intuitively, the hard examples receive high weights. At the end of each stage the weak hypothesis is added to the linear combination, which forms the current hypothesis of the boosting algorithm.

The most well known boosting algorithm is AdaBoost [5]. It is "adaptive" in that the linear coefficient of the weak hypothesis depends on error of the weak hypothesis. Earlier work on boosting includes [18, 3]. AdaBoost has two interesting properties. First, along with earlier boosting algorithms [18], it has the property that its training

* This work was done while G. Rätsch was at Fraunhofer FIRST Berlin and at UC Santa Cruz. G. Rätsch was partially funded by DFG under contract JA 379/91, JA 379/71, MU 987/1-1 and by EU in the NeuroColt II project. M.K. Warmuth and visits of G. Rätsch to UC Santa Cruz were partially funded by the NSF grant CCR-9821087. G. Rätsch thanks S. Mika, S. Lemm and K.-R. Müller for discussions.

error converges exponentially fast to zero. More precisely, if the training error of the t-th weak learner is $\epsilon_t = \frac{1}{2} - \frac{1}{2}\gamma_t$, then an upper bound on the training error of the linear combination is reduced by a factor of $1 - \frac{1}{2}\gamma_t^2$ at stage t. Second, it has been observed experimentally that AdaBoost continues to "learn" even after the training error of the linear combination is zero [19]; in experiments the generalization error is continuing to improve. When the training error is zero, then all examples are on the "right side" of the linear combination (viewed as a *hyperplane* in a feature space, where each base hypothesis is one feature/dimension). The *margin* of an example is the signed distance to the hyperplane times its $\pm$ label. As soon as the training error is zero, the examples are on the right side and have positive margin. It has also been observed that the margins of the examples continue to increase even after the training error is zero. There are theoretical bounds on the generalization error of linear classifiers [e.g. 19, 2, 10] that improve with the size of the minimum margin of the examples. So the fact that the margins improve experimentally seems to explain why AdaBoost still learns after the training error is zero.

There is one shortfall in this argument. AdaBoost has not been proven to maximize the minimum margin of the examples. In fact, in our experiments in Section 4 we observe that AdaBoost does not seem to maximize the margin. [2] proposed a modified algorithm – Arc-GV (**Arc**ing-**G**ame **V**alue) – suitable for this task and showed that it *asymptotically* maximizes the margin (similar results are shown in [7] and [1]). In this paper we give an algorithm that produces a final hypothesis whose margin lies in $[\varrho^* - \nu, \varrho^*]$, where ϱ^* is the maximum margin and ν a precision parameter. The final stage of the algorithm produces a linear combination of at most $2 \log N/\nu^2$ weak learners. The algorithm requires a guess of the margin in the interval $[\varrho^* - \nu, \varrho^*]$ and up $\log_2(2/\nu)$ stages are needed before finding a guess in that interval.

To our knowlegde, this is the first result on the fast convergence of a boosting algorithm to the maximum margin solution that works for all $\varrho^* \in [-1, 1]$. Using previous results one can only show that AdaBoost *assymptically* converges to a final hypothesis with margin at least $\varrho^*/2$ (cf. Corollary 5) if $\varrho^* > 0$ and if other subtle conditions on the base learner are satisfied.

AdaBoost was designed to find a final hypothesis of margin at least zero. Our algorithm maximizes the margin for all values of ϱ^*. This includes the inseparable case ($\varrho^* < 0$), where one minimizes the overlap between the two classes. Our algorithm is also useful when the hypotheses given to the Boosting algorithm are *strong* in the sense that they already separate the data and have margin greater than zero, but less than one. In this case $0 < \varrho^* < 1$ and AdaBoost aborts immediately because the linear coefficients for such hypotheses are unbounded. Our new algorithm also maximizes the margin by combining strong learners.

The paper is structured as follows: In Section 2 we first extend the original AdaBoost algorithm leading to *AdaBoost*$_\varrho$. Then we propose *Marginal AdaBoost*, which uses AdaBoost$_\varrho$ as a subroutine. In Section 3 we give a more detailed analysis of both algorithms. First, we prove that if the training error of the t-th weak learner is $\epsilon_t = \frac{1}{2} - \frac{1}{2}\gamma_t$, then an upper bound on the fraction of examples with margin smaller than ϱ is reduced by a factor of $1 - \frac{1}{2}(\varrho - \gamma_t)^2$ at stage t of AdaBoost$_\varrho$ (cf. Section 3.2). To achieve a large margin, however, one needs to assume that the guess ϱ is at most ϱ^*.

Exploiting this property, we prove the exponential convergence rate of our algorithm (cf. Theorem 6) if $\varrho \leq \rho^*$. A good guess at most ν below ϱ^* can be found in essentially $\log_2(2/\nu)$ stages. We complete the paper with experiments confirming our theoretical analysis (Section 4) and a conclusion.

2 Marginal Boosting

Boosting algorithms form linear combinations of weak hypothesis. In iteration t the weak hypothesis $h_t(\mathbf{x})$ is added to the linear combination. For AdaBoost it has been shown it generates a combined hypothesis

$$f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_t \frac{\alpha_t}{\sum_r \alpha_r} h_t(\mathbf{x})$$

that is consistent with the training set in a small number of iterations [5]. This is desirable for the analysis in the PAC setting [18, 21]. Consistency on the training set means that the margin[1] of each training example is greater than zero, i.e. $\min_{n=1,\ldots,N} y_n f(\mathbf{x}_n) > 0$.

We start with a slight modification of AdaBoost, which does not only aim to find a hypothesis with margin of at least zero, but with at least margin ϱ, where ϱ is prespecified (cf. Algorithm 1).[2] We call this algorithm AdaBoost$_\varrho$, as it naturally gen-

Algorithm 1 – The AdaBoost$_\varrho$ algorithm.

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, No. of Iterations T, margin target ϱ
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots, T$,
   (a) Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1, 1]$
   (b) Calculate the edge γ_t of h_t: $\gamma_t = \sum_{n=1}^N d_n^t y_n h_t(\mathbf{x}_n)$.
   (c) **if** $\gamma_t \leq \varrho$ or $\gamma_t = 1$, **then** break
   (d) Set $\alpha_t = \frac{1}{2} \log \frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2} \log \frac{1+\varrho}{1-\varrho}$.
   (e) Update weights: $d_n^{t+1} = d_n^t \exp\left\{-\alpha_t y_n h_t(\mathbf{x}_n)\right\} / Z_t$, s.t. $\sum_{n=1}^N d_n^{t+1} = 1$.
4. **Output:** $f(\mathbf{x}) = \sum_{t=1}^T \frac{\alpha_t}{\sum_r \alpha_r} h_t(\mathbf{x})$, $\gamma = \min_{t=1,\ldots,T} \gamma_t$, and $\rho = \max_{n=1,\ldots,N} y_n f(\mathbf{x}_n)$

eralizes AdaBoost for the case when the *target margin* is ϱ. The original AdaBoost algorithm now becomes $AdaBoost_0$, as it targets $\varrho = 0$.

[1] If $\|\boldsymbol{\alpha}\|_1 = 1$, then a result by [11] shows that the dot product $\mathbf{x} \cdot \boldsymbol{\alpha}$ is equal to the signed distance of the point $\mathbf{x}$ to the closest point on the plane with normal vector $\boldsymbol{\alpha}$ through the origin, where the distance is measured with the ℓ_∞-norm. This explains why we can use $y_n f(\mathbf{x}_n)$ instead of the ℓ_∞-norm distance to the separating hyperplane.

[2] The original AdaBoost algorithm was formulated in terms of weighted training error ϵ_t of a weak hypothesis. Here we use an equivalent formulation in terms of the edge γ, where $\epsilon_t = \frac{1}{2} - \frac{1}{2}\gamma_t$ (cf. Section 3.1), since it is more convenient [2].

The algorithm AdaBoost$_\varrho$ is already known as *unnormalized Arcing* [2] or *AdaBoost-type Algorithm* [17]. Moreover, it is related to a algorithms proposed in [6] and [23]. The only difference to AdaBoost is the choice of the hypothesis coefficients α_t: An additional term $-\frac{1}{2}\log\frac{1+\varrho}{1-\varrho}$ appears in the expression for α_t. This term is zero for AdaBoost$_0$. The constant ϱ might be seen as a *guess* of the maximum margin ϱ^*. If ϱ is chosen properly (slightly below ϱ^*), then AdaBoost$_\varrho$ will converge exponentially fast to a combined hypothesis with a near maximum margin. See Section 3.2 for details.

The following example illustrates how AdaBoost$_\varrho$ works. Assume the base learner returns the constant hypothesis $h_t(\mathbf{x}) \equiv 1$. The error of this hypothesis is the sum of all negative weights, i.e. $\epsilon_t = \sum_{y_n=-1} d_n^t$, its edge is $\gamma_t = 1 - 2\epsilon_t$. If $\gamma_t > \rho$, then the new distribution $\mathbf{d}^{t+1}$ is computed (otherwise the algorithm stops). The parameter α_t is chosen so that the edge of h_t with respect to the new distribution is exactly ϱ (instead of 0 as in AdaBoost). Actually, the edge is ϱ for any ± 1-valued base hypotheses or more generally, if one chooses α_t s.t. $Z_t(\alpha_t)$ is minimized. In [9] the standard boosting algorithms are interpreted as approximate solutions to the following optimization problem: choose distribution $\mathbf{d}$ subject to the constraints that the edges of the previous hypotheses are *equal* zero. In this paper we use the *inequality* constraints that the edges of the previous hypotheses are at most ϱ. The α_t's are the Lagrange multipliers for these inequality constraints, which are always positive.

Since one does not know the value of ϱ^* beforehand, one also needs to find ϱ^*. We propose an algorithm that constructs a sequence $\{\varrho_r\}_{r=1}^R$ converging to ϱ^*: A fast way to find a real value up to a certain accuracy ν on the interval $[-1, 1]$ is to use a *binary search* – one needs only $\log_2(2/\nu)$ search steps.[3] AdaBoost$_{\varrho_r}$ (Algorithm 1) is used to decide whether the current guess ϱ is *larger* or *smaller* than ϱ^*. This leads to the *Marginal AdaBoost* algorithm (cf. Algorithm 2).

The algorithm proceeds in R steps, where R is determined by the accuracy ν that we would like to reach: In each iteration r it calls AdaBoost$_{\varrho_r}$ (cf. step 3b in Marginal AdaBoost), where ϱ_r is chosen to be in the middle of an interval $[l_r, u_r]$ (cf. step 3a). Based on the success of AdaBoost$_{\varrho_r}$ to achieve a large enough margin, the interval is updated (cf. step 3c). We can show that the interval is chosen such that it always contains ϱ^*, the *unknown* maximal margin, while the length of the interval is almost reduced by a factor of two. Finally, in the last step of the algorithm, one has reached a good estimate ϱ_R of ϱ^* and calls AdaBoost$_\varrho$ for $\varrho = l_{R+1} - \nu$ generating a combined hypothesis with margin at least $\varrho^* - 4\nu$ (as we will show later). In the next section we will give a detailed analysis of how the algorithm works and how many calls to the base learner are needed to approximate ϱ^*: in the worst case one needs about $\log_2(2/\nu)$ times more iterations than for the case when ϱ^* is known.

Since AdaBoost$_\varrho$ is used as a sub-routine and starts from the beginning in each round, one can think of several speed-ups, which might help to reduce the computation time. For instance, one could store the base hypotheses of previous iterations and instead of calling the base learner, one first sifts through these previously used hypotheses. Furthermore, one may stop AdaBoost$_\varrho$, when the combined hypothesis has reached a margin of ϱ or the base learner returns a hypothesis with edge lower than ϱ.

[3] If one knows that $\varrho^* \in [a, b]$, one needs only $\log_2((b-a)/\nu)$ steps.

Algorithm 2 – The Marginal AdaBoost algorithm

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, Accuracy ν
2. **Initialize:** $\varrho_1 = 0$, $l_1 = -1$, $u_1 = 1$, $R = \lceil \log_2(1/\nu) \rceil$, $T = \lceil 2\log(N)/\nu^2 \rceil + 1$.
3. **Do for** $r = 1, \ldots R$,
   (a) $\varrho_r = \frac{l_r + u_r}{2}$
   (b) $[f_r, \gamma_r, \rho_r] = \mathrm{AdaBoost}_{\varrho_r}(S, T)$
   (c) **if** $\rho_r \geq \varrho_r$, **then** $l_{r+1} = \max(\rho_r, l_r)$, $u_{r+1} = \min(\gamma_r, u_r)$
   else $l_{r+1} = \max(\rho_r, l_r)$, $u_{r+1} = \min(\gamma_r, \varrho_r + \nu, u_r)$
   (d) **if** $u_{r+1} - l_{r+1} \leq 3\nu$, **then** break
4. **Output:** $f = \mathrm{AdaBoost}_{l_{r+1}-\nu}(S, 2\log(N)/\nu^2)$

Constraining the edges of the previous hypotheses to equal zero (as done in the *totally corrective algorithm* of [9]) leads to a problem if there is no solution satisfying these constraints. Because of duality, the maximal edge over all base hypotheses is always greater than the minimal margin of the examples. Thus, if there is a separation with margin greater than 0, then the equality constraints on the edges are not satisfyable. In Marginal AdaBoost we consider a system of *inequality constraints* $\sum_{n=1}^{N} y_n d_n h_t(\mathbf{x}_n) \leq \varrho$, where ϱ is adapted. Again, if there is a separation with margin greater than ϱ, then the system of inequalities has no solution (and the Lagrange multipliers diverge to infinity). However, in this case our algorithm stops and increases ϱ, which increases the size of the feasible set. Finally, if $\varrho = \varrho^*$, then there is presumably only one point left, which is then the dual solution of the maximum margin solution.

3 Detailed Analysis

3.1 Weak learning and margins

The standard assumption made on the base learning algorithm in the PAC-Boosting setting is that it returns a hypothesis h from a fixed set H that is slightly better than random guessing on any training set.[4] More formally this means that the error rate ϵ is consistently smaller than $\frac{1}{2} - \frac{1}{2}\gamma$ for some $\gamma > 0$. Note that the error rate of $\frac{1}{2}$ (i.e. $\gamma = 0$) could easily be reached by a fair coin, assuming both classes have the same prior probabilities. Thus, this requirement is the least we can expect from a learning algorithm [5].

The error rate ϵ is defined as the fraction of points that are misclassified. In Boosting this is extended to weighted example sets and one defines

$$\epsilon_h(\mathbf{d}) = \sum_{n=1}^{N} d_n \, \mathbf{I}(y_n \neq \mathrm{sign}(h(\mathbf{x}_n))),$$

where h is the hypothesis returned by the base learner and $\mathbf{I}$ is the indicator function with $\mathbf{I}(\mathrm{true}) = 1$ and $\mathbf{I}(\mathrm{false}) = 0$. The weighting $\mathbf{d} = [d_1, \ldots, d_N]$ of the examples is such that $d_n \geq 0$ and $\sum_{n=1}^{N} d_n = 1$. A more convenient quantity to measure the

[4] In the PAC setting, the base learner is allowed to fail with probability δ. Since we are seeking for simple presentation, we ignore this fact here. Our algorithm can be extended to this case.

quality of the hypothesis h is the edge $\gamma_h(\mathbf{d}) = \sum_{n=1}^{N} d_n y_n h(\mathbf{x}_n)$, which is an affine transformation of $\epsilon_h(\mathbf{d})$ in the case when $h(\mathbf{x}) \in \{-1,+1\}$: $\epsilon_h(\mathbf{d}) = \frac{1}{2} - \frac{1}{2}\gamma_h(\mathbf{d})$. Recalling from Section 1, the margin of a given example $(\mathbf{x}_n, y_n)$ is defined by $y f_{\boldsymbol{\alpha}}(\mathbf{x})$.

Suppose we would combine all possible hypotheses from H (assuming H is finite), then the following well-known theorem establishes the connection between margins and edges [first seen in connection with Boosting in 4, 2]:

Theorem 1 (Min-Max-Theorem, [22]).

$$\gamma^* = \min_{\mathbf{d}} \max_{h \in H} \sum_{n=1}^{N} d_n y_n h(\mathbf{x}_n) = \max_{\boldsymbol{\alpha}} \min_{n=1,\ldots,N} y_n \sum_{k=1}^{|H|} \alpha_k h_k(\mathbf{x}_n) = \varrho^*, \qquad (1)$$

where $\mathbf{d} \in \mathcal{P}^N$, $\boldsymbol{\alpha} \in \mathcal{P}^{|H|}$ *and* $\mathcal{P}^k$ *is the* k-dimensional probability simplex.

Thus, the minimal edge γ^* that can be achieved over all possible weightings $\mathbf{d}$ of the training set is equal to the maximal margin ϱ^* of a combined hypothesis from H. Also, for any non-optimal weightings $\mathbf{d}$ and $\boldsymbol{\alpha}$ we always have $\max_{h \in H} \gamma_h(\mathbf{d}) \geq \gamma^* = \varrho^* \geq \min_{n=1,\ldots,N} y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)$. If the base learning algorithm guarantees to return a hypothesis with edge at least γ for any weighting, there exists a combined hypothesis with margin at least γ. If $\gamma = \gamma^*$, i.e. the lower bound γ is largest possible, then there exists a combined hypothesis with margin exactly $\gamma = \varrho^*$ (only using hypotheses that are actually returned by the base learner). From this discussion we can derive a sufficient condition on the base learning algorithm to reach the maximal margin: If it returns hypotheses whos edges are at least γ^*, there exists a linear combination of these hypotheses that has margin $\gamma^* = \varrho^*$. This explains the termination condition (cf. step 3c in AdaBoost$_\varrho$). We will prove later that our Marginal AdaBoost algorithm is able to compute the coefficients of the combination efficiently (cf. Theorem 6).

3.2 Convergence properties of AdaBoost$_\varrho$

We now analyze a slightly generalized version of AdaBoost$_\varrho$, where ϱ is not fixed but could be adapted in each iteration. We therefore consider sequences $\{\varrho_t\}_{t=1}^{T}$, which might either be specified before running the algorithm or computed based on results during the algorithm. For instance, the idea proposed by [2] is to compute ϱ_t by $\min_{n=1,\ldots,N} y_n f_{\boldsymbol{\alpha}_{t-1}}(\mathbf{x}_n)$, which leads to Arc-GV *asymptotically* converging to the maximum margin solution.

In the following we are answering the question how good AdaBoost$_{\{\varrho_t\}}$ is able to increase the margin and bound the fraction of examples, which have a margin smaller than say θ. We start with the following useful lemma [similar to a lemma shown in 19, 20]:

Lemma 2. *Assume that in a run of AdaBoost*$_\varrho$ *holds* $\alpha_t \geq 0$ *for all* t. *Then we have*

$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}\left(y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n) \leq \theta\right) \leq \left(\prod_{t=1}^{T} Z_t\right) \exp\left\{\sum_{t=1}^{T} \theta\alpha_t\right\} = \prod_{t=1}^{T} \exp\left\{\theta\alpha_t + \log Z_t\right\} \quad (2)$$

where Z_t *is as in AdaBoost*$_\varrho$, *step* (3e).

This leads to a result generalizing Theorem 5 in [5] for the case $\varrho \neq 0$:

Proposition 3 ([17]). *Let γ_t be the edge of h_t in the t-th step of AdaBoost$_\varrho$. Assume $-1 \leq \varrho_t \leq \gamma_t$. Then holds for all $\theta \in [-1,1]$*

$$\exp\left\{\theta\alpha_t + \log Z_t\right\} \leq \exp\left(-\frac{1+\theta}{2}\log\left(\frac{1+\varrho_t}{1+\gamma_t}\right) - \frac{1-\theta}{2}\log\left(\frac{1-\varrho_t}{1-\gamma_t}\right)\right). \quad (3)$$

The algorithm makes progress, if the right hand side of (3) is smaller than one. Suppose we would like the reach a margin θ on all training examples, where we obviously need to assume $\theta \leq \varrho^*$. Then the question arises, which sequence of $\{\varrho_t\}_{t=1}^T$ one should use to find such combined hypothesis in as few iterations as possible. Since the right hand side (rhs.) of (3) is minimized for $\varrho_t = \theta$, one should always use this choice, independent of how the base learner performs.

We therefore assume for the rest of the paper that ϱ is held constant in a run of AdaBoost$_\varrho$. Let us reconsider (3) of Proposition 3 for the special case $\varrho_t = \varrho \equiv \theta$:

$$\exp\left\{\varrho\alpha_t + \log Z_t\right\} \leq \exp\left\{-\Delta_2(\varrho,\gamma_t)\right\}, \quad (4)$$

where $\Delta_2(\varrho,\gamma_t) := \frac{1+\varrho}{2}\log\frac{1+\varrho}{1+\gamma_t} + \frac{1-\varrho}{2}\log\frac{1-\varrho}{1-\gamma_t}$ is the binary relative entropy. This means that the right hand side of (2) is reduced by a factor exactly $\exp(-\Delta_2(\varrho,\gamma_t))$, which can be bounded from above by using Taylor expansion and the convexity in the second argument of $\exp(-\Delta_2(\cdot,\cdot))$:

$$\exp(-\Delta_2(\varrho,\gamma_t)) \leq 1 - \frac{1}{2}\frac{(\varrho-\gamma_t)^2}{1-\varrho^2} \quad (5)$$

where we need to assume[5] $\gamma_t \geq \varrho$ and $0 \leq \varrho \leq 1$. Note that this is the same form of bound one also obtains for original AdaBoost (i.e. $\varrho = 0$), where the fraction of points with margin smaller than zero is reduced by at least $1 - \frac{1}{2}\gamma_t^2$, if $\gamma_t > 0$. Note that one yields a factor $\frac{1}{1-\varrho^2}$ in (5), which makes the convergence faster, if $\varrho > 0$.

Now we determine an upper bound on the number of iterations needed by AdaBoost$_\varrho$ for achieving a margin of ϱ on all examples, given that the maximum margin is ϱ^*:

Corollary 4. *Assume the base learner always achieves an edge $\gamma_t \geq \varrho^*$. If $0 \leq \varrho \leq \varrho^* - \nu$, $\nu > 0$, then AdaBoost$_\varrho$ will converge to a solution with margin of at least ϱ on all examples in at most $\lceil \frac{2\log(N)(1-\varrho^2)}{\nu^2} \rceil + 1$ steps.*

Proof. We use Lemma 2 and (5) and can bound

$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}\left(y_n f(\mathbf{x}_n) \leq \varrho\right) \leq \prod_{t=1}^{T}\left(1 - \frac{1}{2}\frac{(\varrho-\gamma_t)^2}{1-\varrho^2}\right) \leq \left(1 - \frac{1}{2}\frac{\nu^2}{1-\varrho^2}\right)^T.$$

The margin is at least ϱ for all examples, if the right hand side is smaller than $\frac{1}{N}$, hence after at most $\left\lceil \frac{\log(N)}{-\log\left(1-\frac{1}{2}\frac{\nu^2}{1-\varrho^2}\right)} \right\rceil + 1 \leq \left\lceil \frac{2\log(N)(1-\varrho^2)}{\nu^2} \right\rceil + 1$ iterations, which proves the statement. ∎

[5] For the case $\varrho < 0$ one gets a slightly worse bound: $1 - \frac{3}{14}(\varrho-\gamma_t)^2$.

A similar result is found for $\varrho < 0$ in [15], but without the additional factor $1 - \varrho^2$; in that case the number of iterations is bounded by $\lceil 2\log(N)/\nu^2 \rceil + 1$.

3.3 Asymptotical Margin of AdaBoost$_\varrho$

With the methods shown so far, we can also analyse to what value the maximum margin of the original AdaBoost algorithm converges to asymptotically. First, we state a lower bound on the margin that is achieved by AdaBoost$_\varrho$. We find, that the size of the margin is not as large as it could be theoretically (by considering Theorem 1). In a second part we consider an experiment that shows that depending on some subtle properties of the base learner, the margin of combined hypotheses generated by AdaBoost can converge to quite different values (while the maximum margin is kept constant). We observe that the previously lower bound on the margin is almost tight in empirical cases.

As long each factor in Eq. (3) is smaller than 1, the right hand side of the bound reduces. If it is bounded away from 1,[6] then the rhs. converges exponentially fast to zero. The following corollary considers the asymptotical case and gives a lower bound on the margin.

Corollary 5 ([15]). *Assume AdaBoost$_\varrho$ generates hypothesis $h_1, h_2, \ldots$ with edges γ_1, $\gamma_2, \ldots$ and coefficients $\alpha_1, \alpha_2, \ldots$. Let $\gamma = \inf_{t=1,2,\ldots} \gamma_t$ and assume $\gamma > \varrho$. Then the asymptotically ($t \to \infty$) smallest margin $\hat{\rho}$ of the combined hypothesis is bounded from below by*

$$\hat{\rho} \geq \frac{\log(1-\varrho^2) - \log(1-\gamma^2)}{\log\left(\frac{1+\gamma}{1-\gamma}\right) - \log\left(\frac{1+\varrho}{1-\varrho}\right)}. \tag{6}$$

From (6) one can understand the interaction between ϱ and γ: if the difference between γ and ϱ is small, then the right hand side of (6) is small. Thus, if ϱ with $\varrho \leq \gamma$ is large, then $\hat{\rho}$ must be large, i.e. choosing a larger ϱ results in a larger margin on the training examples. Note that (6) implies the weaker bound $\hat{\rho} \geq \frac{\gamma+\varrho}{2}$. Previously it was only known that $\hat{\rho} \geq \varrho$ [2, Theorem 7.2].

However, in the Section 3.1 we have shown that the maximal achievable margin is at least γ. Thus if ϱ is chosen too small, then we guarantee only a *suboptimal asymptotical margin*. In the original formulation of AdaBoost we have $\varrho = 0$ and we guarantee only that AdaBoost$_0$ achieves a margin of at least $\frac{1}{2}\gamma + \frac{1}{12}\gamma^3$. *This gap in the theory motivates our Marginal AdaBoost algorithm.*

Experimental Illustration of Corollary 5: To illustrate that the above-mentioned gap, we perform an experiment showing how tight (6) can be. We analyze two different settings: the base learner selects (i) the hypothesis with largest edge over all hypotheses and (ii) the hypothesis with minimal edge among all hypothesis with edge larger than ϱ^*. This is the best and the worst case, respectively. Note that Corollary 5 holds in particular for the *worst case*, where the base learner is allowed to return any hypothesis with edge larger than ϱ^* (the maximal achievable margin). In practice, however, the base learner may perform better.

[6] This means that there exists a constant $\mu > 0$ such that the square-root term is *always* smaller than $1 - \mu$. Otherwise, the sequence of square-root terms might converge to 1 and the product of them might not converge to zero (this case is analyzed in greater detail in [15].

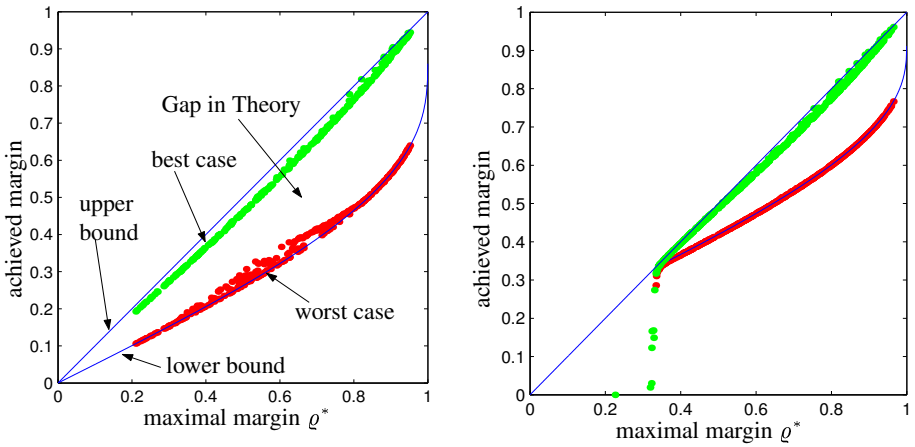

Fig. 1: Achieved margins of AdaBoost$_\varrho$ using the best (green) and the worst (red) selection on random data for $\varrho = 0$ [left] and $\varrho = \frac{1}{3}$ [right]: On the abscissa is the maximal achievable margin ϱ^* and on the ordinate the margin achieved by AdaBoost$_\varrho$ for one data realization. For comparison the line $y = x$ ("upper bound") and the bound (6) ("lower bound") are plotted. On the interval $[\varrho, 1]$, Eq. (6) lower bounds the achieved margin of the worst strategy tightly. The optimal strategy almost achieves the maximal margin. There is a clear gap between best and worst performance of the selection scheme, which we cannot explain by theory. If ϱ is chosen appropriately, then this gap is reduced to 0.

We use random data with N training examples, where N is drawn uniformly between 10 and 200. The labels are drawn at random from a binomial distribution with equal probability. We use a hypothesis set with 10^4 hypotheses. The output of every hypothesis is either -1 or $+1$ and is chosen such that with probability p it is equal to the previously generated label.[7] First we compute the solution of the margin-LP problem (left hand side of (1)) and determine ϱ^*. Then we compute the combined hypothesis generated by AdaBoost$_\varrho$ after 10^4 iterations for $\varrho = 0$ and $\varrho = \frac{1}{3}$ using the best and the worst selection strategy, respectively. The latter depends on ϱ^*. This procedure is repeated for 300 random draws of p from an uniform distribution. The parameter p ensures that there are cases with small and large optimal margins.

The resulting margins computed by the three algorithms are plotted in Figure 1. On the abscissa is the maximal achievable margin ϱ^* for a particular data realization. On the ordinate is the margin of AdaBoost$_\varrho$ using the best and the worst strategy. We observe a great difference between both selection strategies. Whereas the margin of the worst strategy is *tightly lower bounded* by (6), the best strategy has near maximal margin.

These experiments show that one obtains different results by changing some properties of the base learning algorithm. These properties are not used for computing the bound (only the minimal edge is used). This is indeed a problem, as one is not able to predict where AdaBoost$_\varrho$ is converging to.[8] We therefore propose Marginal AdaBoost, which solves this problem. Roughly speaking, by choosing ϱ near ϱ^*, one can make the difference between worst and best case performance of the base learner arbitrary small (cf. Figure 1). This is exploited in the next section.

[7] With low probability, the output of an hypothesis is equal to all labels. These cases are excluded.

[8] One might even be able to construct cases where the outputs are not at all converging.

3.4 Convergence of Marginal AdaBoost

So far we have always considered the case where we already know some proper value of ϱ. Let us now assume the case that the maximum achievable margin is ϱ^* and we would like to achieve a margin of $\varrho^* - \nu$. Our algorithm *guesses* a value of ϱ^* starting with 0. We need to understand what happens if our guess is too high, i.e. $\varrho > \varrho^*$, or too low, i.e. $\varrho < \varrho^*$?

First, if $\varrho > \varrho^*$, then one cannot reach the margin of ϱ since the maximum achievable margin is ϱ^*. By Corollary 4, if AdaBoost$_\varrho$ has not reached a margin of at least ϱ in $2\log(N)/\nu^2$ steps, we can conclude that $\varrho > \varrho^* - \nu$ (cf. step 3c in Marginal AdaBoost). This is the worst case. The better case is, if the distribution $\mathbf{d}$ generated by AdaBoost$_\varrho$ will be too difficult for the base learner and it eventually fails to achieve an edge γ of at least ϱ and AdaBoost$_\varrho$ will stop (cf. stopping condition in Marginal AdaBoost).

Second, assume ϱ is chosen too low, say $\varrho < \varrho^* - \nu$, then one achieves a margin of ϱ in a few steps by Corollary 4 . Since the maximum margin is always greater that a certain achieved margin, one can conclude that $\varrho^* \geq \varrho$ (cf. step 3c in Marginal AdaBoost).

Note that there is a *small gap* in the proposed binary search procedure: We are not able to identify the case $\varrho^* - \nu \leq \varrho \leq \varrho^*$ efficiently. This means that we cannot reduce the length of the search interval by *exactly* a factor of two in *each* iteration. This makes the analysis slightly more difficult, but eventually leads to the following theorem on the *worst case performance* of Marginal AdaBoost:

Theorem 6. *Assume the base learner always achieves an edge $\gamma_t \geq \varrho^*$. Then Marginal AdaBoost (Algorithm 2) will find a combined hypothesis f that maximizes the margin up to accuracy 4ν in at most $\lceil \frac{2\log(N)}{\nu^2} + 1 \rceil \lceil \log_2(2/\nu) \rceil$ calls of the base learner. The final hypothesis combines at most $\lceil \frac{2\log(N)}{\nu^2} + 1 \rceil$ base hypotheses.*

Proof. See Algorithm 2 for definitions of $u_r, l_r, \gamma_r, \rho_r$.

We claim that in any iteration $u_r \geq \varrho^* \geq l_r$. We show, if $u_{r-1} \geq \varrho^* \geq l_{r-1}$, then $u_r \geq \varrho^* \geq l_r$ for all $r = 1, \ldots, R$. It holds $l_1 \geq \varrho^* \geq u_1$ (induction start). By assumption $\gamma_{t,r} \geq \varrho^*$ and we may set $u_{r+1} = \min_{r_0=1,\ldots,r} \min_{t=1,\ldots,T} \gamma_{t,r}$. By Theorem 1 holds $\varrho^* \geq \rho_r$ for all $r = 1, 2, \ldots$ and, hence, we may set $l_{r+1} = \max_{r_0=1,\ldots,r} \rho_{r_0}$. We have to consider two cases. (a) $\rho_r \geq \varrho_r$ and (b) $\rho_r < \varrho_r$. In case (b) we have an additional term in u_{r+1}, which follows from $\varrho_r + \nu \geq \varrho^*$, justified by $\rho_r < \varrho_r$ and Proposition 3.

By construction, the length of interval $[l_r, u_r]$ is (almost) decreased in each iteration by a factor of two. We show $u_r - l_r \leq 2^{-r+1} + 2\nu$. In case (a) the interval is reduced by at least a factor of two. The worst case is if always (b) happens:

$$\begin{aligned} u_r - l_r &\leq \varrho_r + \nu - l_r \\ &= \frac{\varrho_{r-1} + \nu + l_{r-1}}{2} + \nu - l_r \\ &= \frac{\varrho_{r-1} + \nu + l_{r-1} + 2\nu - 2l_r}{2} \end{aligned}$$

$$= \frac{\varrho_{r-j} + \nu \sum_{i=0}^{j} 2^j + \sum_{i=1}^{j} 2^{j-i} l_{r-i} - 2^j l_r}{2^j}$$
$$= \frac{\varrho_{r-j} + (2^{j+1}-1)\nu + \sum_{i=1}^{j} 2^{j-i} l_{r-i} - 2^j l_r}{2^j}$$
$$= \frac{\varrho_1 + (2^r - 1)\nu + \sum_{i=1}^{r-1} 2^{r-1-i} l_{r-i} - 2^{r-1} l_r}{2^{r-1}}.$$

Since l_r is non-decreasing, $\sum_{i=1}^{j} 2^{r-1-i} l_{r-i} - 2^{r-1} l_r$ is maximized for $l_r = \text{const}$, i.e. $\sum_{i=1}^{j} 2^{r-1-i} l_{r-i} - 2^{r-1} l_r \leq l_r \sum_{i=0}^{r-2} 2^i - 2^{r-1} l_r = l_r(2^{r-1} - 1 - 2^{r-1}) = -l_r$. We continue by using $\varrho_1 = 0$ and $\min_r l_r = -1$:

$$\leq \frac{(2^r - 1)\nu - l_r}{2^{r-1}}$$
$$\leq \frac{1}{2^{r-1}} + 2\nu.$$

Thus after $R = \lceil \log_2(1/\nu) \rceil$ steps we have $u_{R+1} - l_{R+1} \leq 3\nu$ and $\varrho^* - l_{R+1} \leq 3\nu$.[9]

Now we run AdaBoost $_{l_{R+1}-\nu}(S, 2\log(N)/\nu^2)$ and achieve a margin of at least $l_{R+1} - \nu$ by Corollary 4. This can only be 4ν away from ϱ^*.

We called $R + 1 = \lceil \log_2(1/\nu) + 1 \rceil$ times AdaBoost$_\varrho$, each time calling at most $\lceil 2\log(N)/\nu^2 + 1 \rceil$ times the base learning algorithm. Marginal AdaBoost returns only the last hypothesis, combining only $\lceil 2\log(N)/\nu^2 + 1 \rceil$ base hypotheses. ∎

The bound could be improved by a factor $1 - \varrho^{*2}$, if $\varrho \geq 0$, since we have not exploited the additional factor $(1 - \varrho^2)$ as in Proposition 3.

3.5 Infinite Hypothesis Sets

So far we have implicitly assumed that the hypothesis space is finite. In this section we will show that this assumption is (often) not necessary. Also note, if the output of the hypotheses is discrete, the hypothesis space is effectively finite [16]. For *infinite hypothesis sets*, Theorem 1 can be restated in a weaker form as:

Theorem 7 (Weak Min-Max, e.g. [12]).

$$\gamma^* := \min_{\mathbf{d}} \sup_{h \in H} \sum_{n=1}^{N} y_n h(\mathbf{x}_n) d_n \geq \sup_{\boldsymbol{\alpha}} \min_{n=1,\ldots,N} y_n \sum_{t:\alpha_t > 0} \alpha_t h_t(\mathbf{x}_n) =: \varrho^*, \quad (7)$$

where $\mathbf{d} \in \mathcal{P}^N$, $\boldsymbol{\alpha} \in \mathcal{P}^{|H|}$ *with finite support. We call* $\Gamma = \gamma^* - \varrho^*$ *the "duality gap".*

In particular for any $\mathbf{d} \in P^N$: $\sup_{h \in H} \sum_{n=1}^{N} y_n h(\mathbf{x}_n) d_n \geq \gamma^*$ and for any $\boldsymbol{\alpha} \in P^{|H|}$ with finite support: $\min_{n=1,\ldots,N} y_n \sum_{t:\alpha_t \geq 0} \alpha_t h_t(\mathbf{x}_n) \leq \varrho^*$.

In theory the duality gap may be nonzero. However, Proposition 3 and Theorem 6 do not assume finite hypothesis sets and show that the margin will converge arbitrarily close to ϱ^*, as long as the base learning algorithm can return *a single* hypothesis that has an edge not smaller than ϱ^*.

[9] Note, if one chooses ϱ_r not exactly in the middle of the interval $[l_r, u_r]$, one can balance the influence of ν and obtain a slightly better constant for the worst case performance.

In other words, the duality gap may result from the fact that the $\sup$ on the left side can not be replaced by a $\max$, i.e. there might not exists a *single* hypothesis h with edge larger or equal to ϱ^*. By assuming that the base learner is always able to pick good enough hypotheses ($\geq \varrho^*$) one automatically gets that $\Gamma = 0$ (by Proposition 3).

Under certain conditions on H this maximum always exists and strong duality holds (see e.g. [16, 15, 8, 12] for details):

Theorem 8 (Strong Min-Max). *If the set* $\{[h(\mathbf{x}_1), \ldots, h(\mathbf{x}_N)] \mid h \in H\}$ *is compact, then* $\Gamma = 0$.

In general this requirement can be fulfilled by base learning algorithms whose outputs continuously depend on the distribution $\mathbf{d}$. Furthermore, the outputs of the hypotheses need to be bounded (cf. step 3a in AdaBoost$_\varrho$). The first requirement might be a problem with base learning algorithms such as some variants of decision stumps or decision trees. However, there is a simple trick to avoid this problem: Roughly speaking, at each point with discontinuity $\hat{\mathbf{d}}$, one adds all hypotheses to H that are limit points of $L(S, \mathbf{d}^s)$, where $\{\mathbf{d}^s\}_{s=1}^{\infty}$ is an arbitrary sequence converging to $\hat{\mathbf{d}}$ and $L(S, \mathbf{d})$ denotes the hypothesis returned by the base learning algorithm for weighting $\mathbf{d}$ and training sample S [15].

4 Experimental Illustration of Marginal AdaBoost

First of all, we would like to note that we are aware of the fact that maximizing the margin of the ensemble does not lead in all cases to an improved generalization performance. For fairly noisy data sets even the opposite has been reported [cf. 14, 2, 7, 17]. Also, Breiman once reported an example where the margins of all examples are larger in one ensemble than another and the latter generalized considerably better.

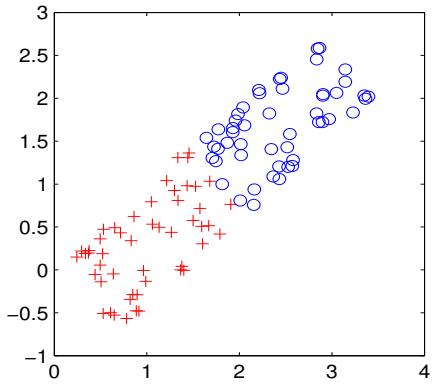

Fig. 2: The two *discriminative dimensions* of our separable one hundred dimensional data set.

However, at least for well separable data the theory applies for *hard margins* and for larger margins one can prove better generalization error bounds. We expect that one should be able to measure differences in the generalization error, if one function approximately maximizes the margin while another function does not, as similar results have been obtained in [19] on a multi-class optical character recognition problem.

Here we report experiments on artificial data to (a) illustrate how our algorithm works and (b) how it compares to AdaBoost. Our data is 100 dimensional and contains 98 nuisance dimensions with uniform noise. The other two dimensions are plotted exemplary in Figure 2. For training we use only 100 examples and there is obviously the need to carefully control the capacity of the ensemble.

As base learning algorithm we use C4.5 decision trees provided by Ross Quinlan [13, cf.] using an option to control the number of nodes in the tree. We have set it such that C4.5 generates trees with about three nodes. Otherwise, the base learner often classifies all training examples correctly and over-fits the data already. Furthermore, since in this case the margin is already maximal (equal to 1), both algorithms would stop since $\gamma = 1$. We therefore need to limit the complexity of the base learner, in good agreement with the bounds on the generalization error [19].

In Figure 3 (left) we see a typical run of Marginal AdaBoost for $\nu = 0.1$. It calls AdaBoost$_\varrho$ three times. The first call of AdaBoost$_\varrho$ for $\gamma = 0$ already stops after four iterations, since it has generated a consistent combined hypothesis. The lower bound l on ϱ^* as computed by our algorithm is $l = 0.07$ and the upper bound u is 0.94 (cf. step 3c in Marginal AdaBoost). The second time ϱ is chosen to be in the middle of the interval $[l, u]$ and AdaBoost$_\varrho$ reaches the margin of $\varrho = 0.51$ after 80 iterations. The interval is now $[0.51, 0.77]$. Since the length of the interval $u - l = 0.27$ is small enough, Marginal AdaBoost leaves the loop through exit condition 3d, calls AdaBoost$_\varrho$ the last time for $\varrho = u - \nu = 0.41$ and finally achieves a margin of $\rho = 0.55$. For comparison we also plot the margins of the hypotheses generated by AdaBoost (cf. Figure 3 (right)). One observes that it is not able to achieve a large margin efficiently ($\rho = 0.37$ after 1000 iterations).

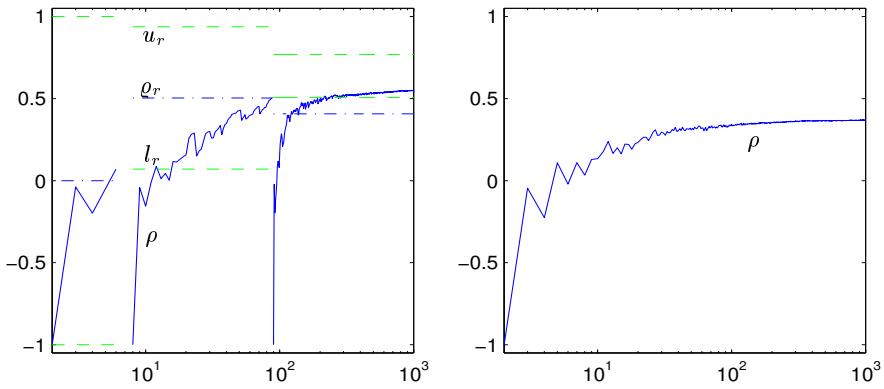

Fig. 3: Illustration of the achieved margin of Marginal AdaBoost (upper) and AdaBoost$_0$ (lower) at each iteration. Our algorithm calls AdaBoost$_\varrho$ three times while adapting ϱ (dash-dotted). We also plot the values for l and u as in Marginal AdaBoost (dashed). (For details see text.)

	C4.5	AB	Marginal AB
E_{gen}	$7.4 \pm 0.11\%$	$4.0 \pm 0.11\%$	$3.6 \pm 0.10\%$
ϱ	—	0.31 ± 0.01	0.58 ± 0.01
wins	$1/200$	$59/200$	$140/200$

Table 1: Estimated generalization performances and margins with confidence intervals for decision trees (C4.5), AdaBoost (AB) and Marginal AB on the toy data. The last row shows the number of times the algorithm had the smallest error. All numbers are averaged over 200 splits into 100 training and 19900 test examples.

In Table 1 we see the average performance of the three classifiers. For AdaBoost we combined 200 hypotheses for the final prediction. For Marginal AdaBoost we use $\nu = 0.1$ and let the algorithm combine only 200 hypotheses for the final prediction to get a fair comparison. We see a large improvement of both ensemble methods compared to the single classifier. There is also a slight, but – according to a T-test with confidence level 98% – significant difference between the generalization performances of both boosting algorithms. Note also that the margins of the combined hypothesis achieved by Marginal AdaBoost are on average almost twice as large as for AdaBoost.

5 Conclusion

We have analyzed a generalized version of AdaBoost in the context of large margin algorithms. From von Neumann's Min-Max theorem the maximal achievable margin ϱ^* is at least γ, if the base learner always returns a hypothesis with weighted classification error less than $\frac{1}{2} - \frac{1}{2}\gamma$. The asymptotical analysis lead us to a lower bound on the margin of the combined hypotheses generated by AdaBoost$_\varrho$ in the limit, which was shown to be rather tight in empirical cases. Our results indicate that AdaBoost generally does not maximize the margin, but achieves a reasonable large margin.

To overcome these problems we proposed an algorithm for which we have shown the convergence to the maximum margin solution. This is achieved by increasing ϱ iteratively, such that the gap between the best and the worst case becomes arbitrarily small. In our analysis we did not need to assume additional properties of the base learning algorithm. To the best of our knowledge this is the first algorithm that combines the merits of Boosting with the maximum margin solution.

In a simulation experiment we have illustrated the validity of our analysis and also that a larger margin can improve the generalization ability when learning on high dimensional data with a few informative dimensions.

References

[1] K.P. Bennett, A. Demiriz, and J. Shawe-Taylor. A column generation algorithm for boosting. In P. Langley, editor, *Proceedings, 17th ICML*, pages 65–72, San Francisco, 2000.

[2] L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11(7):1493–1518, 1999. Also Technical Report 504, Statistics Dept., University of California Berkeley.

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

[4] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Proc. 13th International Conference on Machine Learning*, pages 148–146. Morgan Kaufmann, 1996.

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

[6] Y. Freund and R.E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[7] A.J. Grove and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proc. of the Fifteenth National Conference on Artifical Intelligence*, 1998.

[8] R. Hettich and K.O. Kortanek. Semi-infinite programming: Theory, methods and applications. *SIAM Review*, 3:380–429, September 1993.

[9] J. Kivinen and M. Warmuth. Boosting as entropy projection. In *Proc. 12th Annu. Conference on Comput. Learning Theory*, pages 134–144. ACM Press, New York, NY, 1999.

[10] V. Koltchinskii, D. Panchenko, and F. Lozano. Some new bounds on the generalization error of combined classifiers. In *Advances in Neural Inf. Proc. Systems*, volume 13, 2001.

[11] O.L. Mangasarian. Arbitrary-norm separating plane. *Op. Res. Letters*, 24(1):15–23, 1999.

[12] S. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York, 1996.

[13] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1992.

[14] J.R. Quinlan. Boosting first-order learning. *Lecture Notes in Comp. Sci.*, 1160:143, 1996.

[15] G. Rätsch. *Robust Boosting via Convex Optimization*. PhD thesis, University of Potsdam, October 2001. http://mlg.anu.edu.au/˜raetsch/thesis.ps.gz.

[16] G. Rätsch, A. Demiriz, and K. Bennett. Sparse regression ensembles in infinite and finite hypothesis spaces. *Machine Learning*, 48(1-3):193–221, 2002. Special Issue on New Methods for Model Selection and Model Combination. Also NeuroCOLT2 Technical Report 2000-085.

[17] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, March 2001. also NeuroCOLT Technical Report NC-TR-1998-021.

[18] R.E. Schapire. *The Design and Analysis of Efficient Learning Algorithms*. PhD thesis, MIT Press, 1992.

[19] R.E. Schapire, Y. Freund, P.L. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651 ff., 1998.

[20] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, December 1999. also Proceedings of the 14th Workshop on Computational Learning Theory 1998, pages 80–91.

[21] L.G. Valiant. A theory of the learnable. *Comm. of the ACM*, 27(11):1134–1142, 1984.

[22] J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Math. Ann.*, 100:295–320, 1928.

[23] T. Zhang. Sequential greedy approximation for certain convex optimization problems. Technical report, IBM T.J. Watson Research Center, 2002.