Journal of Machine Learning Research 6 (2005) 2131–2152 Submitted 1/03; Revised 7/05; Published 12/05

Efficient Margin Maximizing with Boosting*

Gunnar Rätsch GUNNAR.RAETSCH@TUEBINGEN.MPG.DE
Friedrich Miescher Laboratory of the Max Planck Society
Spemannstrasse 35
72076 Tübingen, Germany

Manfred K. Warmuth MANFRED@CSE.UCSC.EDU
University of California at Santa Cruz
Santa Cruz, CA 95060, USA

Editor: John Shawe-Taylor

Abstract

AdaBoost produces a linear combination of base hypotheses and predicts with the sign of this linear combination. The linear combination may be viewed as a hyperplane in feature space where the base hypotheses form the features. It has been observed that the generalization error of the algorithm continues to improve even after all examples are on the correct side of the current hyperplane. The improvement is attributed to the experimental observation that the distances (margins) of the examples to the separating hyperplane are increasing even after all examples are on the correct side.

We introduce a new version of AdaBoost, called AdaBoost$^*_\nu$, that explicitly maximizes the minimum margin of the examples up to a given precision. The algorithm incorporates a current estimate of the achievable margin into its calculation of the linear coefficients of the base hypotheses. The bound on the number of iterations needed by the new algorithms is the same as the number needed by a known version of AdaBoost that must have an explicit estimate of the achievable margin as a parameter. We also illustrate experimentally that our algorithm requires considerably fewer iterations than other algorithms that aim to maximize the margin.

1. Introduction

Boosting algorithms are greedy methods for forming linear combinations of base hypotheses. In the most common scenario the algorithm is given a fixed set of labeled training examples and in each iteration updates a distribution on these examples (i.e. a set of non-negative weights that sum to one). It then is given a *base* hypothesis whose weighted error (probability of wrong classification) is slightly below 50%. This base hypothesis is used to update the distribution on the examples: The algorithm increases the weights of those examples that were wrongly classified by the base hypothesis. At the end of each stage the base hypothesis is added to the linear combination, and the sign of this linear combination forms the current hypothesis of the boosting algorithm.

*. Part of this work was done while G. Rätsch was at Fraunhofer FIRST Berlin, at UC Santa Cruz, the Australian National University and the Max Planck Institute for biological Cybernetics. 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. Sonnenburg, S. Lemm and K.-R. Müller for discussions. M.K. Warmuth thanks J. Liao and Karen Glocer for their help.

The most well known boosting algorithm is AdaBoost (Freund and Schapire, 1997). It is "adaptive" in that the linear coefficient of the base hypothesis depends on the weighted error of the base hypothesis at the time when the base hypothesis was added to the linear combination. AdaBoost has two interesting properties. First, along with earlier boosting algorithms (Schapire, 1992; Freund, 1995), its training error has the following exponential convergence property: if the weighted training error of the t-th base hypothesis is $\varepsilon_t = \frac{1}{2} - \frac{1}{2}\gamma_t$, then an upper bound on the training error of the signed 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 signed linear combination is zero (Schapire et al., 1998). That is, in experiments the generalization error continues to improve. The signed linear combination can be viewed as a homogeneous *hyperplane* in a feature space, where each base hypothesis represents one feature or dimension. We define the *margin* of an example as a signed distance to the hyperplane times its $\pm$ label (See Section 2 and Appendix A for precise definitions). As soon as the training error is zero, the examples are on the right side and all 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. Schapire et al., 1998; Breiman, 1999; Koltchinskii et al., 2001) that improve with the margin of the classifier, which is defined as the size of the minimum margin of the examples. Thus the fact that the margins improve experimentally seems to explain why AdaBoost still learns after the training error is zero.

There is one flaw in this argument: AdaBoost has not been proven to maximize the margin of the final hypothesis. We demonstrate this experimentally in Section 5. Moreover, Rudin et al. (2004a, 2005) recently showed that there are cases where AdaBoost provably does not maximize the margin. Breiman (1999) proposed a modified algorithm – called 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 Grove and Schuurmans (1998) and Bennett et al. (2000). In this paper we present an algorithm that produces a final hypothesis with margin at least $\rho^* - \nu$, where ρ^* is the unknown maximum margin achievable by any convex combination of base hypotheses and ν a precision parameter.

If we know ρ^*, then a linear combination with margin at least $\rho^* - \nu$ can be found by a parameterized version of AdaBoost called AdaBoost$_\rho$ (cf. Rätsch et al. (2001); Rätsch and Warmuth (2002)): When the parameter ρ of AdaBoost$_\rho$ is set to $\rho^* - \nu$, then after $\frac{2\ln N}{\nu^2}$ iterations, where N is the number of examples, the margin of the produced linear combination is guaranteed to be at least $\rho^* - \nu$. The case when ρ^* is not known is more difficult. In a preliminary conference paper (Rätsch and Warmuth, 2002) we used AdaBoost$_\rho$ iteratively in a binary search like fashion: $\log_2(2/\nu)$ calls to AdaBoost$_\rho$ are guaranteed to produce a margin at least $\rho^* - \nu$. All but the last call to AdaBoost$_\rho$ are used to find a suitable value of the parameter ρ and in the last call this parameter is used to create the final linear combination in at most $\frac{2\ln N}{\nu^2}$ iterations.

In this paper we greatly simplify our answer for the case when ρ^* is unknown. We have a new *one pass* algorithm called AdaBoost$^*_\nu$ that produces a linear combination with margin at least $\rho^* - \nu$ after $\frac{2\ln N}{\nu^2}$ iterations. Note that this is the same guarantee we had on the number of iterations of AdaBoost$_\rho$ when it used the theoretically optimal parameter $\rho = \rho^* - \nu$. The new algorithm AdaBoost$^*_\nu$ uses the parameter ν and a *current estimate* of the achievable margin in the computation of the linear coefficients of the base learners.

Except for the algorithm presented in the previous conference paper, this is the first result on the fast convergence of a boosting algorithm to the maximum margin solution that works for all $\rho^* \in [-1, 1]$. Using previous results one can only show that AdaBoost *asymptotically* converges to

a final hypothesis with margin at least $\rho^*/2$ if $\rho^* > 0$ and if subtle conditions on the chosen base hypotheses are satisfied (cf. Corollary 5).

Recently other versions of AdaBoost have been published that are guaranteed to produce a linear combination of margin at least $\rho^* - \nu$ after $\Omega(\nu^{-3})$ iterations (Rudin et al., 2004c,b). Even though these algorithms have weaker iteration bounds than AdaBoost$^*_\nu$, they were reported to perform better experimentally (Rudin et al., 2004c,a). We briefly compare AdaBoost$^*_\nu$ to these more recent algorithms and show that the better empirical performance was due to the wrong choice of ν.

The original 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 (i.e. $\rho^* < 0$), where one minimizes the overlap between the two classes. In this case AdaBoost runs forever without necessarily increasing the margin. Our algorithm is also useful when the base 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 $0 < \rho^* < 1$ and AdaBoost aborts immediately because the linear coefficients of such hypotheses become unbounded. In contrast, our new algorithm also maximizes the margin when presented with strong learners.

The big advantage of this algorithm is an absolute bound on the number of iterations: After $\frac{2\ln N}{\nu^2}$ iterations AdaBoost$^*_\nu$ is guaranteed to produce a hypothesis with margin at least $\rho^* - \nu$. Our algorithm is applicable in sophisticated settings where the number of hypotheses may be infinite. In Appendix B we use AdaBoost$^*_\nu$ to learn a convex combination of support vector kernels and show that the same guarantees hold on the number of iterations of the algorithm.

The paper is structured as follows: Section 2 introduces some basic notation and in Section 3 we first describe *AdaBoost*$_\rho$ which requires a lower bound ρ of the maximum margin ρ^* as a parameter. Then we present our new algorithm *AdaBoost*$^*_\nu$, which is similar to AdaBoost$_\rho$, but continuously adapts ρ based on a precision parameter ν. Up to this point we stay at a high level of presentation with the goal of making our algorithms accessible to the quick reader. In Section 4 we introduce more notation and give a detailed analysis of both algorithms. First, we prove that if the weighted training error of the t-th base hypothesis is $\varepsilon_t = \frac{1}{2} - \frac{1}{2}\gamma_t$, then an upper bound on the fraction of examples with margin smaller than ρ is reduced by a factor of $1 - \frac{1}{2}(\rho - \gamma_t)^2$ at stage t of AdaBoost$_\rho$ (cf. Section 4.2) (A slightly improved factor is shown for the case when $\rho > 0$). However, to achieve a large margin one needs to assume that the guess ρ is smaller than ρ^*. For the latter case we prove an exponential convergence rate of AdaBoost$_\rho$. Then we discuss a method for automatically tuning ρ depending on the errors of the base hypotheses and a precision parameter ν. We show that after roughly $\frac{2\ln N}{\nu^2}$ iterations our new one-pass algorithm AdaBoost$^*_\nu$ is guaranteed to produce a linear combination with margin at least $\rho^* - \nu$. This strengthens the results of our preliminary conference paper (Rätsch and Warmuth, 2002), which had an additional $\log_2(2/\nu)$ factor in the total number times the weak learner is called and much higher constants. In Section 5, we compare the algorithms experimentally and discuss heuristics for tuning ν in Section 5.2. Finally we briefly summarize and discuss our results in the Conclusion Section.

2. Preliminaries and Basic Notation

We consider the standard two-class supervised machine learning problem: Given a set of N i.i.d. training examples $(\mathbf{x}_n, y_n)$, $n = 1, \ldots, N$, with $\mathbf{x}_n \in \mathcal{X}$ and $y_n \in \mathcal{Y} := \{-1, +1\}$, we would like to learn a function $f : \mathcal{X} \to \mathcal{Y}$ that is able to generalize well on unseen data generated from the same distribution as the training data.

In the case of ensemble learning (like boosting), there is a fixed underlying set of *base* hypotheses $\mathcal{H} := \{h \mid h : \mathcal{X} \to [-1,1]\}$ from which the ensemble is built. For now we only assume that $\mathcal{H}$ is finite, but we will show in Section 4.5 that this assumption can be dropped in most cases and that all of the following analysis also applies to the case of infinite hypothesis sets.

Boosting algorithms iteratively form non-negative linear combinations of hypotheses from $\mathcal{H}$. In each iteration t, a base hypothesis $h_t \in \mathcal{H}$ with a non-negative coefficient α_t is added to the linear combination. We denote the combined hypothesis as follows (Note that we normalized the weights):

$$\tilde{f}_\alpha(\mathbf{x}) = \operatorname{sign} f_\alpha(\mathbf{x}), \text{ where } f_\alpha(\mathbf{x}) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=t}^{T} \alpha_r} h_t(\mathbf{x}),\ h_t(\mathbf{x}) \in \mathcal{H}, \text{ and } \alpha_t \geq 0 \ .$$

The "black box" that selects the base hypothesis in each iteration is called the *weak* learner. For AdaBoost, it has been shown that if the weak learner is guaranteed to select base hypotheses of weighted error slightly below 50%, then the combined hypothesis is consistent with the training set in a small number of iterations (Freund and Schapire, 1997). We will discuss bounds on the number of iterations in detail in Section 4. Since at most one new base hypothesis is added in each iteration, the size of the final hypothesis is bounded by the number of iterations. These bounds are important because the sample size bounds provable in the PAC model grow with the size of the final hypothesis (Schapire, 1992; Freund, 1995).

In more recent research (Schapire et al., 1998) it was also shown that a bound on the generalization error decreases with the size of the margin of the final hypothesis f. The margin of a single example $(\mathbf{x}_n, y_n)$ w.r.t. f is defined as $y_n f_\alpha(\mathbf{x}_n)$. Thus the margin quantifies by how far this example is on the y_n side of the hyperplane $\tilde{f}$. In Appendix A we clarify how the margin of an example is related to its ℓ_∞-distance to the hyperplane with normal α. The margin of the combined hypothesis f is the *minimum margin* of all N examples. The goal of this paper is to find a small non-negative linear combination of base hypotheses from $\mathcal{H}$ with margin close to the maximum achievable margin.

The following table gives some of the main notations that will be used throughout this paper:

Symbol	Description
n, N	index and number of examples
m, M	index and number of hypotheses if finite
t, T	index and number of iterations
$\mathcal{X}$	input space
$\mathcal{Y}$	label space $\{\pm 1\}$
$(\mathbf{x}, y)$	an example and its label
$\mathcal{H}, h_m$	set of base hypotheses and the m-th element
α	hypothesis weight vector
$\mathbf{d}$	weighting on the training set
$\mathbf{I}(\cdot)$	the indicator function: $\mathbf{I}(true) = 1$ and $\mathbf{I}(false) = 0$
ρ	the margin parameter of AdaBoost_ρ
$\{\rho_t\}$	the sequence of margin parameters of $\text{AdaBoost}_{\{\rho_t\}}$
ρ^*	the maximum margin
$\hat{\rho}_t$	margin in the t-th iteration
ν	the accuracy parameter of AdaBoost^*_ν
ε	weighted classification error
γ^*	the minimum edge

Symbol	Description
γ	an arbitrary edge threshold
γ_t	the edge of the t-th hypothesis

3. AdaBoost$_\rho$ and AdaBoost$^*_\nu$

The original AdaBoost was designed to find a consistent hypothesis $\tilde{f}$ which is defined as a signed linear combination f with margin greater zero. We start with a slight modification of AdaBoost, which finds (if possible) a linear combination of base learners with margin ρ, where ρ is a parameter (cf. Algorithm 1).[1] We call this algorithm AdaBoost$_\rho$, as it naturally generalizes AdaBoost for the case when the *target margin* is ρ. The original AdaBoost algorithm now becomes AdaBoost$_0$.

Algorithm 1: – The AdaBoost$_\rho$ algorithm – with margin parameter ρ

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:** $d_n^1 = \frac{1}{N}$ 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** $|\gamma_t| = 1$, **then** $\alpha_1 = \text{sign}(\gamma_t)$, $h_1 = h_t$, $T = 1$; break
   (d) Set $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2} \ln \frac{1+\rho}{1-\rho}$
   (e) Update weights: $d_n^{t+1} = \frac{d_n^t \exp\left(-\alpha_t y_n h_t(\mathbf{x}_n)\right)}{Z_t}$,
   where $Z_t = \sum_{n=1}^{N} d_n^t \exp\left(-\alpha_t y_n h_t(\mathbf{x}_n)\right)$
4. **Output:** $f_\alpha(\mathbf{x}) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T} \alpha_r} h_t(\mathbf{x})$

The algorithm AdaBoost$_\rho$ was already known as *unnormalized Arcing* (Breiman, 1999) or *AdaBoost-type Algorithm* (Rätsch et al., 2001). Moreover, it is related to algorithms proposed in Freund and Schapire (1999) and Zhang (2002). The only difference from AdaBoost is the choice of the hypothesis coefficients α_t: An additional term $-\frac{1}{2} \ln \frac{1+\rho}{1-\rho}$ appears in the expression for the hypothesis coefficient α_t. This term vanishes when $\rho = 0$. The parameter ρ can be seen as a *guess* of the maximum margin ρ^*. If ρ is chosen properly (slightly below ρ^*), then AdaBoost$_\rho$ will converge exponentially fast to a combined hypothesis with nearly the maximum margin. See Section 4.2 for details.

1. The original AdaBoost algorithm was formulated in terms of weighted training error ε_t of a base hypothesis. Here we use an equivalent more convenient formulation in terms of the edge γ_t, where $\varepsilon_t = \frac{1}{2} - \frac{1}{2}\gamma_t$ (cf. Section 4.1).

The following example illustrates how AdaBoost$_\rho$ works. Assume the weak learner returns the constant hypothesis $h_t(\mathbf{x}) \equiv 1$. The weighted error of this hypothesis is the sum of all negative weights, i.e. $\varepsilon_t = \sum_{y_n=-1} d_n^t$ and its edge is $\gamma_t = 1 - 2\varepsilon_t$. The coefficient α_t is chosen so that the edge of h_t with respect to the new distribution is exactly ρ (instead of 0 as for the original AdaBoost). More precisely, the given choice of α_t assures that this edge is ρ only for ± 1-valued base hypotheses.

For a more general base hypothesis h_t with continuous range $[-1,+1]$, choosing α_t such that Z_t as a function of α_t is minimized, guarantees that the edge of h_t with respect to the distribution $\mathbf{d}^{t+1}$ is ρ. See Schapire and Singer (1999) for a similar discussion. Choosing α_t as in step 3 (d) approximately minimizes Z_t when the range of h_t is $[-1,+1]$.

In Kivinen and Warmuth (1999) and Lafferty (1999), the standard boosting algorithms are interpreted as approximate solutions to the following optimization problem: choose a distribution $\mathbf{d}$ of maximum entropy subject to the constraints that the edges of the previous hypotheses are *equal* to zero. In this paper we use the *inequality* constraints that the edges of the previous hypotheses are at most ρ. The α_t's function as Lagrange multipliers for these inequality constraints. Since $g(x) = \frac{1}{2}\ln\frac{1+x}{1-x}$ is an increasing function,

$$\alpha_t = \frac{1}{2}\ln\frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2}\ln\frac{1+\rho}{1-\rho} \;\geq\; 0 \quad \text{iff} \quad \gamma_t \geq \rho \;. \tag{1}$$

Notice that when $\rho = 0$, adding h_t or $-h_t$ leads to the same distribution $\mathbf{d}^{t+1}$. This symmetry is broken for $\rho \neq 0$.

Since one does not know the value of the optimum margin ρ^* is not known beforehand, one also needs to find ρ^*. In Rätsch and Warmuth (2002) we presented the *Marginal AdaBoost* algorithm which constructs a sequence $\{\rho_r\}_{r=1}^R$ converging to ρ^*. A fast way to find a real value up to a certain accuracy ν in the interval $[-1,1]$ is a *binary search* since one needs only $\log_2(2/\nu)$ steps.[2] Thus the previous Marginal AdaBoost algorithm uses AdaBoost$_{\rho_r}$ (Algorithm 1) to decide whether the current guess ρ_r is *larger* or *smaller* than ρ^*. Depending on the outcome, ρ_r can be chosen so that the region of uncertainty for ρ^* is roughly cut in half. However, in the previous algorithm all but the last of the $\log_2(2/\nu)$

In this paper we propose a different algorithm, called AdaBoost$^*_\nu$. Here $\nu > 0$ is a precision parameter. The algorithm finds a non-negative linear combination with margin at least $\rho^* - \nu$. Like Arc-GV (Breiman, 1999), the new algorithm essentially runs AdaBoost$_\rho$ once but instead of using a fixed margin estimate ρ, it updates ρ in an appropriate way. We shall show iteration bounds for our algorithm AdaBoost$^*_\nu$ which are not known for Arc-GV. The latter algorithm produces an essentially[3] monotonically increasing sequence of margin estimates, while in AdaBoost$^*_\nu$ we use a monotonically decreasing sequence. The improved sequence of estimates is based on two new theoretical insights, which will be developed in the next section.

We will show that the number of iterations required by the new one-pass AdaBoost$^*_\nu$ algorithm (see Algorithm 2 for pseudo-code) is at most $\frac{2\ln N}{\nu^2}$. This equals the iteration bound for the best algorithm we know of for the case when ρ^* is known and we seek a linear combination of margin at least $\rho^* - \nu$: AdaBoost$_\rho$ with parameter $\rho = \rho^* - \nu$. The iteration bound for the new algorithm is the same as the iteration bound for the last call to AdaBoost$_\rho$ of the previous Marginal AdaBoost algorithm.

2. If one knows that $\rho^* \in [a,b]$, one needs only $\log_2((b-a)/\nu)$ steps.
3. In the original formulation the sequence was not necessarily increasing, but Rätsch (2001) showed that it leads to the same result and easier proofs if one restricts it to be monotonically increasing.

Algorithm 2: – The AdaBoost$^*_\nu$ algorithm – with accuracy parameter ν

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, No. of Iterations T, desired accuracy ν
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** $|\gamma_t| = 1$, **then** $\alpha_1 = \mathrm{sign}(\gamma_t)$, $h_1 = h_t$, $T = 1$; break
   (d) $\gamma_t^{\min} = \min_{r=1,\ldots,t} \gamma_r$; $\rho_t = \gamma_t^{\min} - \nu$
   (e) Set $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2} \ln \frac{1+\rho_t}{1-\rho_t}$
   (f) Update weights: $d_n^{t+1} = \frac{d_n^t \exp(-\alpha_t y_n h_t(\mathbf{x}_n))}{Z_t}$, where $Z_t = \sum_{n=1}^{N} d_n^t \exp(-\alpha_t y_n h_t(\mathbf{x}_n))$
4. **Output:** $f_\alpha(\mathbf{x}) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T} \alpha_r} h_t(\mathbf{x})$

4. Detailed Analysis

In this section we are going to analyze the algorithms in detail. We start by showing the relationship between optimal edges and margins, prove and illustrate the convergence properties of AdaBoost$_\rho$ and finally prove the convergence of AdaBoost$^*_\nu$.

4.1 Weak learning and margins

The standard assumption made on the weak learning algorithm for the PAC analysis of Boosting algorithm is that the weak learner returns a hypothesis h from a fixed set $\mathcal{H}$ that is slightly better than random guessing. That is, that the error rate ε is consistently smaller than $\frac{1}{2}$. Note that the error rate of $\frac{1}{2}$ could easily be reached by a fair coin, assuming both classes have the same prior probabilities. More formally, the error ε of a ± 1 valued hypothesis is defined as the fraction of examples that are misclassified. In Boosting this is extended to weighted example sets and the error is defined as

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

where h is the hypothesis returned by the weak learner and $\mathbf{I}$ is the indicator function with $\mathbf{I}(\text{true}) = 1$ and $\mathbf{I}(\text{false}) = 0$. The distribution $\mathbf{d} = (d_1, \ldots, d_N)$ of the examples is such that $d_n \geq 0$ and $\sum_{n=1}^{N} d_n = 1$.

When the range of a hypothesis h is the entire interval $[-1,+1]$, then the *edge* $\gamma_h(\mathbf{d}) = \sum_{n=1}^N d_n y_n h(\mathbf{x}_n)$ is a more convenient quantity for measuring the quality of h. This edge is an affine transformation of the error for the case when h has range ± 1: $\varepsilon_h(\mathbf{d}) = \frac{1}{2} - \frac{1}{2}\gamma_h(\mathbf{d})$ and $\varepsilon_h(\mathbf{d}) \leq \frac{1}{2}$ iff $\gamma_h(\mathbf{d}) \geq 0$.

Recall from Section 2 that the margin of a given example $(\mathbf{x}_n, y_n)$ is defined as $y_n f_\alpha(\mathbf{x}_n)$. Also recall that $\mathcal{H}$ is the set from which the weak learner chooses its base hypotheses. Assume for a moment that $\mathcal{H}$ is finite. If we combine all hypotheses from $\mathcal{H}$, then the following well-known theorem establishes the connection between margins and edges (first seen in connection with Boosting in Freund and Schapire, 1996; Breiman, 1999):[4]

Theorem 1 (Min-Max-Theorem, von Neumann (1928))

$$\gamma^* := \min_{\mathbf{d}} \max_{m=1,\ldots,M} \sum_{n=1}^N d_n y_n h_m(\mathbf{x}_n) \quad = \quad \max_{\alpha} \min_{n=1,\ldots,N} y_n \sum_{m=1}^M \alpha_m h_m(\mathbf{x}_n) =: \rho^*, \tag{2}$$

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

Thus, the minimum edge γ^* that can be achieved over all possible distributions $\mathbf{d}$ of the training set is equal to the maximum margin ρ^* of any linear combination of hypotheses from $\mathcal{H}$. Also, for any non-optimal distributions $\mathbf{d}$ and and hypothesis weights α we always have

$$\max_{h\in\mathcal{H}} \gamma_h(\mathbf{d}) \quad > \quad \gamma^* = \rho^* \quad > \quad \min_{n=1,\ldots,N} y_n f_\alpha(\mathbf{x}_n).$$

In particular, if the weak learning algorithm is guaranteed to return a hypothesis with edge at least γ for any distribution on the examples, then $\gamma^* \geq \gamma$ and by the above duality there exists a combined hypothesis with margin at least γ. If γ is equal to its upper bound γ^* then there exists a combined hypothesis with margin exactly $\gamma = \rho^*$ that only uses hypotheses that are actually returned by the weak learner in response to certain distributions on the examples.

From this discussion we can derive a sufficient condition on the weak learning algorithm to reach the maximum margin (for the case when $\mathcal{H}$ finite). If the weak learner returns hypotheses whose edges are at least γ^*, then there exists a linear combination of these hypotheses that attains a margin $\gamma^* = \rho^*$. We will prove later that our AdaBoost$_\nu^*$ algorithm efficiently finds a linear combination with margin close to ρ^* (cf. Theorem 6).

Constraining the edges of the previous hypotheses to equal zero (as done in the *totally corrective algorithm* of Kivinen and Warmuth (1999)) leads to a problem if there is no solution satisfying these constraints. At the end of trial t, the set of previous hypotheses is $\mathcal{H}_t = \{h_1, \ldots, h_t\}$ and the totally corrective algorithm finds a distribution such that $\gamma_h(\mathbf{d}) = 0$, for all $h \in \mathcal{H}_t$. Because of the above duality and the fact that $\mathcal{H}_t \subseteq \mathcal{H}$,

$$\gamma_t^* := \min_{\mathbf{d}} \max_{h\in\mathcal{H}_t} \gamma_h(\mathbf{d}) \quad \leq \quad \gamma^* = \rho^* \ .$$

The non-decreasing sequence (γ_t^*) converges to ρ^* from below. If $\rho^* > 0$, then the equality constraints on the edges are not satisfiable as soon as $\gamma_t^* > 0$.

In contrast our new algorithm AdaBoost$_\nu^*$ is motivated by a system of inequality constraints $\gamma_h(\mathbf{d}) \leq \rho$, for $h \in H_t$, where ρ is adapted. Again, if $\rho < \rho^*$, then the system of inequalities with this

4. This is a zero-sum game with payoff matrix $y_n h_m(\mathbf{x}_n)$. The row player finds a mixture $\mathbf{d}$ over the rows/examples and the column player a mixture α over the column/hypotheses. Adding a row/example makes the minimax value of the game go down and adding a column/hypothesis makes it go up.

$\hat{\rho}$ may not have a solution (and the Lagrange multipliers may diverge to infinity). In AdaBoost$^*_\nu$ we start with ρ large and decrease it when necessary. As we shall see, the algorithm maintains a margin parameter ρ that is always at least $\rho^* - \nu$.

4.2 Convergence properties of AdaBoost$_\rho$

Let AdaBoost$_{\{\rho_t\}}$ denote the version of AdaBoost$_\rho$ that uses a time varying margin parameter ρ_t at iteration t. Thus in step 3 (d) of the algorithm, ρ is replaced by ρ_t. This extension will be necessary for the later analysis of AdaBoost$^*_\nu$. The sequences $\{\rho_t\}_{t=1}^T$ might be specified while running the algorithm. For instance, in the algorithm Arc-GV, Breiman (1999) chooses ρ_t as $\min_{n=1,\ldots,N} y_n f_{\alpha_{t-1}}(\mathbf{x}_n)$. Breiman (1999) showed that Arc-GV *asymptotically* converges to the maximum margin (see discussion in next section). In the following we answer the question how to best choose the sequence $\{\rho_t\}$ so as to optimize bounds on the fraction of examples which have a margin at most ρ.

Lemma 2 *For any $\rho \in [-1,1]$, the final hypothesis f_α of AdaBoost$_{\{\rho_t\}}$ satisfies the following inequality:*

$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}\left(y_n f_\alpha(\mathbf{x}_n) \le \rho\right) \quad \le \quad \left(\prod_{t=1}^{T} Z_t\right) \exp\left\{\sum_{t=1}^{T} \rho\alpha_t\right\} = \prod_{t=1}^{T} \exp\left\{\rho\alpha_t + \ln Z_t\right\} \tag{3}$$

where $Z_t = \sum_{n=1}^N d_n^t \exp\left(-\alpha_t y_n h_t(\mathbf{x}_n)\right)$ and $\alpha_t = \frac{1}{2}\ln\frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2}\ln\frac{1+\rho_t}{1-\rho_t}$.

The proof directly follows from a simple extension of Theorem 1 in Schapire and Singer (1999) (see also Schapire et al. (1998)).

We now use a lemma from Rätsch et al. (2001) to upper bound the right hand side (rhs) of the above inequality:

Lemma 3 *Let γ_t be the edge of h_t in the t-th iteration of AdaBoost$_{\{\rho_t\}}$. Assume $-1 \le \rho_t \le \gamma_t$. Then for all $\rho \in [-1,1]$,*

$$\exp\left\{\rho\alpha_t + \ln Z_t\right\} \le \exp\left(-\frac{1+\rho}{2}\ln\left(\frac{1+\rho_t}{1+\gamma_t}\right) - \frac{1-\rho}{2}\ln\left(\frac{1-\rho_t}{1-\gamma_t}\right)\right). \tag{4}$$

Note that this generalizes Theorem 5 of (Freund and Schapire, 1997) to the case when the target margin is not zero.

AdaBoost$_{\{\rho_t\}}$ makes progress, if the rhs of (4) is smaller than one. Suppose we would like to reach a margin ρ on all training examples, where we obviously need to assume $\rho \le \rho^*$. We can then ask which sequence of $\{\rho_t\}_{t=1}^T$ one should use to find such combined hypothesis in as few iterations as possible. The rhs of (4) can be rewritten as

$$\exp\left(\Delta_2(\rho,\rho_t) - \Delta_2(\rho,\gamma_t)\right),$$

where $\Delta_2(a,b) := \frac{1+a}{2}\ln\frac{1+a}{1+b} + \frac{1-a}{2}\ln\frac{1-a}{1-b}$ denotes the binary relative entropy between $a,b \in [-1,1]$. Therefore the rhs of (4) is minimized for $\rho_t = \rho$ (independent of γ_t) and one should always use this constant choice.

This means that when $\rho_t = \rho$ then the rhs of (4) is reduced by a factor of $\exp(-\Delta_2(\rho,\gamma_t))$, which can be upper bounded by inspecting the Taylor expansion at $\gamma_t = \rho$ and noticing that when $0 \le \rho < \gamma_t$, all terms of order three and higher are negative:

$$\exp(-\Delta_2(\rho,\gamma_t)) < 1 - \frac{1}{2}\frac{(\rho-\gamma_t)^2}{1-\rho^2}, \text{ for } 0 \le \rho < \gamma_t. \tag{5}$$

The denominator $1-\rho^2$ speeds up the convergence when $\rho \gg 0$. Notice that when $\rho = 0$, then we recover the original AdaBoost bound.

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

Corollary 4 *Assume the weak learner always returns a base hypothesis with an edge $\gamma_t \geq \rho^*$. If $0 \leq \rho \leq \rho^* - \nu$, $\nu > 0$, then AdaBoost$_\rho$ will converge to a solution with margin at least ρ on all examples in at most $\frac{2\ln N(1-\rho^2)}{\nu^2}$ iterations.*

Proof By Lemma 2 and (4), (5):

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

The margin is at least ρ for all examples, if the rhs is smaller than $\frac{1}{N}$; hence after at most

$$\frac{\ln N}{-\ln\left(1-\frac{1}{2}\frac{\nu^2}{1-\rho^2}\right)} \leq \frac{2\ln N(1-\rho^2)}{\nu^2}$$

iterations, which proves the statement. ∎

When $\rho < 0$, then inequality (5) can be replaced with the following weaker inequality which holds for all distinct $\rho, \gamma_t \in [-1,1]$:

$$\exp(-\Delta_2(\rho,\gamma_t)) < \exp\left(-\frac{1}{2}(\rho-\gamma_t)^2\right). \tag{6}$$

This leads to the same bound as in the above corollary except that the factor $(1-\rho^2)$ is omitted. Thus when $\rho < 0$, the bound on the number of iterations becomes $\frac{2\ln N}{\nu^2}$ (Rätsch, 2001, page 25).

4.3 Asymptotic Margin of AdaBoost$_\rho$

With the methods shown so far we can determine the asymptotic value of margin of the hypothesis produced by the original AdaBoost algorithm. First, we state a lower bound on the margin that is achieved by AdaBoost$_\rho$. There is a gap between this lower bound and the upper bound of Theorem 1. In a second part we consider an experiment that shows that depending on some subtle properties of the weak 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 quite tight in empirical cases.

As long as each factor in the rhs of Eq. (3) is smaller than 1, the bound decreases. If the factor is at most $1-\mu$ and $\mu > 0$, then the rhs converges exponentially fast to zero. The following corollary considers the asymptotic case and gives a lower bound on the margin.

Corollary 5 (Rätsch (2001)) *Assume AdaBoost$_\rho$ generates hypothesis $h_1, h_2, \ldots$ with edges γ_1, γ_2, $\ldots$ and coefficients $\alpha_1, \alpha_2, \ldots$. Let $\gamma^{\min} = \inf_{t=1,2,\ldots}\gamma_t$ and assume $\gamma^{\min} > \rho$. Furthermore, let*

$$\hat{\rho}_t = \min_{n=1,\ldots,N} \frac{y_n \sum_{r=1}^{t}\alpha_r h_r(\mathbf{x}_n)}{\sum_{r=1}^{t}\alpha_r}$$

be the achieved margin in the t-th iteration and $\hat{\rho} = \sup_{t=1,2,\ldots} \hat{\rho}_t$. *Then the margin* $\hat{\rho}$ *of the combined hypothesis is bounded from below by*

$$\hat{\rho} \geq \frac{\ln(1-\rho^2) - \ln(1-(\gamma^{\min})^2)}{\ln\left(\frac{1+\gamma^{\min}}{1-\gamma^{\min}}\right) - \ln\left(\frac{1+\rho}{1-\rho}\right)}. \tag{7}$$

From (7) one can understand the interaction between ρ and $\gamma^{\min}$: If the difference between $\gamma^{\min}$ and ρ is small, then the rhs of (7) is small. Thus, if ρ with $\rho \leq \gamma^{\min}$ is large, then $\hat{\rho}$ must be large, i.e. choosing a larger ρ results in a larger margin on the training examples. A Taylor expansion of the rhs of (7) shows that the margin is lower bounded by $\frac{\gamma^{\min}+\rho}{2}$. This known lower bound (Breiman, 1999, Theorem 7.2) is greater than ρ if $\gamma^{\min} > \rho$.

In Section 4.1 we reasoned that $\gamma^{\min} \leq \rho^*$. If the parameter AdaBoost$_\rho$ is chosen too small, then we guarantee only that the margin of the produced linear combination converges asymptotically to a value at below ρ^*. In the original formulation of AdaBoost we have $\rho = 0$ and we guarantee only that AdaBoost$_0$ achieves a margin of at least $\frac{\gamma^{\min}+\rho}{2} = \frac{1}{2}\gamma^{\min}$. This shortfall in the margin provable for AdaBoost motivates our new AdaBoost$^*_\nu$ which is guaranteed to optimize the margin.

4.3.1 EXPERIMENTAL ILLUSTRATION OF COROLLARY 5

To illustrate the above-mentioned gap, we perform an experiment showing how tight (7) can be. We analyze two different settings: (i) the weak learner selects the hypothesis with largest edge over all hypotheses (i.e. the best case) and (ii) the weak learner selects the hypothesis with minimum edge among all hypotheses with edge larger than ρ^* (i.e. the worst case). Corollary 5 holds for both cases since the weak learner is allowed to return *any* hypothesis with edge larger than ρ^*.

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 random hypotheses with range $\{+1,-1\}$. We first choose a parameter p uniformly in (0,1). Then the label of each hypothesis on each example is chosen to agree with the label of the example with probability p.[5] First we compute the solution ρ^* of the margin-LP problem via the left hand side of (2). Then we compute the combined hypothesis generated by AdaBoost$_\rho$ after 10^4 iterations for $\rho = 0$ and $\rho = \frac{1}{3}$ using the best and the worst selection strategy, respectively. The latter algorithm depends on ρ^*. We chose 300 hypothesis sets based on 300 random draws of p. The random choice of p ensures that there are cases with small and large optimal margins. For each hypothesis set we did two runs of AdaBoost$_\rho$ using the best and worst selection strategies. The result of each run is represented as a point in Figure 1. The abscissa is the maximum achievable margin ρ^* for each run. The ordinate is the margin of AdaBoost$_\rho$ using the best (green) and the worst strategy (red).

There is a large difference between these selection strategies. Whereas the margin of the worst strategy is *tightly* lower bounded by (7), the best strategy has near maximum margin. These experiments show that one obtains different results by changing the selection strategy of the weak learning algorithm. Our lower bound holds for both selection strategies. The looseness of the bounds is indeed a problem, as we cannot predict where AdaBoost$_\rho$ converges to.[6] However, note that moving $\hat{\rho}$ closer to ρ^* reduces the gap (see also Figure 1 [right]).

5. We do not allow duplicate hypotheses or hypotheses that agree with the labels on all examples.
6. One might even be able to construct cases where the outputs are not at all converging.

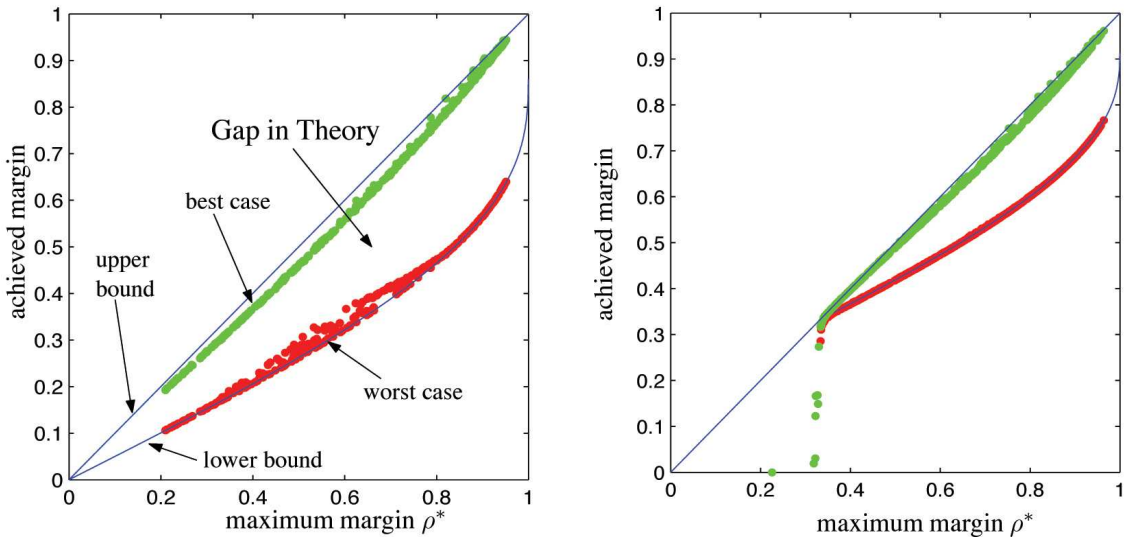

Figure 1: Achieved margins of AdaBoost$_\rho$ using the best (green) and the worst (red) selection on random data for $\rho = 0$ [left] and $\rho = \frac{1}{3}$ [right]. On the abscissa is the maximum achievable margin ρ^* and on the ordinate the margin achieved by AdaBoost$_\rho$ for one data realization. For comparison we plotted the upper bound $y = x$ and the lower bound (7). On the interval $[\rho, 1]$, there is a clear gap between the performance of the worst and best selection strategies. The margin of the worst strategy is very close to the lower bound (7) and the best strategy has near maximum margin. If ρ is chosen slightly below the maximum achievable margin then this gap is reduced to 0.

Recently, it has been shown by Rudin et al. (2005) that there exist cases where the weighting $\mathbf{d}^t$ on the examples cycles indefinitely between non-optimal solutions. This proves that AdaBoost does not generally maximize the margin. Furthermore, it was shown in Rudin et al. (2004b) that the gap exhibited in Figure 1 is not an experimental artifact: under certain conditions the lower bound (7) was proven to be tight.

4.3.2 DECREASING THE STEP SIZE

Breiman (1999) conjectured that the inability to maximize the margin is due to the fact that the normalized hypothesis coefficients may "circulate endlessly through the convex set", which is defined by the lower bound on the margin. In fact, motivated from our previous experiments, it seems possible to implement a weak learner that appropriately switches between optimal and worst case performance, leading to non-convergent normalized hypothesis coefficients.

Rosset et al. (2002) have shown that AdaBoost with infinitesimally small step sizes may maximize the margin, if the weak learner uses the best selection strategy. This is similar to what we found empirically for finite step sizes and motivates us to analyze AdaBoost$_\rho$ with step sizes chosen as follows:

$$\hat{\alpha}_t = \eta \alpha_t = \frac{\eta}{2} \ln \frac{1+\gamma_t}{1-\gamma_t} - \frac{\eta}{2} \ln \frac{1+\rho}{1-\rho},$$

for some $\eta > 0$. For $\eta = 1$ we recover AdaBoost$_\rho$. Following the same proof technique as for Corollary 5, we can show that under the same conditions as given in Corollary 5

$$\hat{\rho} \geq \frac{-\ln\left((1+\gamma)\exp(-\hat{\alpha}) + (1-\gamma)\exp(\hat{\alpha})\right)}{\hat{\alpha}},$$

where $\hat{\alpha} = \frac{\eta}{2}\ln\frac{1+\gamma}{1-\gamma} - \frac{\eta}{2}\ln\frac{1+\rho}{1-\rho}$. Note that if η goes to zero, then $\hat{\rho} = \gamma$. Interestingly, this is independent of the choice of ρ. Thus if the weak learner always returns hypotheses with edges $\gamma_t \geq \rho^*$ ($t = 1, 2, \ldots$), where ρ^* is the maximum margin, then by the Min-Max Theorem, the margin is maximized when η goes to zero. However, there are no guarantees on the convergence speed.

4.4 Convergence of AdaBoost$^*_\nu$

The AdaBoost$^*_\nu$ algorithm is based on two insights:

- According to the discussion after Lemma 3, the most rapid convergence to a combined hypothesis with margin $\rho^* - \nu$ occurs for AdaBoost$_\rho$ when one chooses ρ_t as close as possible to $\rho^* - \nu$.
- For distributions on the examples that are hard for the weak learner (i.e. the weak learner achieves a small edge), the edge γ_t will be close to ρ^*.

The idea is that by choosing $\rho_t = (\min_{r=1,\ldots,t}\gamma_t) - \nu$ we concentrate on the hardest distribution we generated so far and can so find a *close* overestimate of $\rho^* - \nu$. This forces an acceleration of the convergence to a large margin and leads to distributions for which the weak learner has to return small edges.

Note that if the weak learner always returns hypotheses with edge $\gamma_t = \rho^*$ which is the worst case under the assumption that $\gamma_t \geq \rho^*$, then $\rho_t = \rho^* - \nu$ in each iteration. In this case the same smallest step size is taken in every iteration which is determined by ρ^* and ν. This smallest step size decreases with the desired accuracy ν, which matches the intuition from Section 4.3.2 that decreasing the step size achieves larger and therefore more accurate margins.

We will now state and prove our main theorem:

Theorem 6 *Assume the weak learner always returns a base hypothesis with an edge $\gamma_t \geq \rho^*$. Then after $\frac{2\ln N}{\nu^2}$ iterations AdaBoost$^*_\nu$ (Algorithm 2) is guaranteed to produce a combined hypothesis f of margin at least $\rho^* - \nu$.*

Proof Let $\rho = \rho^* - \nu$ be the margin that we would like to achieve. By assumption on the performance of the weak learner, $\rho^* \leq \min_{r=1,\ldots,T}\gamma_r = \gamma_T^{\min}$ and thus $\rho = \rho^* - \nu \leq \gamma_T^{\min} - \nu$. In step 3 (d) of Algorithm 2, ρ_t was set to $\gamma_t^{\min} - \nu$. Hence $\rho \leq \rho_t$ for each iteration.

Lemmas 2 and 3 imply that

$$\frac{1}{N}\sum_{n=1}^{N}\mathbf{I}\left(y_n f(\mathbf{x}_n) \leq \rho\right) \leq \prod_{t=1}^{T}\exp\left(-\frac{1+\rho}{2}\ln\left(\frac{1+\rho_t}{1+\gamma_t}\right) - \frac{1-\rho}{2}\ln\left(\frac{1-\rho_t}{1-\gamma_t}\right)\right)$$

We now rewrite the rhs using $\alpha_t = \frac{1}{2}\ln\frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2}\ln\frac{1+\rho_t}{1-\rho_t}$:

$$= \prod_{t=1}^{T}\exp\left(-\frac{1}{2}\ln\left(\frac{1+\rho_t}{1+\gamma_t}\right) - \frac{1}{2}\ln\left(\frac{1-\rho_t}{1-\gamma_t}\right) + \rho\alpha_t\right)$$

By (1), $\alpha_t \geq 0$ since $\rho_t \leq \gamma_t$. By replacing ρ by its upper bound ρ_t we get:

$$\leq \prod_{t=1}^{T} \exp\left(-\frac{1+\rho_t}{2}\ln\left(\frac{1+\rho_t}{1+\gamma_t}\right) - \frac{1-\rho_t}{2}\ln\left(\frac{1-\rho_t}{1-\gamma_t}\right)\right)$$

Finally, by (6) we have:

$$= \prod_{t=1}^{T} \exp(-\Delta(\rho_t,\gamma_t)) < \prod_{t=1}^{T} \exp(-\frac{(\rho_t-\gamma_t)^2}{2}) \leq \exp(-\frac{T\nu^2}{2}).$$

is at most $\frac{1}{N}$, then by the above chain of inequalities, $\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}(y_n f(\mathbf{x}_n) \leq \rho) < \frac{1}{N}$ and the margin of each of the N examples is at least ρ. The theorem now follows from the fact that $\frac{1}{N} < \exp\left(-\frac{1}{2}T\nu^2\right)$, if the number of iterations T is at least $\frac{2\ln N}{\nu^2}$. ■

If one assumes $\rho_t \geq 0$, then Theorem 6 could be improved by a factor of $(1-\rho_t^2)$ in each iteration, using the refined upper bound of Corollary 4. Since $\rho_t \geq \rho^* - \nu$, one would obtain the bound $\frac{\ln N(1-(\rho^*-\nu)^2)}{\nu^2}$ if $\rho^* \geq \nu$, but this factor will only matter for very large margins.

4.5 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 (Rätsch et al., 2002). For *infinite hypothesis sets*, Theorem 1 can be restated in a weaker form as:

Theorem 7 (Weak Min-Max, e.g. Nash and Sofer (1996))

$$\gamma^* := \min_{\mathbf{d}} \sup_{h\in\mathcal{H}} \sum_{n=1}^{N} y_n h(\mathbf{x}_n) d_n \quad \geq \quad \sup_{\alpha} \min_{n=1,\ldots,N} y_n \sum_{q:\alpha_q>0} \alpha_q h_q(\mathbf{x}_n) =: \rho^*, \tag{8}$$

where $\mathbf{d} \in \mathcal{P}^N$, $\alpha \in \mathcal{P}^{|\mathcal{H}|}$ *with finite support.*

We call $\Gamma = \gamma^* - \rho^*$ the "duality gap". In particular for any $\mathbf{d} \in \mathcal{P}^N$, $\sup_{h\in\mathcal{H}} \sum_{n=1}^{N} y_n h(\mathbf{x}_n) d_n \geq \gamma^*$ and for any $\alpha \in P^{|\mathcal{H}|}$ with finite support, $\min_{n=1,\ldots,N} y_n \sum_{q:\alpha_q\geq 0} \alpha_t h_q(\mathbf{x}_n) \leq \rho^*$.

In theory the duality gap may be nonzero. However, Lemma 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 weak learning algorithm can return a hypothesis in each iteration that has an edge not smaller than ρ^*.

In other words, the duality gap may result from the fact that the sup on the left side cannot 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 weak learner is always able to pick good enough hypotheses ($\geq \rho^*$), one automatically gets by Lemma 3 that $\Gamma = 0$.

Under certain conditions on $\mathcal{H}$ this maximum always exists and strong duality holds (for details see e.g. Rätsch et al., 2002; Rätsch, 2001; Hettich and Kortanek, 1993; Nash and Sofer, 1996):

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

In general, this requirement can be fulfilled by the weak 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$_\rho$). The first requirement might be a problem with weak learning algorithms that are 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 $\mathcal{H}$ 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 weak learning algorithm for distribution $\mathbf{d}$ and training sample S (Rätsch, 2001). This procedure assures that $\mathcal{H}$ is closed.

The above theorem is applied in Appendix B to obtain iteration bounds for AdaBoost$_\nu^*$ in the context of learning a convex combination of support vector kernels.

5. Experimental Comparison

In this section we discuss two experiments: The first one shows that our theoretical bounds can be tight on artificial data and the second one compares our algorithm to the one proposed in Rudin et al. (2004a).

5.1 Illustration on Toy Examples

We are aware that maximizing the margin of the ensemble does not lead to improved generalization performance in all cases. In fact for fairly noisy data sets the opposite has been reported (cf. Quinlan, 1996; Breiman, 1999; Grove and Schuurmans, 1998; Rätsch et al., 2001). Also, Breiman (1998) reported an example where the margins of all examples are larger in one ensemble than another and the latter generalized considerably better.

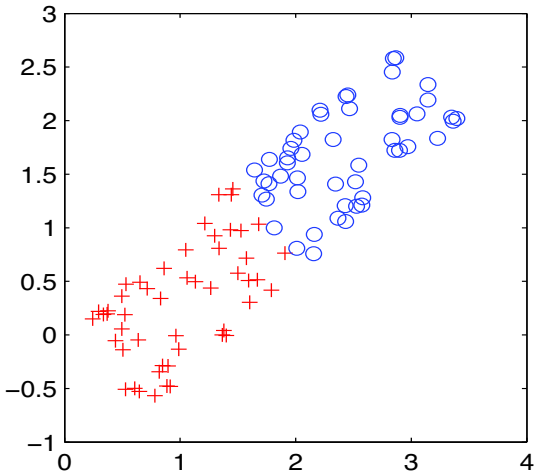

Figure 2: The two *discriminative dimensions* of our separable one hundred dimensional data set.

Nonetheless, the theoretical bounds on the generalization error of linear classifiers improves with the margin. We therefore expect to be able to measure differences in the generalization error between a function that maximizes the margin and one that does not. Similar results have been obtained in Schapire et al. (1998) on a multi-class optical character recognition problem.

Here we report experiments on artificial data to illustrate how our algorithm works and 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 which means that controlling the capacity of the ensemble is essential.

As the weak learning algorithm we use C4.5 decision trees provided by Quinlan (1992) using an option to control the number of nodes in the tree. We have tuned C4.5 to generate trees with about three nodes. Otherwise, the weak learner often classifies all training examples correctly and over-fits the data already. Furthermore, since in this case the margin is already maximum (equal to 1), boosting algorithms would stop since $\gamma = 1$. We therefore need to limit the complexity of the weak learner, in good agreement with the bounds on the generalization error (Schapire et al., 1998).

Moreover, we have to deal with the fact that C4.5 cannot use weighted samples. We therefore use weighted bootstrapping (e.g. Efron and Tibshirani, 1994). However, this amplifies the problem that the resulting hypotheses might in some cases have an edge smaller than the maximum margin, which according to the Min-Max-Theorem should not occur if the weak learner performs optimally. We deal with this problem by repeatedly calling C4.5 on different bootstrap realizations if the edge is smaller than the margin of the current linear combination. Furthermore, for AdaBoost$^*_\nu$, a small edge of one hypothesis can spoil the margin estimate ρ_t. We address this problem by resetting $\rho_t = \hat{\rho}_t + \nu$, whenever $\rho_t \leq \hat{\rho}_t$, where $\hat{\rho}_t$ is the margin of the currently combined hypothesis.

In Figure 3 we see a typical run of AdaBoost, Marginal AdaBoost, AdaBoost$^*_\nu$ and Arc-GV for $\nu = .1$. For comparison we plot the margins of the hypotheses generated by AdaBoost (cf. Figure 3 (left)). One observes that it is not able to achieve a large margin efficiently. After 1000 iterations $\hat{\rho} = .37$.

Marginal AdaBoost as proposed in Rätsch and Warmuth (2002) proceeds in stages and first tries to find an estimate of the margin using a binary search. It calls AdaBoost$_\rho$ three times. The first call of AdaBoost$_\rho$ for $\rho = 0$ stops after four iterations because it has generated a consistent combined hypothesis. The lower bound l on ρ^* as computed by Marginal AdaBoost is $l = .07$ and the upper bound u is .94. The second time ρ is chosen to be in the middle of the interval $[l,u]$ and AdaBoost$_\rho$ reaches the margin of $\rho = .51$ after 80 iterations. The interval is now $[.51,.77]$. Because the length of the interval $u - l = .27$ is small enough, Marginal AdaBoost leaves the loop through an exit condition, calls AdaBoost$_\rho$ the last time for $\rho = u - \nu = .41$ and finally achieves the margin of .55.

In a run of Arc-GV for thousand iterations we observe a margin of the combined hypothesis of .53, while for our new algorithm, AdaBoost$^*_\nu$, we find .58. In this case the margin for AdaBoost$^*_\nu$ is larger than the margins of all other algorithms when executed for one thousand iterations. It starts with slightly lower margins in the beginning, but then catches up due the better choice of the margin estimate.

	C4.5	AdaBoost	Marginal AdaBoost	AdaBoost$^*_\nu$
E_{gen}	$7.4 \pm .11\%$	$4.0 \pm .11\%$	$3.6 \pm .10\%$	$3.5 \pm .10\%$
$\hat{\rho}$	—	$.31 \pm .01$	$.55 \pm .01$	$.58 \pm .01$

Table 2: Estimated generalization performances and margins with confidence intervals for decision trees (C4.5), AdaBoost, Marginal AdaBoost and AdaBoost$^*_\nu$ on the toy data. All numbers are averaged over 200 splits into 100 training and 19900 test examples.

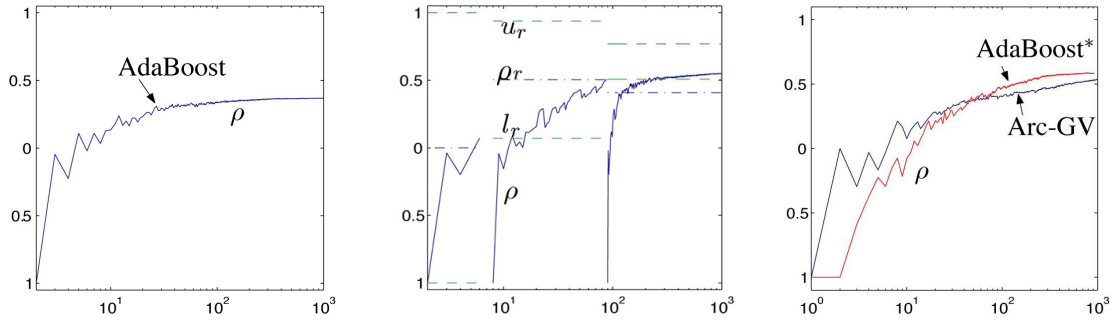

Figure 3: Illustration of the achieved margin of AdaBoost$_0$ (left), Marginal AdaBoost (middle), Arc-GV, and AdaBoost$^*_\nu$ (right) at each iteration. Marginal AdaBoost calls AdaBoost$_\rho$ three times while adapting ρ (dash-dotted). We also plot the values for l and u as in Marginal AdaBoost (dashed). (For details see Rätsch and Warmuth, 2002) AdaBoost$^*_\nu$ achieves larger margins than AdaBoost. Compared to Arc-GV it starts slower, but then catches up in the later iterations. Here the correct choice of the parameter ρ is important.

In Table 2 we see the average performances of the four classifiers. For AdaBoost and AdaBoost$^*_\nu$ we combined 200 hypotheses for the final prediction. For Marginal AdaBoost we use $\nu = .1$ and let the algorithm combine only 200 hypotheses for the final prediction to get a more fair comparison. We see a large improvement of all ensemble methods over 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 AdaBoost and Marginal AdaBoost as well as AdaBoost and AdaBoost$^*_\nu$. Note also that the margins of the combined hypothesis achieved by Marginal AdaBoost and AdaBoost$^*_\nu$ are on average almost twice as large as for AdaBoost. The difference in generalization performance between AdaBoost$^*_\nu$ and Marginal AdaBoost is not statistically significant.

The differences between the achieved margins of both algorithms seem slightly significant (96%). The slightly larger margins generated by Marginal AdaBoost can be attributed to the fact that it uses many more calls to the weak learner than AdaBoost$^*_\nu$ and after an estimate of the achievable margin is available, it starts optimizing the linear combination using this estimate.

It would be natural to use a two-pass algorithm: In the first pass use AdaBoost$^*_\nu$ to get a margin estimate ρ size at least $\rho^* - \nu$ and then use this estimate in a final run of AdaBoost$_\rho$. The hypothesis produced in the second pass should have a larger margin and use fewer base hypotheses.

5.2 Heuristics for Tuning the Precision Parameter ν

Our main results says that after $\frac{2\ln N}{\nu^2}$ iterations AdaBoost$^*_\nu$ produces a hypothesis of margin at least $\rho^* - \nu$. Thus if the algorithm is allowed to run for T iterations, then ν should be set to $\nu_T = \sqrt{\frac{2\ln N}{T}}$. If ν is chosen much larger than ν_T, then after T iterations AdaBoost$^*_\nu$ often achieves a margin below $\rho^* - \nu_T$. Similarly, if ν is chosen much smaller than ν_T, then AdaBoost$^*_\nu$ starts too slowly and after T iterations its margin is typically again below $\rho^* - \nu_T$.

Recently, Rudin et al. (2004a,c) proposed an algorithm, called *Coordinate Ascent Boosting*, which solves the same problem as AdaBoost$^*_\nu$. Their analysis of the algorithm shows that it needs

at most $\Omega(\nu^{-3})$ iterations to achieve a margin of at least $\rho^* - \nu$. While this theoretical result is clearly inferior to the guarantees which we provide for AdaBoost$^*_\nu$, their experimental evaluation of the algorithms seemed to suggest that the algorithm requires significantly fewer iterations than AdaBoost$^*_\nu$ in practice. However, their observations were only due to the improper choice of the accuracy parameter ν for AdaBoost$^*_\nu$: For $\nu = 10^{-3}$ (as chosen in their study), AdaBoost$^*_\nu$ would need millions of iterations to achieve a guaranteed margin $\rho^* - \nu$. However, only the first $20K$ iterations were displayed and in this range their algorithms achieve a larger margin. For $T = 20K$ and $N = 50$, the precision parameter prescribed by our bounds is $\nu_T = .02$. When this parameter is used, then AdaBoost* clearly beats all the other related algorithms (cf. Figure 4). We leave it to the reader to explore other heuristics for tuning ν based on the theoretical results of this paper (See also the discussion at the end of the last subsection).

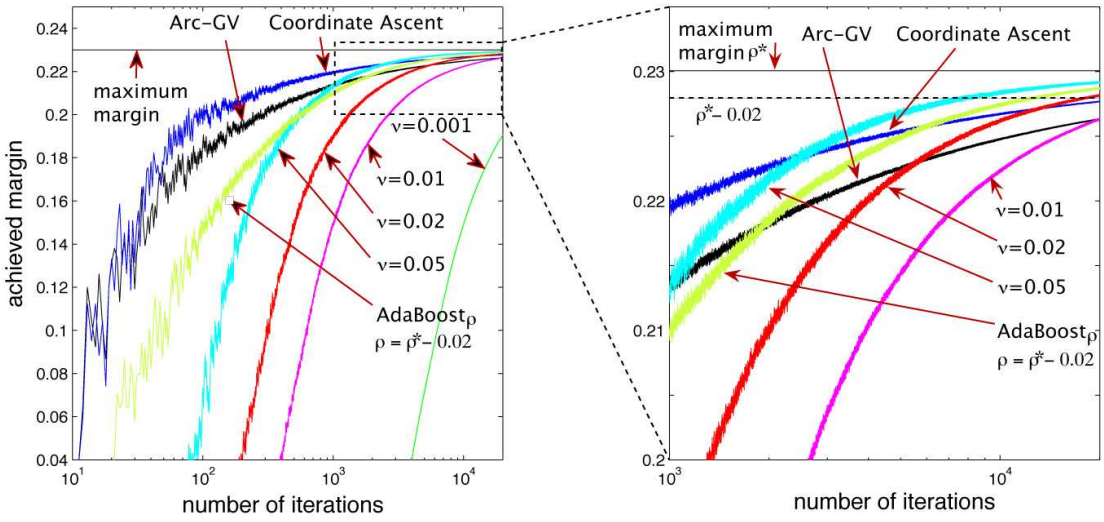

Figure 4: AdaBoost$^*_\nu$ with different choices of ν is compared to Arc-GV and the Coordinate Ascent Algorithm on the same artificial dataset 1 used in Rudin et al. (2004c) (We reconstructed this dataset from a figure given in Rudin et al. (2004b)): The number of iterations is $T = 20K$, the dimension of the examples is $N = 50$, and we assume that the base learner returns a hypothesis with maximum edge. If ν is set to a reasonably close range around the value $\nu_T = .02$ prescribed by our bound, then AdaBoost$^*_\nu$ achieves the margin which is significantly larger than the margins achieved by the other algorithms. If $\nu = .001 \ll \nu_T$ as chosen in Rudin et al. (2004c), then AdaBoost$^*_\nu$ starts too slowly. In the case when the base learner returns a random hypothesis with edge only at least as large as ρ^*, then our algorithm compares even more favorably (not shown).

6. Conclusion

We have analyzed a generalized version of AdaBoost in the context of large margin algorithms. From von Neumann's Min-Max theorem we know that if the weak learner always returns a hypothesis with weighted classification error less than $\frac{1}{2} - \frac{1}{2}\gamma$ then the maximum achievable margin ρ^* is at least γ. The asymptotic analysis lead us to a lower bound on the margin of the final hypotheses generated by AdaBoost$_\rho$, which was shown to be rather tight in empirical cases. Our results indicate that vanilla AdaBoost generally does not maximize the margin, and only achieves a margin of about half the optimum.

To overcome these problems we provided an algorithm AdaBoost$^*_\nu$ with the following provable guarantees: It produces a linear combination with margin at least $\rho^* - \nu$ and the number of base hypotheses used in this linear combination is at most $\frac{2\ln n}{\nu^2}$. The new algorithm decreases its estimate ρ of the margin iteratively, such that the gap between the best and the worst case becomes arbitrarily small. Our analysis did not require additional properties of the weak learning algorithm. In simulation experiments we have illustrated the validity of our theoretical analysis.

Appendix A. Margins

First recall the definition of margin used in this paper, which is defined for a fixed set of examples $\{(\mathbf{x}_n, y_n) : 1 \le n \le N\}$ and a set of hypotheses $\mathcal{H} = \{h_1, \ldots, h_M\}$ (here finite for the sake of simplicity):

$$\rho^*(\mathcal{H}) = \max_{\alpha} \min_{n=1,\ldots,N} y_n \sum_{m=1}^{M} \alpha_m h_m(\mathbf{x}_n), \text{ where } \alpha \text{ is on the simplex } \mathcal{P}^M.$$

Note that we minimize over the margins of individual examples and maximize over the hyperplanes. Define the one-norm margin $\rho_1^*(\mathcal{H})$ in the same way but now α lies in the larger set $\{\alpha : \alpha \in \mathbb{R}^M \text{ and } ||\alpha||_1 = 1\}$. It is well known that for a fixed example $(\mathbf{x}_n, y_n)$ and normal $\alpha \in \mathbb{R}^M$, the one-norm margin $\frac{\sum_{m=1}^M \alpha_m h_m(\mathbf{x}_n)}{\sum_m^M |\alpha_m|}$ is the minimum ℓ_∞-distance of the example to the hyperplane with normal α (Mangasarian, 1999; Rätsch et al., 2002), where the latter distance is defined as

$$\inf_{\mathbf{z} \in \mathbb{R}^M \text{ s.t. } \alpha \cdot \mathbf{z} = \mathbf{0}} y_n \max_{m=1,\ldots,M} |h_m(\mathbf{x}_n) - z_m|.$$

Note that in this appendix, margins are defined as a function of the the hypotheses set $\mathcal{H}$ because we will vary this set in a moment. Let $\mathrm{cl}(\mathcal{H})$ be the closure of $\mathcal{H}$ under negation, i.e. $\mathrm{cl}(\mathcal{H}) = \mathcal{H} \cup \{-h : h \in \mathcal{H}\}$. Now, the following relationships are straightforward:

1. $\rho^*(\mathcal{H}) \le \rho_1^*(\mathcal{H})$, $\rho^*(\mathrm{cl}(\mathcal{H})) \ge 0$, and $\rho^*(\mathrm{cl}(\mathcal{H})) \ge \rho_1^*(\mathcal{H})$.
2. If $\rho^*(\mathrm{cl}(\mathcal{H})) > 0$, then $\rho^*(\mathrm{cl}(\mathcal{H})) = \rho_1^*(\mathcal{H})$.
3. If $\rho_1^*(\mathcal{H}) \ge 0$, then $\rho^*(\mathrm{cl}(\mathcal{H})) = \rho_1^*(\mathcal{H})$.

In summary, if the one-norm margin of $\mathcal{H}$ is non-negative, then the margin of the closed hypotheses class $\mathrm{cl}(\mathcal{H})$ coincides with the one-norm margin.

Appendix B. An Application to Multiple Kernel Learning

Sonnenburg et al. (2005) proposed a new algorithm for solving the multiple kernel learning (MKL) problem that was introduced in Lanckriet et al. (2004); Bach et al. (2004). The idea of MKL is to find a convex combination of J support vector kernels $k_j : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ ($j = 1, \ldots, J$) that maximizes the SVM soft margin (cf. Bach et al. (2004)). In Sonnenburg et al. (2005) the original quadratically-constraint quadratic program was reformulated to the following semi-infinite linear program:

$$\min_{\beta \in \mathcal{P}^J} \sup_{\alpha \in \mathcal{A}} \sum_{j=1}^{J} \beta_j S_j(\alpha) \tag{9}$$

where

$$\begin{aligned} S_j(\alpha) &:= -\frac{1}{2} \sum_{r,s=1}^{N} \alpha_r \alpha_s y_r y_s k_j(\mathbf{x}_r, \mathbf{x}_s) + \sum_{n=1}^{N} \alpha_n \\ \mathcal{A} &:= \left\{ \alpha \;\middle|\; \alpha \in \mathbb{R}^N, \mathbf{0} \le \alpha \le \mathbf{1}C, \sum_{n=1}^{N} y_n \alpha_n = 0 \right\} \end{aligned}$$

and C is the SVM regularization constant. Note that this problem has infinitely many constraints: one for every vector α in its domain $\mathcal{A}$. Note that problem (9) is of the same type as the semi-infinite programming problem (8) which can be solved with AdaBoost$^*_\nu$ (cf. discussion in Section 4.5). Since the $S_j(\alpha)$ are continuous functions and $\mathcal{A}$ is compact, it follows from Theorem 8 that the duality gap is zero.

When AdaBoost$*_\nu$ is applied to this problem, a hypothesis with large edge has to be found in each iteration. In this case the hypotheses are α vectors and the edge is

$$\sum_{j=1}^{J} \beta_j S_j(\alpha) = -\frac{1}{2} \sum_{r,s} \alpha_r \alpha_s y_r y_s \left(\sum_{j=1}^{J} \beta_j k_j(\mathbf{x}_r, \mathbf{x}_s) \right) + \sum_i \alpha_i.$$

It has been noted that the edge in this case is nothing else than the negative SVM objective function for the combined kernel $k(\mathbf{x}_r, \mathbf{x}_s) = \sum_{j=1}^{J} \beta_j k_j(\mathbf{x}_r, \mathbf{x}_s)$. Hence, identifying an α vector with maximum edge amounts to solving the vanilla SVM quadratic optimization problem. Fortunately, many efficient SVM packages are available to solve this problem. Thus, the MKL problem can be efficiently solved using AdaBoost$^*_\nu$ and our iteration bound for AdaBoost$^*_\nu$ is applicable.

References

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Twenty-first international conference on Machine learning*. ACM Press, 2004. ISBN 1-58113-828-5.

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. Morgan Kaufmann.

L. Breiman. Are margins relevant in voting? Talk at the NIPS'98 workshop on Large Margin Classifiers, December 1998.

L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11(7):1493–1518, 1999. Also Technical Report 504, Statistics Department, University of California Berkeley, Dec. 1997.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistic and Applied Probability*. Chapman and Hall/CRC, New York, 1994.

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

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.

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.

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

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

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

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.

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

J. Lafferty. Additive models, boosting, and inference for generalized divergences. In *Proc. 12th Annu. Conf. on Comput. Learning Theory*, pages 125–133, New York, NY, 1999. ACM Press.

G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 2004.

O. L. Mangasarian. Arbitrary-norm separating plane. *Operation Research Letters*, 24(1):15–23, 1999.

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

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

J. R. Quinlan. Boosting first-order learning. *Lecture Notes in Computer Science*, 1160:143, 1996.

G. Rätsch. *Robust Boosting via Convex Optimization*. PhD thesis, University of Potsdam, Neues Palais 10, 14469 Potsdam, Germany, October 2001.

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 NC-TR-2000-085.

G. Rätsch, S. Mika, B. Schölkopf, and K.-R. Müller. Constructing boosting algorithms from SVMs: an application to one-class classification. *IEEE PAMI*, 24(9), September 2002. In press. Earlier version is GMD TechReport No. 119, 2000.

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.

G. Rätsch and M. K. Warmuth. Maximizing the margin with boosting. In *Proc. COLT*, volume 2375 of *LNAI*, pages 319–333, Sydney, 2002. Springer.

S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin separator. Technical report, Department of Statistics, Stanford University, 2002.

C. Rudin, I. Daubechies, and R. E. Schapire. On the dynamics of boosting. In *Advances in Neural Information Processing*, volume 15, 2004a.

C. Rudin, I. Daubechies, and R. E. Schapire. The dynamics of AdaBoost: Cyclic behavior and convergence of margins. *Journal of Machine Learning Research*, 2005.

C. Rudin, R. E. Schapire, and I. Daubechies. Analysis of boosting algoritms using the smooth margin function: A study of three algorithms. Unpublished manuscript, October 2004b.

C. Rudin, R. E. Schapire, and I. Daubechies. Boosting based on a smooth margin. In *Proc. COLT'04*, LNCS. Springer Verlag, July 2004c.

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

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

R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, December 1999.

S. Sonnenburg, G. Rätsch, and C. Schäfer. Learning interpretable svms for biological sequence analysis. In *Proc. RECOMB'05*, LNCB. Springer Verlag, 2005.

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

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