Speed and Sparsity of Regularized Boosting

Yongxin Taylor Xi **Zhen James Xiang** **Peter J. Ramadge**
Princeton University
Department of Electrical Engineering
Princeton, NJ 08544

Robert E. Schapire
Princeton University
Department of Computer Science
35 Olden Street, Princeton, NJ 08540

Abstract

Boosting algorithms with l_1-regularization are of interest because l_1 regularization leads to sparser composite classifiers. Moreover, Rosset et al. have shown that for separable data, standard l_p-regularized loss minimization results in a margin maximizing classifier in the limit as regularization is relaxed. For the case $p = 1$, we extend these results by obtaining explicit convergence bounds on the regularization required to yield a margin within prescribed accuracy of the maximum achievable margin. We derive similar rates of convergence for the ε-AdaBoost algorithm, in the process providing a new proof that ε-AdaBoost is margin maximizing as ε converges to 0. Because both of these known algorithms are computationally expensive, we introduce a new hybrid algorithm, AdaBoost+L_1, that combines the virtues of AdaBoost with the sparsity of l_1-regularization in a computationally efficient fashion. We prove that the algorithm is margin maximizing and empirically examine its performance on five datasets.

1 INTRODUCTION

Boosting is a technique for constructing from a finite dataset a composite classifier as a linear combination of a given large set of weak classifiers. Typically, it is assumed that for every distribution, there exists one weak classifier whose error is slightly below chance, and the objective is to construct a composite classifier with a much lower probability of misclassification. AdaBoost was the first practical boosting algorithm (Freund & Schapire, 1997), and many other generalizations and extensions have since been proposed (see, for example, (Meir & Rätsch, 2003) and (Schapire, 2002) for overviews).

Theoretical work on boosting's ability to generalize have centered on two approaches. The first relies on sparsity, i.e., a limit on the number of weak classifiers with non-zero weights, or alternatively, a bound on the magnitude (say, the l_1-norm) of those weights (Freund & Schapire, 1997; Lugosi & Vayatis, 2004; Zhang & Yu, 2005). Sparsity may be desirable not only for generalization performance, but also as a means of identifying a small set of useful features. Sparsity can be enforced using early stopping or regularization, although AdaBoost often performs well even when neither of these techniques is employed. In an alternative approach, AdaBoost can be analyzed by showing that it achieves large margins on the training data, and that these large margins are sufficient to guarantee good test performance (Schapire et al., 1998). This paper focuses on the intertwined relationship between these important aspects of boosting, namely, sparsity, regularization and large margins.

Rosset et al. (2004b) have shown that for separable data, standard l_p-regularized loss minimization results in a margin maximizing classifier in the limit as regularization is relaxed. So, for datasets that admit classification using a sparse but unknown set of features, l_1-regularized loss minimization offers two potential advantages: (1) a sparse composite classifier with (2) a large margin on the training data. The major disadvantage of standard l_1-regularized loss minimization over a huge space of weak classifiers is the considerable computational burden. As an alternative, Hastie et al. (2001) studied the ε-AdaBoost algorithm in which the coefficient of a selected weak classifier is increased by a small amount ε at each iteration, instead of the potentially large step size of AdaBoost. This algorithm is slow but simpler to implement and shares some properties of l_1-regularized loss minimization. In particular, as ε converges to 0, the classifier attains the maximum possible margin on the training data (Zhang & Yu, 2005).

In Section 3 we extend Rosset et al.'s results when $p = 1$

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5. Copyright 2009 by the authors.

by deriving explicit bounds on the regularization parameter to ensure the composite classifier margin is within prescribed accuracy of the maximum achievable margin. For completeness, in an appendix we derive similar results for ε-AdaBoost and give a new proof that it is margin maximizing. (See also (Zhang & Yu, 2005).)

Known methods for solving the regularized loss minimization problem are computationally expensive. To address this, in Section 4 we introduce a new hybrid algorithm, AdaBoost+L_1, that combines in a computationally efficient fashion the virtues of AdaBoost with the sparsity produced by l_1-regularization. We prove that AdaBoost+L_1 is margin maximizing given standard assumptions of weak learnability. We empirically investigate AdaBoost+L_1 on five datasets in Section 5.

2 PRELIMINARIES

For any integer $k > 0$, let $\mathbb{R}^k$ denote k-dimensional Euclidean space and Δ_k denote the probability simplex in $\mathbb{R}^k$. Let $\mathbf{z} = (z_1, \ldots, z_k)$ denote the components of $\mathbf{z} \in \mathbb{R}^k$ and $\|\mathbf{z}\|_1 = \sum_{i=1}^k |z_i|$.

Let $S = \langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ be a sequence of m labelled training examples, where $x_i \in X$ is an instance and $y_i \in \{-1, +1\}$ its label. A binary classifier on domain X is a map $h\colon X \to \{-1, +1\}$. The performance of the classifier h with respect to a distribution $\mathbf{d} \in \Delta_m$ on S can be measured by its *edge*:

$$e(h, \mathbf{d}) = \mathrm{E}_{i \sim \mathbf{d}}\left[y_i h(x_i)\right] = \sum_{i=1}^m d_i y_i h(x_i). \qquad (1)$$

This value is contained in the interval $[-1, 1]$ and is a measure of the correlation (under $\mathbf{d}$) between the predictions $h(x_i)$ and the labels y_i. It is linearly related to the weighted error by $\Pr_{i \sim \mathbf{d}}[y_i \neq h(x_i)] = (1 - e(h, \mathbf{d}))/2$.

Let $\mathcal{H} = \{h_1, h_2, \ldots, h_N\}$ be a finite set of N weak classifiers. For example, the weak classifiers might be simple decision stumps or decision trees of bounded size. For our present purposes, we say that the training examples S are *weakly learnable* under $\mathcal{H}$ if there exists $\theta > 0$ such that for any distribution $\mathbf{d} \in \Delta_m$ one can find $h_j \in \mathcal{H}$ such that $e(h_j, \mathbf{d}) \geq \theta$. This is equivalent to there being a weak classifier that performs better than chance under distribution $\mathbf{d}$. Now for any distribution $\mathbf{d}$, $\max_j e(h_j, \mathbf{d})$ is the best edge over classifiers in $\mathcal{H}$. So

$$\theta^* = \min_{\mathbf{d} \in \Delta_m} \max_j e(h_j, \mathbf{d}) \qquad (2)$$

is the best edge over $\mathcal{H}$ under the least favorable distribution on the examples, and weak learnability is simply equivalent to θ^* being strictly positive so that, for any $\mathbf{d} \in \Delta_m$, there exists h_j with $e(h_j, \mathbf{d}) \geq \theta^* > 0$.

Now consider the composite classifier $h_{\boldsymbol{\alpha}}(x) = \sum_{j=1}^N \alpha_j h_j(x)$, where $\boldsymbol{\alpha} \in \mathbb{R}^N$. This gives the binary classification $y = \mathrm{sign}(h_{\boldsymbol{\alpha}}(x))$. Assume that $h_j \in \mathcal{H}$ implies $-h_j \in \mathcal{H}$. Then without loss of generality we can assume that $\alpha_j \geq 0$, $j = 1, \ldots, N$. We say that h_j is *active* with respect to weight coefficients $\boldsymbol{\alpha}$ if $\alpha_j > 0$.

The *margin* of $h_{\boldsymbol{\alpha}}$ on the i-th training example is

$$\mu_i(\boldsymbol{\alpha}) \quad = \quad y_i h_{\boldsymbol{\alpha}}(x_i)/\|\boldsymbol{\alpha}\|_1. \qquad (3)$$

Clearly, $\mu_i(\boldsymbol{\alpha}) \in [-1, 1]$. $\mu_i(\boldsymbol{\alpha})$ is a measure of the robustness or confidence of the composite classifier's decision on the i-th instance. The *margin of the classifier* $h_{\boldsymbol{\alpha}}$ is the least margin over all examples, $\mu(\boldsymbol{\alpha}) = \min_{i=1\ldots m} \mu_i(\boldsymbol{\alpha})$.

A *maximum-margin classifier* results by selecting $\boldsymbol{\alpha}$ to maximize $\mu(\boldsymbol{\alpha})$. Such a classifier has margin

$$\mu^* = \max_{\boldsymbol{\alpha} \in \Delta_N} \min_i \sum_{j=1}^N \alpha_j y_i h_j(x_i). \qquad (4)$$

As shown in (Freund & Schapire, 1996) and (Rätsch & Warmuth, 2005), von Neumann's min-max theorem applied to the $m \times N$ game matrix $M(i, j) = y_i h_j(x_i)$, yields:

$$\begin{aligned} \mu^* &= \max_{\boldsymbol{\alpha} \in \Delta_N} \min_{i=1,\ldots,m} \sum_{j=1}^N \alpha_j y_i h_j(x_i) \\ &= \min_{\mathbf{d} \in \Delta_m} \max_{j=1,\ldots,N} \sum_{i=1}^m d_i y_i h_j(x_i) = \theta^* \end{aligned}$$

where the last equality follows from (1) and (2). Hence $\theta^* > 0$ is the maximum achievable margin for any composite classifier.

3 l_1-REGULARIZED LOSS MINIMIZATION

We now state the general l_1 regularized loss minimization problem and give conditions under which its solution yields a maximum margin classifier as regularization is relaxed. Moreover, we give an explicit bound on the value of the regularization parameter to yield a margin within prescribed accuracy of θ^*.

Fix a loss function $L\colon \mathbb{R} \to \mathbb{R}$. Common choices include the exponential loss $L(z) = e^{-z}$, used in AdaBoost, and logistic loss $L(z) = \ln(1 + e^{-z})$. The l_1-*Regularized Loss Minimization Problem* (RLMP) with loss function L and parameter $r > 0$ is:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^m L\left(y_i \sum_{j=1}^N \alpha_j h_j(x_i)\right) \\ &\text{such that } \|\alpha\|_1 \leq r, \quad \alpha_j \geq 0, \quad j = 1, 2, \ldots, N. \end{aligned}$$

For continuous L, RLMP has at least one solution. If L is convex, then so is $\mathcal{L}$ and for any $r > 0$, RLMP will have the same solution as a convex program of the form

$$\min \sum_{i=1}^{m} L\left(y_i \sum_{j=1}^{N} \alpha_j h_j(x_i)\right) + \beta\|\alpha\|_1$$

for some $\beta > 0$.

We first prove a fundamental property: at a solution of RLMP, all active weak classifiers have identical edges.

Lemma 1 (Uniform Learning)**.** *Assume the training examples are weak learnable. Assume L is differentiable and strictly monotone decreasing, i.e., $L'(z) < 0$ for all $z \in \mathbb{R}$. Then any solution $\boldsymbol{\alpha}$ of RLMP satisfies $\sum_{j=1}^{N} \alpha_j = r$. Moreover, if we select $\mathbf{d} \in \Delta_m$ such that*

$$d_i = \frac{L'(y_i \sum_{j=1}^{N} \alpha_j h_j(x_i))}{\sum_{i=1}^{m} L'(y_i \sum_{j=1}^{N} \alpha_j h_j(x_i))}, \quad i = 1, \ldots, m, \tag{5}$$

(as in AdaBoost), then the edges of all active weak classifiers are equal and no smaller than θ^*.

Proof. Using (5) then (1)

$$\frac{\partial \mathcal{L}(\boldsymbol{\alpha})}{\partial \alpha_k} = \sum_{i=1}^{m} y_i h_k(x_i) L'\left(y_i \sum_{j=1}^{N} \alpha_j h_j(x_i)\right) \tag{6}$$

$$= e(h_k, \mathbf{d}) \sum_{i=1}^{m} L'\left(y_i \sum_{j=1}^{N} \alpha_j h_j(x_i)\right). \tag{7}$$

Under weak learnability, there exists $h_k \in \mathcal{H}$ such that $e(h_k, \mathbf{d}) \geq \theta^*$. Hence $\partial\mathcal{L}/\partial\alpha_k$ is strictly negative. If $\sum_{j=1}^{N} \alpha_j < r$, then the objective is decreased by increasing α_k; contradicting optimality.

Since $\boldsymbol{\alpha}$ is a solution to RLMP, $\partial\mathcal{L}(\boldsymbol{\alpha})/\partial\alpha_j$ must be equal for all $\alpha_j > 0$. Otherwise the objective can be further decreased by jittering the coefficients: increase those with smaller derivative and decrease those with larger derivative by the same amount. From (7) we see that the partial derivative with respect to α_k is proportional to the edge $e(h_k, \mathbf{d})$. So the edges of all active weak classifiers are equal.

Alternatively, for convex problems this follows by the KKT conditions. Let $f_i(\boldsymbol{\alpha}) = -\alpha_i, i = 1, \ldots, N$ and $f_{N+1}(\boldsymbol{\alpha}) = \sum_{j=1}^{N} \alpha_j - r$. Then the problem can be written as: $\min_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha})$ subject to $f_i(\boldsymbol{\alpha}) \leq 0$, $i = 1, 2, \ldots, N+1$. By the KKT conditions, the optimizer $\boldsymbol{\alpha}$ satisfies $\nabla\mathcal{L}(\boldsymbol{\alpha}) + \sum_{i=1}^{N+1} \lambda_i \nabla f_i(\boldsymbol{\alpha}) = 0$ with $f_i(\boldsymbol{\alpha}) = 0$ or $\lambda_i = 0$, $i = 1, 2, \ldots, N+1$. Obviously if $\alpha_i > 0$, $\lambda_i = 0$. So for all $\alpha_i > 0$, $\frac{\partial\mathcal{L}}{\partial\alpha_i} = -\lambda_{N+1}$, and by (7), the edges of all active weak classifiers are equal.

Let the edge of all active weak classifiers be θ. Assume $\theta < \theta^*$. By weak learnability, there exists $h_k \in \mathcal{H}$ and $\alpha_k = 0$ such that $e(h_k, \mathbf{d}) \geq \theta^*$. Increasing α_k and decreasing any current active weak classifier by the same small amount will reduce the objective function, contradicting optimality. □

We now use Lemma 1 to significantly extend a result of Rosset et al. (2004b) (for $p = 1$) by giving an explicit bound on the value r in RLMP to yield a margin within prescribed accuracy of θ^*.

Theorem 1. *Assume L is convex, differentiable and $L'(z) < 0$ for all z. Let $\boldsymbol{\alpha}^{(r)}$ be a solution of RLMP with parameter r and $h_{\boldsymbol{\alpha}^{(r)}}$ have margin $\mu(\boldsymbol{\alpha}^{(r)})$. Then for small $\varepsilon > 0$*

$$\frac{L'(r\theta^*)}{L'(r(\theta^* - \varepsilon))} < \frac{\varepsilon}{m(1-\theta^*)} \Rightarrow \mu(\boldsymbol{\alpha}^{(r)}) \geq \theta^* - \varepsilon.$$

Thus, if $\forall\varepsilon > 0$, $\lim_{z\to\infty} L'(z)/L'(z(1-\varepsilon)) = 0$, then $\boldsymbol{\alpha}^{(r)}$ is margin maximizing in the limit, i.e., $r_k \uparrow \infty$ implies $\lim_{k\to\infty} \mu(\boldsymbol{\alpha}^{(r_k)}) = \theta^*$.

Proof. Consider a solution $\boldsymbol{\alpha}$ of RLMP for fixed r. By Lemma 1, $r = \sum_j \alpha_j$. Let $\mu_i = \mu_i(\boldsymbol{\alpha})$ denote the margin of the i-th example. Then the edges of the active classifiers with respect to $\mathbf{d}$, given by (5), satisfy:

$$\begin{aligned}
\theta^* &\leq e(h_j, \mathbf{d}) \\
&= \sum_{i=1}^{m} y_i h_j(x_i) \frac{L'(y_i \sum_{j=1}^{N} \alpha_j h_j(x_i))}{\sum_{k=1}^{m} L'(y_k \sum_{j=1}^{N} \alpha_j h_j(x_k))} \\
&= \sum_{i=1}^{m} y_i h_j(x_i) \frac{L'(r\mu_i)}{\sum_{k=1}^{m} L'(r\mu_k)}.
\end{aligned}$$

Multiplying both sides by $(\alpha_j/\|\boldsymbol{\alpha}\|_1) \sum_{k=1}^{m} L'(r\mu_k)$, summing over j, and combining terms yields

$$\sum_{i=1}^{m} L'(r\mu_i)(\mu_i - \theta^*) \leq 0. \tag{8}$$

Next, we split this sum into three sums over $\mu_i < \theta^* - \varepsilon$, $\theta^* - \varepsilon \leq \mu_i < \theta^*$, and $\mu_i \geq \theta^*$. This yields:

$$\begin{aligned}
0 \geq & \sum_{i:\mu_i < \theta^* - \varepsilon} L'(r\mu_i)(\mu_i - \theta^*) \\
& + \sum_{i:\theta^* - \varepsilon \leq \mu_i < \theta^*} L'(r\mu_i)(\mu_i - \theta^*) \\
& + \sum_{i:\mu_i \geq \theta^*} L'(r\mu_i)(\mu_i - \theta^*) \\
\geq & -K(\varepsilon)L'(r(\theta^* - \varepsilon))\varepsilon + 0 + m(1-\theta^*)L'(r\theta^*).
\end{aligned}$$

The first sum uses the convexity of L with $K(\varepsilon)$ defined to be the number of examples for which $\mu_i < \theta^* - \varepsilon$. The second term is at least 0 because $L'(x) < 0, \forall x$. The last term is at least $m(1-\theta^*)L'(r\theta^*)$, using convexity and that μ_i's are bounded by 1. Thus,

$$K(\varepsilon) \leq \frac{m(1-\theta^*)L'(r\theta^*)}{\varepsilon L'(r(\theta^*-\varepsilon))}. \qquad (9)$$

If $L'(r\theta^*)/L'(r(\theta^*-\varepsilon)) < \varepsilon/(m(1-\theta^*))$, then $K(\varepsilon) < 1$, so all example margins are at least $(\theta^*-\varepsilon)$. Finally, given the additional assumption stated in the theorem, for any $\varepsilon > 0$, $K(\varepsilon) \to 0$ as $r_k \to \infty$. Hence $\lim_{k\to\infty} \mu(\boldsymbol{\alpha}^{(r_k)}) = \theta^*$. □

Theorem 1 implies that loss functions with exponential decay tails, such as exponential loss $L(x) = e^{-x}$ and logistic loss $L(x) = \ln(1+e^{-x})$, are margin maximizing. Polynomial decay functions $L(x) = x^{-k}, k > 0$ do not belong to this category. Theorem 1 further provides an explicit rate of convergence of the margin to θ^*. For instance, for exponential loss, it shows that if $r > (1/\varepsilon)\ln(m(1-\theta^*)/\varepsilon)$ then $\mu(\boldsymbol{\alpha}^{(r)}) \geq \theta^* - \varepsilon$.

Rosset et al. (2004b) define a non-increasing and non-negative loss function L to be a *margin maximizing loss function* if there exists $R > 0$ (possibly $R = \infty$) such that for all $\varepsilon > 0$, $\lim_{z\nearrow R} L(z(1-\varepsilon))/L(z) = \infty$. For such functions, they show the solution of RLMP maximizes the margin as $r \to \infty$. To see how this relates to our results, we consider separately the two cases that R is finite or infinite. When $R = \infty$, it must be that $\lim_{z\to\infty} L(z) = 0$. Using l'Hôpital's rule yields $\lim_{z\to\infty} L(z(1-\varepsilon))/L(z) = \lim_{z\to\infty} L'(z(1-\varepsilon))/L'(z)$. The latter condition matches the condition in Theorem 1. When $R < \infty$, such as for the hinge loss function, we do not need to let $r \to \infty$, as suggested in (Rosset et al., 2004b), to achieve the maximum margin. Actually, as the following theorem shows, for loss functions that decrease monotonically to a certain positive point and remain constant afterwards, we can always find a maximum margin classifier by solving RLMP with finite r.

Theorem 2. *Assume L is continuous, strictly monotone decreasing up to $R > 0$ and constant afterwards, i.e., $L(z) = L(R), \forall z \geq R$. Then a solution of RLMP with $r = R/\theta^*$ is margin maximizing.*

Proof. There exists $\boldsymbol{\alpha}^*$ with $\|\boldsymbol{\alpha}^*\|_1 = 1$ such that $h_{\boldsymbol{\alpha}^*}$ is margin maximizing. We first show that $\boldsymbol{\alpha} = \boldsymbol{\alpha}^* R/\theta^*$ is a solution to RLMP, and then show that any non-margin-maximizing composite classifier cannot be a solution to RLMP. First, since $\boldsymbol{\alpha}^*$ is margin maximizing $y_i \sum_{j=1}^N \boldsymbol{\alpha}_j^* h_j(x_i) \geq \theta^*$ for each i. Replacing $\boldsymbol{\alpha}^*$ with $\boldsymbol{\alpha}$ yields

$$y_i \sum_{j=1}^N \alpha_j h_j(x_i) \geq \theta^* R/\theta^* = R\,,\, i = 1, \ldots, m.$$

Then for each i, $L(y_i \sum_{j=1}^N \alpha_j h_j(x_i)) = L(R)$. Thus $\mathcal{L}(\boldsymbol{\alpha}) = mL(R)$ achieves the least possible value of $\mathcal{L}$. So $\boldsymbol{\alpha}$ is a solution and $mL(R)$ is the minimum achievable loss. Assume $\bar{\boldsymbol{\alpha}}$ is a solution of RLMP with $r = R/\theta^*$ that achieves margin $\bar{\theta} < \theta^*$. Then for at least one i, $y_i \sum_{j=1}^N \bar{\alpha}_j h_j(x_i)/\sum_{j=1}^N \bar{\alpha}_j = \bar{\theta}$. Additionally, $\sum_{j=1}^N \bar{\alpha}_j \leq r = R/\theta^*$. So for this i, we have $y_i \sum_{j=1}^N \bar{\alpha}_j h_j(x_i) \leq \bar{\theta}R/\theta^* < R$, which makes $\mathcal{L}(\bar{\boldsymbol{\alpha}}) > mL(R)$, a contradiction of optimality. □

An example is the hinge loss $L(x) = \max\{0, (1-x)\}$. Theorem 2 says RLMP with hinge loss yields a margin maximizing solution when $r = 1/\theta^*$.

Finally, if we slightly strengthen the margin maximizing loss function condition of Rosset et al. (2004b), we obtain the following stronger result.

Theorem 3. *Assume that L is monotone decreasing and that for $z \geq 0$ and $\varepsilon \geq 0$, $L((1-\varepsilon)z)/L(z) \geq f(\varepsilon z)$ where f is a strictly monotone increasing function. Then*

$$\theta^* - \mu(\boldsymbol{\alpha}^{(r)}) \leq \frac{f^{-1}(m)}{r}.$$

Proof. We note that $\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^m L(y_i h_{\boldsymbol{\alpha}}(x_i)) = \sum_{i=1}^m L(\mu_i(\boldsymbol{\alpha})\|\boldsymbol{\alpha}\|_1)$. Hence by monotonicity of L

$$L(\mu(\boldsymbol{\alpha})\|\boldsymbol{\alpha}\|_1) \leq \mathcal{L}(\boldsymbol{\alpha}) \leq mL(\mu(\boldsymbol{\alpha})\|\boldsymbol{\alpha}\|_1).$$

Let $\boldsymbol{\alpha}^*$ be margin maximizing with $\|\boldsymbol{\alpha}^*\|_1 = r$. Then by optimality of $\boldsymbol{\alpha}^{(r)}$,

$$L(\mu(\boldsymbol{\alpha}^{(r)})r) \leq \mathcal{L}(\boldsymbol{\alpha}^{(r)}) \leq \mathcal{L}(\boldsymbol{\alpha}^*) \leq mL(\theta^* r).$$

Hence $L(\mu(\boldsymbol{\alpha}^{(r)})r)/L(\theta^* r) \leq m$. Let $z = \theta^* r$. We can write

$$L(\mu(\boldsymbol{\alpha}^{(r)})r) = L(z\mu(\boldsymbol{\alpha}^{(r)})/\theta^*) = L((1-\varepsilon_r)z)$$

with $\varepsilon_r = 1 - \mu(\boldsymbol{\alpha}^{(r)})/\theta^*$. Since $\mu(\boldsymbol{\alpha}^{(r)}) \leq \theta^*$, $\varepsilon_r \geq 0$. So by the theorem's assumption

$$f(\varepsilon_r z) \leq L((1-\varepsilon_r)z)/L(z) \leq m.$$

Inverting f yields $\varepsilon_r z \leq f^{-1}(m)$ and substituting for ε_r and z gives the result. □

The exponential loss function satisfies the assumptions of Theorem 3 with $f(\varepsilon z) = e^{\varepsilon z}$. Hence for exponential loss, $\theta^* - \mu(\boldsymbol{\alpha}^{(r)}) \leq \ln(m)/r$.

4 NEW ALGORITHM: ADABOOST+L$_1$

Although conceptually simple, the direct solution of RLMP and related approximate methods such as ε-boosting are too slow to be practical for large weak classifier spaces. To address this, we propose a new hybrid algorithm, AdaBoost+L_1 (see Figure 1), that efficiently penalizes the

AdaBoost+L$_1$

```
1. Initialize: select $\nu \in (0,1]$, set $r_0 = 0$, $\boldsymbol{\alpha}_0 = \mathbf{0} \in \mathbb{R}^N$, $U_0 = \emptyset$, and $d_0(i) = \frac{1}{m}, i = 1, \ldots, m$.
For $t = 1, 2, \ldots$
    2. Find $h_k \in \mathcal{H}$ such that $e(h_k, \mathbf{d}_{t-1}) \geq \theta^*$.
    3. Update $U$ and $r$:
       $U_t = U_{t-1} \cup \{k\}, \quad r_t = r_{t-1} + \frac{\nu}{2} \ln \frac{1+e(h_k,\mathbf{d}_{t-1})}{1-e(h_k,\mathbf{d}_{t-1})}$.
    4. Solve the (small) convex minimization problem over $\{\alpha_j\}_{j \in U_t}$:
       $\min \sum_{i=1}^{m} \exp\left(-y_i \sum_{j \in U_t} \alpha_j h_j(x_i)\right)$
       $s.t. \sum_{j \in U_t} \alpha_j \leq r_t, \quad \alpha_j \geq 0, \ \forall j \in U_t$
    5. Update the coefficients:
       $\alpha_t(j) = \begin{cases} \alpha_j & \text{if } j \in U_t; \\ 0 & \text{otherwise} \end{cases}$.
    6. Update the distribution:
       $d_t(i) = \frac{\exp(-y_i \sum_{j=1}^{N} \alpha_t(j) h_j(x_i))}{\sum_{i=1}^{m} \exp(-y_i \sum_{j=1}^{N} \alpha_t(j) h_j(x_i))}, \quad i = 1, \ldots, m.$
```

Figure 1: The AdaBoost+L_1 algorithm.

l_1 norm of the coefficients. The hybrid nature of the algorithm is reminiscent of Grove and Schuurmans' LP-AdaBoost algorithm (1998). Steps 2 and 3 of each iteration perform the first part of a standard AdaBoost update with possibly non-optimal weak classifier selection (meaning the selected weak classifier need not have maximum edge, just not worse than θ^*). Before updating the coefficients, step 4 solves a small l_1-regularized loss minimization problem *using only the coefficients in the set U_t*. This can be thought of as a balancing step that seeks to bring the active classifiers into the uniform learning condition. Finally, steps 5 and 6 update the coefficients and the distribution. A key point is that the optimization problem in step 4 is convex and only involves a small number ($\ll N$) of the weak classifiers. The set U_t consists of the indices that have been used at least once up to and including round t. The parameter $\nu \in (0, 1]$ controls the speed of relaxation of the regularization, and is essentially the same as Friedman's (2001) use of "shrinkage" with boosting. Smaller values of ν impose more aggressive regularization and hence seek sparser solutions.

The next theorem shows that AdaBoost+L_1 yields a maximum margin solution as $t \to \infty$.

Theorem 4. *Let $\boldsymbol{\alpha}_t$ be the solution of AdaBoost+L_1 after round t and set $\tilde{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}_t / \|\boldsymbol{\alpha}_t\|_1$. Then $\lim_{t\to\infty} \mu(\boldsymbol{\alpha}_t) = \theta^*$, and every limit point of $\tilde{\boldsymbol{\alpha}}_t$ is margin maximizing.*

To prove the theorem, we first introduce a lemma. Let $A_t = \{j : \alpha_t(j) > 0\}$.

Lemma 2. *The edges $e(h_j, \mathbf{d}_t)$, $j \in A_t$, have a common value θ_t. Moreover, $\liminf_{t\to\infty} \theta_t \geq \theta^*$.*

Proof. By the proof of Lemma 1, the edges $e(h_j, \mathbf{d}_t)$, $j \in A_t$, have a common value θ_t. If $\theta_t < \theta^*$, then in the next round $k \notin U_t$ with $e(h_k, \mathbf{d}_t) \geq \theta^*$ is selected and added to U_{t+1}. But $\{U_t\}$ is a monotone increasing set sequence bounded above by a finite set. Hence it can only be enlarged a finite number of times. So there exits t_0 such that $\theta_t \geq \theta^*$, $t \geq t_0$. Hence $\liminf_{t\to\infty} \theta_t \geq \theta^*$. □

Proof. (Of Theorem 4) In each round, r_t increases by at least $\frac{1}{2}\nu \ln((1 + \theta^*)/(1 - \theta^*))$. So $\lim_{t\to\infty} r_t = \infty$. By Lemma 2 there exists t_1 such that $\theta_t > 0$ for $t \geq t_1$, where θ_t is as above. Henceforth we restrict attention to $t \geq t_1$. Then the edges with $j \in A_t$ are positive, and by (7), $\|\boldsymbol{\alpha}_t\|_1 = r_t$.

Let $K_t(\varepsilon)$ denote the number of examples with margin below $(\theta_t - \varepsilon)$ after round t. Following the steps in the proof of Theorem 1 we obtain

$$0 \leq K_t(\varepsilon) \leq \frac{m(1 - \theta_t)L'(r_t\theta_t)}{\varepsilon L'(r_t(\theta_t - \varepsilon))}.$$

Now $1 - \theta_t$ is bounded, $K_t(\varepsilon)$ is an integer and for $L(z) = e^{-z}$, $\lim_{t\to\infty} L'(r_t\theta_t)/L'(r_t(\theta_t - \varepsilon)) = 0$. Hence there exists $t(\varepsilon) \geq t_1$ with $K_t(\varepsilon) = 0$ for $t \geq t(\varepsilon)$. Thus for $t \geq t(\varepsilon)$, $\theta_t - \varepsilon \leq \mu(\boldsymbol{\alpha}_t) \leq \theta^*$. Taking the $\liminf$ of each term and using Lemma 2 gives: $\theta^* - \varepsilon \leq \liminf_{t\to\infty}(\theta_t - \varepsilon) \leq \liminf_{t\to\infty} \mu(\boldsymbol{\alpha}_t) \leq \limsup_{t\to\infty} \mu(\boldsymbol{\alpha}_t) \leq \theta^*$. Since $\varepsilon > 0$ is arbitrary, $\lim_{t\to\infty} \mu(\boldsymbol{\alpha}_t) = \theta^*$. Let subsequence $\{\tilde{\boldsymbol{\alpha}}_{t_k}\}$ converge to a limit point $\boldsymbol{\beta}$ of $\{\tilde{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}_t / \|\boldsymbol{\alpha}_t\|_1\}$. Since μ is continuous in $\boldsymbol{\alpha}$, $\theta^* = \lim_{k\to\infty} \mu(\boldsymbol{\alpha}_{t_k}) = \lim_{k\to\infty} \mu(\tilde{\boldsymbol{\alpha}}_{t_k}) = \mu(\lim_{k\to\infty} \tilde{\boldsymbol{\alpha}}_{t_k}) = \mu(\boldsymbol{\beta})$. □

We note that the Boost+L_1 framework can be generalized to other loss functions such as logistic loss, to achieve a maximum margin solution in the limit as regularization is relaxed.

5 EXPERIMENTS

We compared AdaBoost+L_1 with AdaBoost on five datasets: a synthetic dataset, *Ringnorm*, and four datasets from the UCI online machine learning repository. *Ringnorm* is a binary classification task between two 20-dimensional Gaussian distributions: $N(\mathbf{0}, 4I)$ and $N(\mu, I)$ ($\mu = (a, a, \ldots, a)$, $a = 1/\sqrt{20}$). The three real datasets are *Diabetes* (detecting diabetes based on medical information), *German* (determining credit risk based on financial histories), *Spam* (identifying spam email messages based on word frequencies) and *Ionosphere* (classifying radar returns from the ionosphere). In *Ringnorm* we randomly generated 100 examples for training and 5000 examples for testing; In *Diabetes*, we used 100 examples for training and the other 668 for testing; In *Spam*, we used 100 for training and the other 4501 for testing; In *German*, we used 200 for training and the other 800 for testing; In *Ionosphere*, we used 100 for training and the other 250 for testing. For each dataset, we ran both algorithms until the error rates stopped decreasing and overfitting occurs. We ran all the experiments 20 times (using different samples of training/testing examples) and averaged the results.

We used simple decision stumps as weak classifiers. This illustrates the relative performance of the new algorithm. Better performance may be achieved by more complex weak classifiers such as decision trees.

Since each algorithm learns at a different rate, direct round-by-round comparison of test performance or sparsity is not a useful metric. Each algorithm will also begin overfitting at different rounds, so comparing results at the final round is also not appropriate. In applications one could use cross validation to determine the stopping point for training that yields the best $\boldsymbol{\alpha}$ for each dataset. Hence it makes more sense to compare the best test error of each algorithm over all rounds and the associated sparsity of the classifier at this "sweet spot".

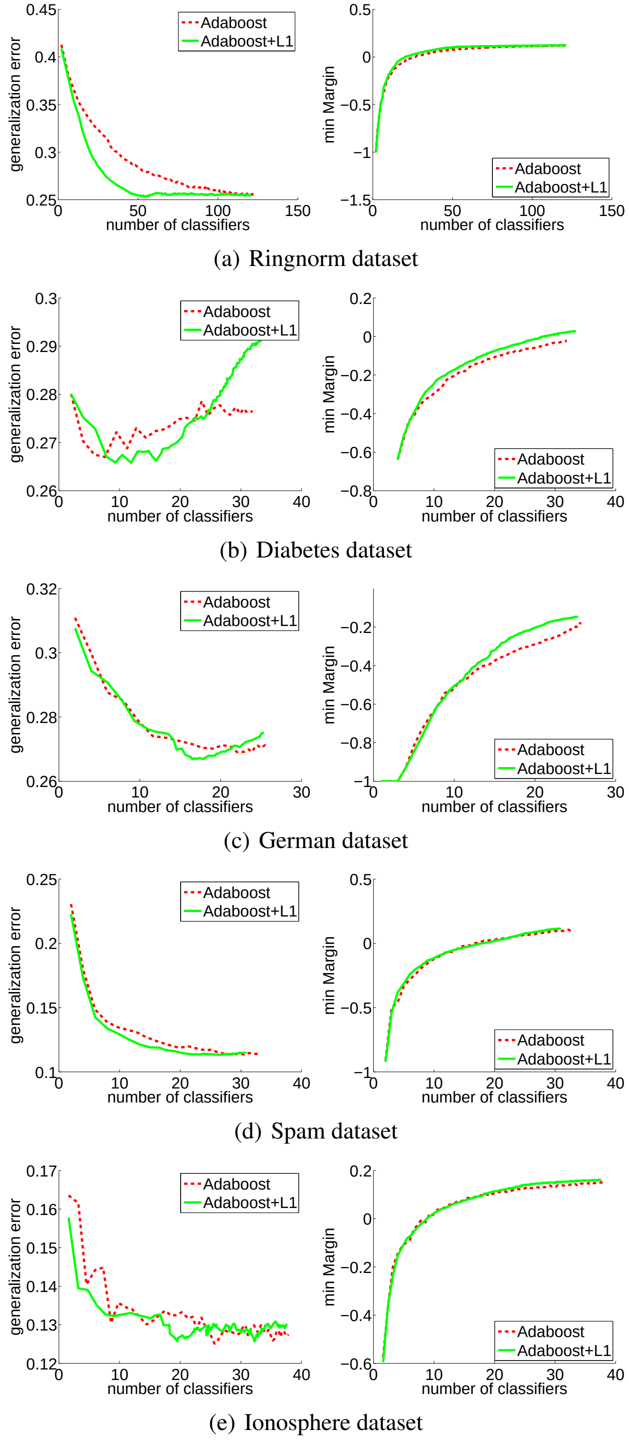
To this end, in Figure 2 we plot the test error and classifier margin $\mu(\boldsymbol{\alpha})$ (on training examples) as functions of the number of classifiers chosen. The best error rates achieved at the "sweet spots" and the associated number of base classifiers (average over 20 trials) are shown in Tables 1 and Table 2.

	AdaBoost	AdaBoost+L_1	Rel. Improvement
Ringnorm	25.5%	25.3%	0.8%
Diabetes	26.7%	26.6%	0.4%
German	26.9%	26.7%	0.8%
Spam	11.3%	11.3%	0.2%
Ionosphere	12.5%	12.6%	-0.6%

Table 1: Comparison of error rates

	AdaBoost	AdaBoost+L_1	Rel. Improvement
Ringnorm	121.8	55.0	54.8%
Diabetes	7.6	11.9	-56.6%
German	22.1	16.5	25.2%
Spam	31.3	26.5	15.3%
Ionosphere	26.7	19.6	26.8%

Table 2: Comparison of numbers of classifiers



(a) Ringnorm dataset

(b) Diabetes dataset

(c) German dataset

(d) Spam dataset

(e) Ionosphere dataset

Figure 2: Experiment results on five datasets. The left column shows the error rate on testing data, the right column shows the minimal margin on training data.

As shown in the left column of Figure 2 and the tables, AdaBoost+L_1 achieves similar best error rates to AdaBoost. But in most cases, AdaBoost+L_1 achieves its best error rate with much fewer base classifiers (Table 2). The only exception is the *Diabetes* dataset. However, it should be noted that the actual numbers in this case are very small, so a small difference results in a large percentage change. The right column of Figure 2 shows that with a fixed number of classifiers, AdaBoost+L_1 achieves equal or slightly better margins on the training examples.

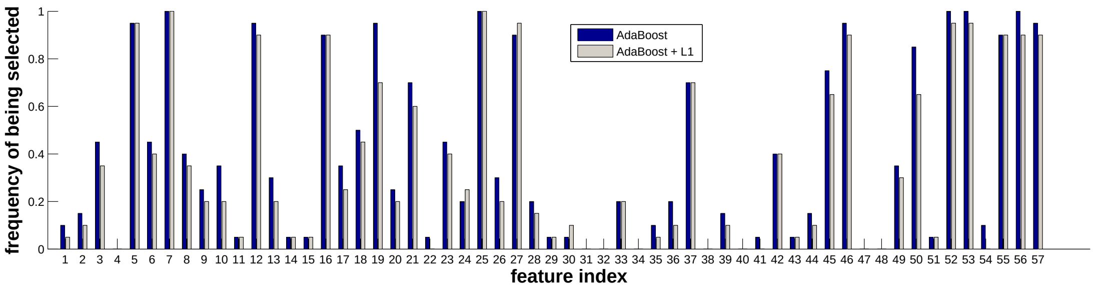


Figure 3: The frequencies of different features being selected, Spam Dataset

From Table 1 we can see that the improvement in error rate is modest, or even too small to justify the extra complexity of the algorithm. Therefore AdaBoost+L_1 is unnecessary if the only objective is low classification error. On the other hand, as shown in Table 2, except for the *Diabetes* dataset, AdaBoost + L_1 reduces the number of active classifiers by a considerable percentage. Therefore AdaBoost + L_1 is preferable in applications where a sparse set of features is advantageous for other purposes.

In the *Diabetes* and *German* datasets, the error rate of AdaBoost+L_1 becomes higher than that of AdaBoost at the end of the curve. As mentioned above, this does not mean that AdaBoost+L_1 has worse performance. In the *German* dataset, for example, AdaBoost+L_1 hits the overfitting watershed earlier than AdaBoost. With 20 classifiers, AdaBoost is still "fitting" the data, while AdaBoost+L_1 is already overfitting after passing a sweet spot with better performance.

AdaBoost+L_1 takes more rounds to select the same number of classifiers since it repeatedly adjusts the weights of selected classifiers and removes some previously selected classifiers (by zeroing their coefficients). For example, in the *Ringnorm* experiment, to generate around 120 active classifiers (as shown on the graph), we ran AdaBoost+L_1 for 1000 iterations, while AdaBoost reached this number of active classifiers after 700 iterations. This accords with the main goal of the algorithm: to select fewer active classifiers over a given number of rounds while at the same time achieving at least as good generalization performance. As the results indicate, this is achieved in most cases.

The results suggest that AdaBoost+L_1 chose more informative classifiers/features. To better understand this, we investigated the *Spam* experiments in greater detail. For each of the 20 repetitions, we recorded whether a single-feature classifier had been selected by AdaBoost and AdaBoost+L_1 at the 60th iteration. For each feature, we then plotted a histogram showing the selection frequency under each algorithm (Figure 3). Both algorithms select important features such as "free", "$" and "!". However, AdaBoost tends to select presumably irrelevant words such as "you" (number 19 on the plot), "mail" (10) and left parenthesis "(" (50) about 20% more often than AdaBoost+L_1. Other words such as "all" (3), "people" (13) and "re" (45) are chosen by AdaBoost about 10% more often. By avoiding these non-discriminative features, AdaBoost+L_1 is able to achieve roughly the same classification accuracy using a smaller set of weak classifiers.

6 CONCLUSION

We have shown that l_1-regularized boosting (RLMP), under various conditions (Theorems 1, 2, 3), attains the maximum margin as regularization is relaxed. Moreover, we obtained a quantitative relationship between the regularization parameter r and the achievable margin. We provided similar convergence rate results for ε-AdaBoost and gave a simple proof that it is also margin-maximizing as ε vanishes. We observe that approximate uniform learning is also enforced in other iterative margin maximizing algorithms, such as TotalBoost (Warmuth et al., 2006) and LPBoost (Grove & Schuurmans, 1998), under the name "totally corrective learning." The AdaBoost+L_1 algorithm provably achieves the maximum margin as $t \to \infty$ and has been shown empirically to efficiently yield sparser solutions than AdaBoost with roughly equal "sweet spot" generalization performance. This suggests it is doing better feature selection. A more precise quantitative analysis of the trade-off between sparsity, margin and generalization merits further investigation.

APPENDIX: ε-ADABOOST

In this appendix, we give an analysis of ε-boosting (Hastie et al., 2001). This method works much like AdaBoost and related algorithms: a composite classifier $h_{\boldsymbol{\alpha}}$ is built up over a sequence of rounds. In each round, a distribution $\mathbf{d}$ as in (5) is defined, and a weak classifier h_j with large edge $e(h_j, \mathbf{d})$ selected. However, unlike AdaBoost, ε-boosting increases α_j by only some small $\varepsilon > 0$. Rosset et al. (2004a) show that ε-boosting approximates the solution to the l_1-regularized boosting problem, and for ε small, maximizes the margin in the limit of convergence (Zhang & Yu, 2005).

ε-AdaBoost uses the exponential loss $L(z) = \exp(-z)$. Below we provide an explicit bound for ε-AdaBoost on the difference of the margin from θ^* as a function of ε and the number of rounds. Let $\boldsymbol{\alpha}_t$ denote the coefficient vector after t rounds with $\boldsymbol{\alpha}_0 = \mathbf{0}$.

Theorem 5. *Fix $0 < \varepsilon < \theta^*$. After t rounds, the margin $\mu(\boldsymbol{\alpha}_t)$ of ε-AdaBoost satisfies*

$$\begin{aligned} \theta^* - \mu(\boldsymbol{\alpha}_t) &\le \theta^* - \frac{\ln(1 + \frac{\varepsilon(\theta^*-\varepsilon)}{1-\theta^*\varepsilon})}{\varepsilon} + \frac{\ln m}{t\varepsilon} \\ &\simeq \varepsilon + \frac{\ln m}{t\varepsilon} \quad \textit{for } \varepsilon \textit{ small}. \end{aligned}$$

To prove Theorem 5, we first introduce a lemma.

Lemma 3. *For $0 < \varepsilon < 0.5$,*

$$\mathcal{L}(\boldsymbol{\alpha}_t) \le \left(\frac{1-\theta^*\varepsilon}{1-\varepsilon^2}\right)^t m. \tag{10}$$

Proof. At round t, suppose the j-th weak classifier with edge $e(h_j, \mathbf{d}_t) \ge \theta^*$ is selected. Then according to the algorithm, $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1} + \varepsilon \mathbf{e}_j$, where $\mathbf{e}_j$ is the j-th natural basis vector. Using Taylor Series Expansion (about $\varepsilon = 0$), we obtain $\mathcal{L}(\boldsymbol{\alpha}_{t-1} + \varepsilon\mathbf{e}_j) = \sum_{k=0}^{\infty} \frac{C(k)}{k!}\varepsilon^k$, where

$$\begin{aligned} C(k) &= \left.\frac{d^k \mathcal{L}(\boldsymbol{\alpha}_{t-1} + \varepsilon\mathbf{e}_j)}{d\varepsilon^k}\right|_{\varepsilon=0} \\ &= \sum_{i=1}^{m} (-y_i h_j(x_i))^k \exp(-y_i h_{\boldsymbol{\alpha}_{t-1}}(x_i)) \\ &= \begin{cases} C(1) & k \text{ odd} \\ \mathcal{L}(\boldsymbol{\alpha}_{t-1}) & k \text{ even.} \end{cases} \end{aligned}$$

Similar to (7), we have $C(1) = -e(h_j, \mathbf{d}_t)\mathcal{L}(\boldsymbol{\alpha}_{t-1}) \le -\theta^*\mathcal{L}(\boldsymbol{\alpha}_{t-1})$. Therefore,

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}_t) &= \mathcal{L}(\boldsymbol{\alpha}_{t-1} + \varepsilon\mathbf{e}_j) \\ &\le \mathcal{L}(\boldsymbol{\alpha}_{t-1})\left(1 - \theta^*\varepsilon + \frac{\varepsilon^2}{2!} - \frac{\theta^*\varepsilon^3}{3!} + \cdots\right) \\ &\le \mathcal{L}(\boldsymbol{\alpha}_{t-1})(1 - \theta^*\varepsilon + \varepsilon^2 - \theta^*\varepsilon^3 + \cdots) \\ &= \mathcal{L}(\boldsymbol{\alpha}_{t-1})\left(\frac{1-\theta^*\varepsilon}{1-\varepsilon^2}\right). \end{aligned}$$

In the third line, we used the fact that $\varepsilon \le 0.5$. Repeating the argument for each round t and noting that $\mathcal{L}(\boldsymbol{\alpha}_0) = m$ gives (10). □

Proof. (Of Theorem 5) At round t, let the margin on the i-th example be $\mu_i = \mu_i(\boldsymbol{\alpha}_t) = y_i h_{\boldsymbol{\alpha}_t}(x_i)/\|\boldsymbol{\alpha}_t\|_1$. Now $\|\boldsymbol{\alpha}_t\|_1 = t\varepsilon$, because every coefficient is monotone non-decreasing. Hence $\mathcal{L}(\boldsymbol{\alpha}_t) = \sum_{i=1}^{m} \exp(-y_i h_{\boldsymbol{\alpha}_t}(x_i)) = \sum_{i=1}^{m} \exp(-\|\boldsymbol{\alpha}_t\|_1 \mu_i) = \sum_{i=1}^{m} \exp(-t\varepsilon\mu_i) \ge \exp(-t\varepsilon \min_i \mu_i)$. Using $\mu(\boldsymbol{\alpha}_t) = \min_i \mu_i$, and combining this inequality with Lemma 3 yields $\exp(-t\varepsilon\mu(\boldsymbol{\alpha}_t)) \le \left(\frac{1-\theta^*\varepsilon}{1-\varepsilon^2}\right)^t m$. Taking logs and rearranging gives (10). □

Acknowledgements

This research was supported in part by NSF Grants IIS-0325500 and CCR-0325463.

References

Freund, Y., & Schapire, R. E. (1996). Game theory, on-line prediction and boosting. In *COLT '96: Proc. ninth annual conf. on Comp. learning theory*, pp. 325–332 New York, NY, USA. ACM.

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

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5).

Grove, A. J., & Schuurmans, D. (1998). Boosting in the limit: Maximizing the margin of learned ensembles. In *Proc. Fifteenth National Conf. on Artificial Intelligence*.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Lugosi, G., & Vayatis, N. (2004). On the Bayes-risk consistency of regularized boosting methods. *Annals of Statistics*, *32*(1), 30–55.

Meir, R., & Rätsch, G. (2003). An introduction to boosting and leveraging. In Mendelson, S., & Smola, A. (Eds.), *Advanced Lectures on Machine Learning (LNAI2600)*, pp. 119–184. Springer.

Rätsch, G., & Warmuth, M. (2005). Efficient margin maximization with boosting. *J. Machine Learning Research*, *6*, 2131–2152.

Rosset, S., Zhu, J., & Hastie, T. (2004a). Boosting as a regularized path to a maximum margin classifier. *J. Machine Learning Res.*, *5*, 941–973.

Rosset, S., Zhu, J., & Hastie, T. (2004b). Margin maximizing loss functions. In *Advances in Neural Information Processing Systems 16*.

Schapire, R. E. (2002). The boosting approach to machine learning: An overview. In *Nonlinear Estimation and Classification*. Springer.

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

Warmuth, M. K., Liao, J., & Rätsch, G. (2006). Totally corrective boosting algorithms that maximize the margin. In *Proc. Int. Conf. on Machine Learning*.

Zhang, T., & Yu, B. (2005). Boosting with early stopping: Convergence and consistency. *Annals of Statistics*, *33*(4), 1538–1579.