Precise Statements of Convergence for AdaBoost and arc-gv

Cynthia Rudin, Robert E. Schapire, and Ingrid Daubechies

We wish to dedicate this paper to Leo Breiman.

Abstract. We present two main results, the first concerning Freund and Schapire's AdaBoost algorithm, and the second concerning Breiman's arc-gv algorithm. Our discussion of AdaBoost revolves around a circumstance called the case of "bounded edges", in which AdaBoost's convergence properties can be completely understood. Specifically, our first main result is that if AdaBoost's "edge" values fall into a small interval, a corresponding interval can be found for the asymptotic margin. A special case of this result closes an open theoretical question of Rätsch and Warmuth. Our main result concerning arc-gv is a convergence rate to a maximum margin solution. Both of these results are derived from an important tool called the "smooth margin", which is a differentiable approximation of the true margin for boosting algorithms.

1. Introduction

In Caruana and Niculescu-Mizil's 2006 empirical survey of classification algorithms [3], boosted decision trees were rated as the best of the state-of-the-art algorithms. Boosting algorithms are clearly among the most competitive and most successful algorithms for statistical learning, yet there is still much to be understood about the convergence properties of these algorithms. The field of boosting was started in 1989 by Schapire [16] who showed that it was possible to construct a "strong" classifier out of a collection of "weak" classifiers that perform only slightly better than a random guess. Freund and Schapire's AdaBoost algorithm [5] was the first practical boosting algorithm. Due to AdaBoost's success, many similar boosting algorithms have since been introduced. Some of these algorithms, such

2000 *Mathematics Subject Classification.* Primary 68W40,68Q25; Secondary 68Q32.

Key words and phrases. boosting, large margin classification, coordinate ascent/descent, AdaBoost, arc-gv, convergence rates.

This material is based upon work supported by the National Science Foundation under Grant Numbers 0434636, CCR-0325463, IIS-0325500 and DMS-9810783; and by AFOSR award F49620-01-1-0099. This work was done while CR's affiliations were Program in Applied and Computational Mathematics and Department of Computer Science, Princeton University, and Center for Neural Science, Courant Institute, and Howard Hughes Medical Institute, New York University.

as Breiman's arc-gv algorithm [2], were designed as empirical tools to study AdaBoost's convergence properties; AdaBoost is difficult to analyze theoretically in the separable case so such empirical tools are quite useful. Breiman's arc-gv is quite similar to AdaBoost (in fact the pseudocodes differ by only one line), though AdaBoost has been found to exhibit interesting dynamical behavior that may sometimes resemble chaos, or may sometimes converge to provably stable cycles [13] (one can now imagine why AdaBoost is difficult to analyze) whereas arc-gv converges very nicely. See [17] or [7] for an introduction to boosting.

When AdaBoost was introduced by Freund and Schapire, a prescribed number of iterations was associated with the algorithm. The theory suggested that the user must stop iterating after the prescribed number is reached in order to prevent overfitting. However, although overfitting is sometimes observed, in many cases, the opposite effect was observed by experimentalists, namely that AdaBoost often does not suffer from overfitting, even after a large number of iterations past the prescribed number [1, 8, 4]. This lack of overfitting has been explained to some extent by the margin theory [18]. This theory not only explains the success of AdaBoost, but also helps us to understand cases in which the algorithm overfits or otherwise gives poor results.

The *margin* of a boosted classifier (also called the *minimum margin over training examples*) is a number between -1 and 1 that can be used to provide a guarantee on the generalization performance via the margin theory. A large margin indicates that the classifier tends to perform well on test data. Moreover, the margin of the boosted classifier is directly controlled by the *edges* of the weak classifiers, a relationship whose understanding we significantly strengthen in this paper.

The margin theory for boosting was developed partly in response to experiments, and partly to the corresponding margin theory for Support Vector Machines (SVM's) [19]. Yet, there is a difference in the history for AdaBoost and SVMs: the SVM algorithm developed alongside the SVM margin bounds, so SVM was designed to maximize its margin, whereas AdaBoost was not; AdaBoost was introduced *before* the margin bounds for boosting. AdaBoost's convergence with respect to the margin is actually quite difficult to understand; in fact prior to this work, the only separable case where convergence could be understood was the very special cyclic case [13].

In this paper, we discuss fundamental convergence properties of both AdaBoost and arc-gv with respect to the margin. For AdaBoost, we present a new important case in which convergence can be completely understood (called the case of "bounded edges"). For arc-gv, we prove convergence, with a fast convergence rate, to a maximum margin solution. A special case of the first result closes the "gap in theory" referred to by Rätsch and Warmuth [10, 11], and answers the question "how far below maximal could AdaBoost's margin be?" The second result answers what had been posed as an open problem Meir and Rätsch [7].

1.1. Background and Discussion of Results. The training set consists of $m > 1$ examples with labels $\{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,m}$, where $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\}$, and our decision function is $f : \mathcal{X} \to \mathbb{R}$, where $\text{sign}(f)$ indicates the predicted class. The margin is defined by $\mu(f) := \min_i y_i f(\mathbf{x}_i)$, and for boosting, f can be written in the form $f = \sum_j \lambda_j h_j(\mathbf{x}_i)/\|\boldsymbol{\lambda}\|_1$, where the h_j's are the "weak classifiers", or "features", $h_j : \mathcal{X} \to \{-1, 1\}$ and $\boldsymbol{\lambda} \in \mathbb{R}^n_+$. According to the margin theory, if all other factors are equal, maximizing the margin indicates a lower generalization

error, i.e., the vector $\boldsymbol{\lambda}$ should be chosen to maximize $\mu(\boldsymbol{\lambda})$. The corresponding maximum margin is denoted by

$$\rho := \max_{\boldsymbol{\lambda} \in \mathbb{R}^n_+} \mu(\boldsymbol{\lambda}).$$

This expression can be rewritten by defining an $m \times n$ matrix $\mathbf{M}$ elementwise by $M_{ij} = y_i h_j(\mathbf{x}_i)$, and thus

$$\rho = \max_{\boldsymbol{\lambda} \in \mathbb{R}^n_+} \frac{\min_i (\mathbf{M}\boldsymbol{\lambda})_i}{||\boldsymbol{\lambda}||_1},$$

where $(\cdot)_i$ means the i^{th} vector component. AdaBoost is an iterative update procedure for $\boldsymbol{\lambda}$. At each iteration, AdaBoost's weak learning algorithm selects a weak classifier with a large *edge* value, where the edge can be thought of as the weak classifier's advantage over a random guess, or equivalently a directional derivative of AdaBoost's objective function along one of the coordinates of $\boldsymbol{\lambda}$. As we will discuss, AdaBoost's edge values are required to be greater than ρ, i.e., the weak learning algorithm does not need to be optimal, but must at least be sufficiently good.

As we mentioned, SVM's are designed to maximize the SVM margin, whereas AdaBoost was not designed to maximize the margin for boosting, or even to achieve large margins; it was only after AdaBoost was designed that its margin properties were theoretically and empirically explored. Empirically, many authors have argued that AdaBoost often asymptotically maximizes the margin [6, 10] (even though it is now theoretically known that AdaBoost can fail badly to maximize the margin [13]). There was a notable empirical work to challenge this claim before the theoretical result, namely the work of Breiman [2]. In his paper, Breiman proposed the algorithm arc-gv. His thesis was that arc-gv achieves larger margins than AdaBoost, yet achieves worse generalization error. In fact, Breiman's conjecture is far reaching, since his work on arc-gv suggests that maximizing the margin does not help generalization error, which disagrees with the predictions of the margin theory. Recent work has provided some explanation for his results, firstly, the theoretical result that, indeed, AdaBoost does not generally maximize the margin [13], and secondly, that experimental artifacts concerning the complexity of the weak learning algorithm may be responsible for some of his observations [12]. When this complexity is controlled, arc-gv continues to achieve larger minimum margins $\mu(\boldsymbol{\lambda})$, but there is a significant difference between the distribution of margin values $y_i f(\mathbf{x}_i)$ between the two algorithms; AdaBoost achieves much higher margins overall (and generally better test performance). Years earlier, Grove and Schuurmans [6] observed the same phenomenon; highly controlled experiments showed that AdaBoost achieved smaller minimum margins $\mu(\boldsymbol{\lambda})$, overall larger margins $y_i f(\mathbf{x}_i)$, and often better test performance than LP-AdaBoost. In Section 3 we present a set of very controlled experiments supporting the margin theory; these experiments, which compare AdaBoost with only itself, show that the margin is actually quite a good indicator of generalization in a specific sense.

Let us discuss the theoretical results concerning AdaBoost's convergence properties in more depth. AdaBoost has been shown to achieve large margins, but not maximal margins. To be precise, Schapire et al. [18] showed that AdaBoost achieves at least half of the maximal margin, i.e., if the maximum margin is ρ, AdaBoost will achieve a margin of at least $\rho/2$. This bound has been tightened by Rätsch and

Warmuth, who have shown that AdaBoost asymptotically achieves a margin of at least $\Upsilon(\rho) > \rho/2$, where $\Upsilon : (0,1) \to (0,\infty)$ is a monotonically increasing function shown in Figure 1 (the monotonicity can be shown by considering its derivative),

$$\Upsilon(r) := \frac{-\ln\left(1-r^2\right)}{\ln\left(\frac{1+r}{1-r}\right)}. \tag{1.1}$$

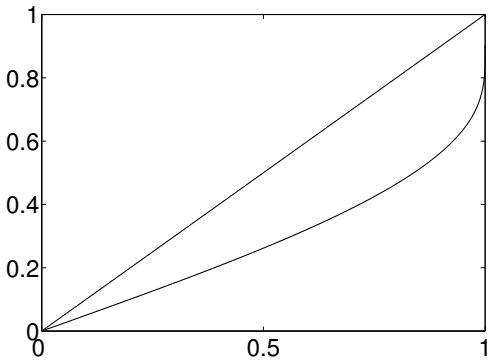

FIGURE 1. Plot of $\Upsilon(r)$ versus r (lower curve), along with the function $f(r) = r$ (upper curve).

Our contribution is from the other direction; we have just described theoretical lower bounds for the margin, whereas we are now interested in upper bounds. Previously, we have shown that it is possible for AdaBoost to achieve a margin that is significantly below the maximal value [13], and in this work, we show that Rätsch and Warmuth's bound is actually tight. In other words, we will prove that it is possible for AdaBoost to achieve an asymptotic margin arbitrarily close to $\Upsilon(\rho)$. So how far below maximal could AdaBoost's margins possibly be? The answer is $\Upsilon(\rho)$, where ρ is the value of the maximum margin. More generally, our theorem regarding the case of "bounded edges" says the following:

- If AdaBoost's edge values are within a range $[\bar{\rho}, \bar{\rho}+\sigma]$ for some $\bar{\rho} \geq \rho$, then AdaBoost's margin asymptotically lies within the interval $[\Upsilon(\bar{\rho}), \Upsilon(\bar{\rho}+\sigma)]$.

Hence, this result is a direct relationship between AdaBoost's edges (which measure the performance of the weak learning algorithm) and the asymptotic margin. As far as we know, this case of bounded edges and the case of stable cycles studied in [13] are the only two currently known circumstances where AdaBoost's convergence, with respect to the margin, can be analytically understood in the separable case (where the training error vanishes). In Section 3, we will state the theorem formally and provide theoretically-driven experiments indicating that an increase in the margins does correlate directly with a decrease in AdaBoost's probability of error on test data. Thus, it is possible that margins are an important indication of generalization error.

We present a convergence rate for Breiman's arc-gv as our second main result. For two similar algorithms, it is quite a contrast; AdaBoost has idiosyncratic tendencies such as cyclic patterns, whereas arc-gv converges to a maximum margin solution with a fast convergence rate.

Both discussions (the one for AdaBoost in Section 3 and the one for arc-gv in Section 4) depend heavily on a quantity called the *smooth margin*. The smooth margin obeys a useful recursion relation which helps greatly in the analyses for both algorithms. For more detailed analysis of the smooth margin function see [14], and for detailed proofs, see the extended version of this work [15].

2. Notation

We have already introduced the collection of weak classifiers $\{h_j\}_{j=1,\ldots,n}$, the coefficient vector $\boldsymbol{\lambda}$, the matrix $\mathbf{M}$, the margin $\mu(\boldsymbol{\lambda})$, and the maximum margin ρ, and recall that the number of examples is m. A boosting algorithm maintains a distribution, or set of weights, over the training examples that is updated at each iteration t. This distribution is denoted $\mathbf{d}_t \in \Delta_m$, and $\mathbf{d}_t^T$ is its transpose. Here, Δ_m denotes the simplex of m-dimensional vectors with non-negative entries that sum to 1. At each iteration t, a weak classifier h_{j_t} is selected by the weak learning algorithm. The probability of error at iteration t, denoted d_-, of the selected weak classifier h_{j_t} on the training examples (weighted by the discrete distribution $\mathbf{d}_t$) is $d_- := \sum_{\{i:M_{ij_t}=-1\}} d_{t,i}$. Also, denote $d_+ := 1-d_-$. The *edge* of weak classifier j_t at time t is $r_t := (\mathbf{d}_t^T\mathbf{M})_{j_t}$, which can be written $r_t = (\mathbf{d}_t^T\mathbf{M})_{j_t} = d_+ - d_- = 1 - 2d_-$. Thus, a larger edge indicates a lower probability of error. Note that $d_+ = (1+r_t)/2$ and $d_- = (1-r_t)/2$. Here, AdaBoost in the *optimal* case means that the best weak classifier is chosen at every iteration: $j_t \in \operatorname{argmax}_j(\mathbf{d}_t^T\mathbf{M})_j$, while AdaBoost in the *non-optimal* case means that any good enough weak classifier is chosen: $j_t \in \{j : (\mathbf{d}_t^T\mathbf{M})_j \geq \rho\}$. The case of *bounded edges* is a subset of the non-optimal case for some $\bar{\rho} \geq \rho$ and $\sigma \geq 0$, namely $j_t \in \{j : \bar{\rho} \leq (\mathbf{d}_t^T\mathbf{M})_j \leq \bar{\rho} + \sigma\}$. Due to the von Neumann Min-Max Theorem for 2-player zero-sum games,

$$\min_{\mathbf{d}\in\Delta_m} \max_j (\mathbf{d}^T\mathbf{M})_j = \max_{\bar{\boldsymbol{\lambda}}\in\Delta_n} \min_i (\mathbf{M}\bar{\boldsymbol{\lambda}})_i = \rho.$$

That is, the minimum value of the maximum edge (left hand side) is the maximum margin ρ. Thus there is at least one edge available satisfying the requirement of the non-optimal case. Pseudocode for AdaBoost and arc-gv can be found in Figure 2.

Let us introduce AdaBoost's objective function $F(\boldsymbol{\lambda})$ and the Smooth Margin $G(\boldsymbol{\lambda})$. AdaBoost is a coordinate descent algorithm for minimizing

$$F(\boldsymbol{\lambda}) := \sum_{i=1}^{m} e^{-(\mathbf{M}\boldsymbol{\lambda})_i}$$

as shown by Breiman [2] and others. The smooth margin is defined by:

$$G(\boldsymbol{\lambda}) := \frac{-\ln F(\boldsymbol{\lambda})}{\|\boldsymbol{\lambda}\|_1}.$$

One can think of G as a smooth approximation of the margin. As $\|\boldsymbol{\lambda}\|_1$ becomes large, $G(\boldsymbol{\lambda})$ tends to $\mu(\boldsymbol{\lambda})$. Specifically for each iteration t, using the notation $s_t := \|\boldsymbol{\lambda}_t\|_1 = \sum_j \lambda_{t,j}$, we have (see [14]):

$$-\frac{\ln m}{s_t} + \mu(\boldsymbol{\lambda}_t) \leq G(\boldsymbol{\lambda}_t) < \mu(\boldsymbol{\lambda}_t) \leq \rho \leq r_t. \tag{2.1}$$

(1) **Input:** Matrix $\mathbf{M}$, No. of iterations $t_{\max}$

(2) **Initialize:** $\lambda_{1,j} = 0$ for $j = 1, ..., n$, also $d_{1,i} = 1/m$ for $i = 1, ..., m$, and $s_1 = 0$.

(3) **Loop for** $t = 1, ..., t_{\max}$

(a) $\left\{ \begin{array}{ll} j_t \in \operatorname{argmax}_j (\mathbf{d}_t^T \mathbf{M})_j & \text{optimal case} \\ j_t \in \{j : (\mathbf{d}_t^T \mathbf{M})_j \geq \rho\} & \text{non-optimal case} \end{array} \right\}$

(b) $r_t = (\mathbf{d}_t^T \mathbf{M})_{j_t}$

(c) $\mu_t = \mu(\boldsymbol{\lambda}_t) = \min_i (\mathbf{M}\boldsymbol{\lambda}_t)_i \Big/ s_t$

(d) $\left\{ \begin{array}{ll} \alpha_t = \frac{1}{2} \ln\left(\frac{1+r_t}{1-r_t}\right) & \text{AdaBoost} \\ \alpha_t = \frac{1}{2} \ln\left(\frac{1+r_t}{1-r_t}\right) - \frac{1}{2} \ln\left(\frac{1+\mu_t}{1-\mu_t}\right) & \text{arc-gv} \end{array} \right\}$

(e) $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \alpha_t \mathbf{e}_{j_t}$, where $\mathbf{e}_{j_t}$ is 1 in position j_t and 0 elsewhere.

(f) $s_{t+1} = s_t + \alpha_t$

(g) $d_{t+1,i} = d_{t,i} e^{-M_{ij_t}\alpha_t}/z_t$ where $z_t = \sum_{i=1}^m d_{t,i} e^{-M_{ij_t}\alpha_t}$

(4) **Output:** $\boldsymbol{\lambda}_{t_{\max}}/s_{t_{\max}}$

FIGURE 2. Pseudocode for the AdaBoost algorithm and the arc-gv algorithm.

The last two inequalities above incorporate $r_t \geq \rho$ (even in the non-optimal case), and that ρ is the largest possible value for the margin $\mu(\boldsymbol{\lambda})$.

Section 3 contains a convergence result for AdaBoost and Section 4 contains a convergence result for arc-gv; thus, we use the superscripts $^{\text{Ada}}$ and $^{\text{arc}}$ for the statements of the main theorems in Sections 3 and 4 respectively, since the sequences of $\boldsymbol{\lambda}_t$, r_t, etc., will be different for the two algorithms. The proofs for Sections 3 and 4 can be found in Sections 5 and 6 respectively. Also, we use the notation $g_t = G(\boldsymbol{\lambda}_t)$.

3. AdaBoost in the Case of Bounded Edges

We now discuss AdaBoost in the case where the edge values lie within a specific interval. That is, throughout the run of AdaBoost, our weak classifiers always have edges within the interval $[\bar{\rho}, \bar{\rho}+\sigma]$ where $\bar{\rho} \geq \rho$. As $\bar{\rho} \to \rho$ and $\sigma \to 0$ we approach the most extreme non-optimal case. The justification for allowing a range of possible edge values is practical rather than theoretical; a weak learning algorithm will probably not be able to achieve an edge of exactly $\bar{\rho}$ at every iteration since the number of training examples is finite, and since the edge is a combinatorial quantity. Thus, we assume only that the edge is within a given interval rather than an exact value. We elaborate on this observation in Theorem 3.2 later. In practice, if the number of weak classifiers is reasonably large, it is likely that the edges can be chosen to be within a specific interval if desired (as we will experimentally show).

Here is our first main result; it shows that in the case of bounded edges, the margins also fall within a small interval asymptotically.

THEOREM 3.1 (Convergence of AdaBoost with Bounded Edges). *Assume that for each t, AdaBoost's weak learning algorithm achieves an edge r_t^{Ada} such that $r_t^{Ada} \in [\bar{\rho}, \bar{\rho}+\sigma]$ for some $\rho \leq \bar{\rho} < 1$ and for some $\sigma > 0$. Then,*

$$\begin{aligned} \limsup_{t\to\infty} g_t^{Ada} &\leq \Upsilon(\bar{\rho}+\sigma), \text{ and} \\ \liminf_{t\to\infty} g_t^{Ada} &\geq \Upsilon(\bar{\rho}), \end{aligned}$$

where Υ is defined in (1.1). For the special case $\lim_{t\to\infty} r_t^{Ada} = \rho$, this implies $\lim_{t\to\infty} g_t^{Ada} = \lim_{t\to\infty} \mu(\boldsymbol{\lambda}_t^{Ada}) = \Upsilon(\rho)$.

This theorem gives an explicit small range for the margin $\mu(\boldsymbol{\lambda}_t^{\mathrm{Ada}})$, since for AdaBoost, $\lim_{t\to\infty}(g_t^{\mathrm{Ada}} - \mu(\boldsymbol{\lambda}_t^{\mathrm{Ada}})) = 0$, i.e., the limiting smooth margin is the limiting true margin.[1] Thus, we have provided a direct relationship between the performance of the weak learning algorithm (measured by the edge) and the asymptotic margin.

The special case $\lim_{t\to\infty} r_t^{\mathrm{Ada}} = \rho$ shows the tightness of the bound of Rätsch and Warmuth [10, 11]. Their result, proved in [9], which we summarize only for AdaBoost rather than for the slightly more general AdaBoost$_\varrho$, states that $\liminf_{t\to\infty} \mu(\boldsymbol{\lambda}_t^{\mathrm{Ada}}) \geq \Upsilon(r_{\inf})$, where $r_{\inf} = \inf_t r_t^{\mathrm{Ada}}$.[2] Theorem 3.1 gives bounds from both above and below, so we now have a much more explicit convergence property of the margin.

In practice, the theorem can be directly applied to estimate a range for AdaBoost's margin on the fly. The value of σ does not need to be small; the proof holds for any interval. So, as long as the edge values do not become unbounded, one can estimate σ and $\bar{\rho}$, and then use the theorem to provide a range for the asymptotic margin. Furthermore, the proof depends only on the asymptotic regime, so the first few edge values can be disregarded in the estimations. Sometimes, σ is very small in practice, in which case Theorem 3.1 specifies the margin value with high precision.

The following theorem shows that AdaBoost is capable of producing all possible values of the margin. Specifically, Theorem 3.2 below shows that Theorem 3.1 can be realized even for arbitrarily small interval size σ. Since we can obtain edges within any arbitrarily small interval, Theorem 3.1 tells us the convergence bound can be made arbitrarily tight. That is, we can coerce AdaBoost into producing whatever asymptotic margin we wish, with whatever precision we wish, as long as the training data and weak classifiers are carefully constructed.

THEOREM 3.2 (Bound of Theorem 3.1 is Non-Vacuous for any σ and $\bar{\rho}$). *Say we are given $0 < \bar{\rho} < 1$ and $\sigma > 0$ arbitrarily small. Then there is some matrix $\boldsymbol{M}$ for which non-optimal AdaBoost may choose an infinite sequence of weak classifiers with edge values in the interval $[\bar{\rho}, \bar{\rho}+\sigma]$. Additionally for this matrix $\boldsymbol{M}$, we have $\bar{\rho} \geq \rho$ (where ρ is the maximum margin for $\boldsymbol{M}$).*

[1] This is not difficult to show using (2.1); we have only to show that $\lim_{t\to\infty} s_t^{\mathrm{Ada}} \to \infty$, which is always true for AdaBoost in the separable case since s_t^{Ada} increases by at least $\tanh^{-1}\rho > 0$ at every iteration.

[2] The statement of their theorem seems to assume the existence of a combined hypothesis and limiting margin, but we believe these strong assumptions are not necessary, and that their proof of the lower bound holds without these assumptions.

The proof is by explicit construction, in which the number of examples and weak classifiers increases as more precise bounds are required, i.e., as the precision width parameter σ decreases.

Let us see Theorem 3.1 in action. Now that one can more-or-less pre-determine the value of AdaBoost's margin simply by choosing the edge values to be within a small range, one might again consider the question of whether AdaBoost's asymptotic margin matters for generalization. To study this empirically, we use AdaBoost only, several times on the same data set with the same set of weak classifiers. Our results show that the choice of edge value (and thus the asymptotic margin) does have a dramatic effect on the test error. Artificial test data for Figure 3 was designed as follows: 300 examples were constructed randomly such that each $\mathbf{x}_i$ lies on a corner of the hypercube $\{-1,1\}^{800}$. The labels are: $y_i = \text{sign}(\sum_{k=1}^{51} \mathbf{x}_i(k))$, where $\mathbf{x}_i(k)$ indicates the k^{th} component of $\mathbf{x}_i$. For $j = 1, ..., 800$, the j^{th} weak classifier is $h_j(\mathbf{x}) = \mathbf{x}(j)$, thus $M_{ij} = y_i \mathbf{x}_i(j)$. For $801 \leq j \leq 1600$, $h_j = -h_{(j-800)}$. There were 10,000 identically distributed randomly generated examples used for testing. The hypothesis space must be the same for each trial as a control; we purposely did not restrict the space via regularization (e.g., norm regulation, early stopping, or pruning). Hence we have a controlled experiment where only the choice of weak classifier is different, and this directly determines the margin via Theorem 3.1. AdaBoost was run 9 times on this dataset, each time for $t_{\max} = 3000$ iterations; the first time with standard optimal-case AdaBoost, and 8 times with non-optimal AdaBoost. For each non-optimal trial, we selected a 'goal' edge value r_{goal} (the 8 goal edge values were equally spaced). The weak learning algorithm chooses the closest possible edge to that goal. In this way, AdaBoost's margin is close to $\Upsilon(r_{\text{goal}})$. The results are shown in Figure 3B, which shows test error versus margins for the asymptotic regime of optimal AdaBoost (lower scattered curve) and the last 250 iterations for each non-optimal trial (the 8 clumps, each containing 250 points). It is very clear that as the margin increases, the probability of error decreases, and optimal AdaBoost has the lowest probability of error.

Note that the *asymptotic* margin is not the whole story; optimal AdaBoost yields a lower probability of error even before the asymptotic regime was reached. Thus, it is the degree of "optimal-ness" of the weak learning algorithm (directly controlling the asymptotic margin) that is inversely correlated with the probability of error for AdaBoost.

4. A Convergence Rate for arc-gv

We now give a convergence rate for arc-gv. We do not know of any general (non-specialized) convergence rate results that would allow us to find such a convergence rate; our proof follows the outline of the (specialized) convergence proof for the Smooth Margin Boosting Algorithms given in [14]. arc-gv is defined as in Figure 2, where the update in Step 3d uses α_t^{arc}:

$$\alpha_t^{\text{arc}} = \frac{1}{2}\ln\left(\frac{1+r_t^{\text{arc}}}{1-r_t^{\text{arc}}}\right) - \frac{1}{2}\ln\left(\frac{1+\mu_t^{\text{arc}}}{1-\mu_t^{\text{arc}}}\right), \text{ where } \mu_t^{\text{arc}} := \mu(\boldsymbol{\lambda}_t^{\text{arc}}).$$

(Note that we are using Breiman's original formulation of arc-gv, not Meir and Rätsch's variation.) Note that α_t^{arc} is non-negative since $\mu_t^{\text{arc}} \leq \rho \leq r_t^{\text{arc}}$. We start our calculation from when the smooth margin is positive; if the data is separable,

one can always use AdaBoost until the smooth margin is positive (see [15]). We denote by $\tilde{1}$ the first iteration where G is positive, so $g_{\tilde{1}}^{\text{arc}} > 0$. Here is our guaranteed convergence rate:

THEOREM 4.1 (Convergence Rate for arc-gv). *Let $\tilde{1}$ be the iteration at which G becomes positive. Then $\max_{\{\ell=\tilde{1},\ldots,t\}} \mu(\boldsymbol{\lambda}_\ell^{arc})$ will be within ϵ of the maximum margin ρ within at most*

$$\tilde{1} + \left(s_{\tilde{1}}^{arc} + \ln 2\right) \epsilon^{-(3-\rho)/(1-\rho)}$$

iterations, for arc-gv.

The fact that arc-gv makes progress with respect to the smooth margin at each iteration is our most useful tool for the convergence proof, especially since the margin itself does not necessarily increase at each iteration. Now that we have stated our main results, we move onto the proofs.

5. Proofs of Theorems 3.1 and 3.2

We drop superscripts $^{\text{Ada}}$ for this section. We will first write recursive equations for F and G as in [14]. Define: $\gamma_t := \tanh^{-1} r_t$. The recursive equation for F is:

$$F(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t}) = \frac{\cosh\gamma_t \cosh\alpha - \sinh\gamma_t \sinh\alpha}{\cosh\gamma_t} F(\boldsymbol{\lambda}_t) = \frac{\cosh(\gamma_t - \alpha)}{\cosh\gamma_t} F(\boldsymbol{\lambda}_t). \tag{5.1}$$

The recursive equation for G comes from this directly (see [14]):

$$(s_t+\alpha)G(\boldsymbol{\lambda}_t+\alpha\mathbf{e}_{j_t}) = s_t G(\boldsymbol{\lambda}_t) + \ln\left(\frac{\cosh\gamma_t}{\cosh(\gamma_t-\alpha)}\right) = s_t G(\boldsymbol{\lambda}_t) + \int_{\gamma_t-\alpha}^{\gamma_t} \tanh u \, du. \tag{5.2}$$

PROOF. (Of Theorem 3.1) Choose $\delta > 0$ arbitrarily small. We shall prove that $\limsup_t g_t \le \Upsilon(\bar{\rho}+\sigma) + \delta$ and $\liminf_t g_t \ge \Upsilon(\bar{\rho}) - \delta$, which (since δ was arbitrarily small) would prove the theorem. Starting with (5.2), subtracting $\alpha_t g_t$ from both sides yields $s_{t+1}(g_{t+1} - g_t) = \Upsilon(r_t)\alpha_t - \alpha_t g_t$, and dividing by s_{t+1},

$$g_{t+1} - g_t = (\Upsilon(r_t) - g_t)\frac{\alpha_t}{s_{t+1}}. \tag{5.3}$$

First we will show that, for some t, if g_t is smaller than $\Upsilon(\bar{\rho}) - \delta$, then g_t must monotonically increase for $\tilde{t} \ge t$ until $g_{\tilde{t}}$ meets $\Upsilon(\bar{\rho}) - \delta$ after a finite number of steps. Suppose g_t is smaller than $\Upsilon(\bar{\rho}) - \delta$, and moreover suppose this is true for N iterations: $\Upsilon(\bar{\rho}) - g_{\tilde{t}} > \delta > 0$, for $\tilde{t} \in \{t, t+1, t+2, ..., t+N\}$. Then, since $\Upsilon(r_{\tilde{t}}) \ge \Upsilon(\bar{\rho})$, we have

$$g_{\tilde{t}+1} - g_{\tilde{t}} > \delta\frac{\alpha_{\tilde{t}}}{s_{\tilde{t}+1}} \ge \delta\frac{\tanh^{-1}\bar{\rho}}{\tanh^{-1}(\bar{\rho}+\sigma)}\frac{1}{\tilde{t}+1},$$

which is strictly positive, so values of $g_{\tilde{t}}$ are strictly increasing. For the last inequality, we have used that $\tanh^{-1}\rho \le \alpha_t \le \tanh^{-1}(\rho+\sigma)$ since $\rho \le r_t \le \rho+\sigma$ and $\alpha_t = \tanh^{-1} r_t$. Since the sum of $1/(1+\tilde{t})$ eventually exceeds any value, and since $1 \ge g_{t+N} - g_t$, a recursive argument yields that N must be finite. An identical argument can be made to show that if $g_t - \Upsilon(\bar{\rho}+\sigma) > \delta > 0$, then the values of $g_{\tilde{t}}$, for $\tilde{t} \ge t$ will monotonically decrease to meet $\Upsilon(\bar{\rho}+\sigma) + \delta$ after a finite number of iterations. To summarize, the sequence of values of g_t cannot remain below $\Upsilon(\bar{\rho}) - \delta$, and cannot remain above $\Upsilon(\bar{\rho}+\sigma) + \delta$.

Next we show that from some t_0 onwards, the g_t's cannot even leave the interval $[\Upsilon(\bar{\rho}) - \delta, \Upsilon(\bar{\rho}+\sigma) + \delta]$. First,

$$|g_{t+1} - g_t| = |\Upsilon(r_t) - g_t| \frac{\alpha_t}{s_{t+1}} \leq \max(\Upsilon(\bar{\rho}+\sigma), 1) \frac{\tanh^{-1}(\bar{\rho}+\sigma)}{\tanh^{-1}\bar{\rho}} \frac{1}{t+1} =: C_\sigma \frac{1}{t+1}.$$

Now, if $t \geq C_\sigma[\Upsilon(\bar{\rho}+\sigma) - \Upsilon(\bar{\rho}) + \delta]^{-1} =: T_1$, then the bound we just proved implies that the g_t for $t \geq T_1$ cannot jump from values below $\Upsilon(\bar{\rho}) - \delta$ to values above $\Upsilon(\bar{\rho}+\sigma) + \delta$ in one time step. Since we know that the g_t cannot remain below $\Upsilon(\bar{\rho}) - \delta$ or above $\Upsilon(\bar{\rho}) + \delta$ for more than a finite number of consecutive steps, it follows that for $t \geq T_1$, the g_t must return to $[\Upsilon(\bar{\rho}) - \delta, \Upsilon(\bar{\rho}+\sigma) + \delta]$ infinitely often. Pick $t_0 \geq T_1$ so that $g_{t_0} \in [\Upsilon(\bar{\rho}) - \delta, \Upsilon(\bar{\rho}+\sigma) + \delta]$. We distinguish three cases: $g_{t_0} < \Upsilon(\bar{\rho})$, $\Upsilon(\bar{\rho}) \leq g_{t_0} \leq \Upsilon(\bar{\rho}+\sigma)$, and $g_{t_0} > \Upsilon(\bar{\rho}+\sigma)$. In the first case, we know from (5.3) that $g_{t_0+1} - g_{t_0} > 0$, so that

$$g_{t_0} < g_{t_0+1} \leq g_{t_0} + C_\sigma \frac{1}{t_0+1} \leq \Upsilon(\bar{\rho}) + \Upsilon(\bar{\rho}+\sigma) - \Upsilon(\bar{\rho}) + \delta,$$

i.e., $g_{t_0+1} \in [\Upsilon(\bar{\rho}) - \delta, \Upsilon(\bar{\rho}+\sigma) + \delta]$. A similar argument applies to the third case. In the middle case:

$$\begin{aligned} \mathrm{dist}(g_{t_0+1}, [\Upsilon(\bar{\rho}), \Upsilon(\bar{\rho}+\sigma)]) &:= \max(0, g_{t_0+1} - \Upsilon(\bar{\rho}+\sigma), \Upsilon(\bar{\rho}) - g_{t_0+1}) \\ &\leq |g_{t_0+1} - g_{t_0}| \leq \frac{C_\sigma}{t_0+1}, \end{aligned}$$

which does not exceed δ if $t_0 \geq C_\sigma \delta^{-1} =: T_2$. It follows that if $t_0 \geq T_0 := \max(T_1, T_2)$, and $g_{t_0} \in [\Upsilon(\bar{\rho}) - \delta, \Upsilon(\bar{\rho}+\sigma) + \delta]$, then g_{t_0+1} will likewise be in $[\Upsilon(\bar{\rho}) - \delta, \Upsilon(\bar{\rho}+\sigma) + \delta]$. By induction we obtain that $g_t \in [\Upsilon(\bar{\rho}) - \delta, \Upsilon(\bar{\rho}+\sigma) + \delta]$ for all $t \geq t_0$. This implies

$$\liminf_{t\to\infty} g_t \geq \Upsilon(\bar{\rho}) - \delta, \text{ and } \limsup_{t\to\infty} g_t \leq \Upsilon(\bar{\rho}+\sigma) + \delta.$$

Since, at the start of this proof, $\delta > 0$ could be chosen arbitrarily small, we obtain $\liminf_{t\to\infty} g_t \geq \Upsilon(\bar{\rho})$, and $\limsup_{t\to\infty} g_t \leq \Upsilon(\bar{\rho}+\sigma)$.

Note that we do not really need uniform bounds on r_t for this proof to work. In fact, we need only bounds that hold "eventually", so it is sufficient that $\limsup_t r_t \leq \bar{\rho} + \sigma$, $\liminf_t r_t \geq \bar{\rho}$. In the special case where $\lim_t r_t = \rho$, i.e., where $\sigma = 0$ and $\bar{\rho} = \rho$, it then follows that $\lim_t g_t = \Upsilon(\rho)$. Hence we have completed the proof. □

PROOF. (Of Theorem 3.2) For any given $\bar{\rho}$ and σ (arbitrarily small), we will create a matrix $\mathbf{M}$ such that edge values can always be chosen within $[\bar{\rho}, \bar{\rho}+\sigma]$. For reasons that will become clear later, choose a constant ϕ such that $\phi \geq (1+\bar{\rho}+\sigma)/(1-\bar{\rho}-\sigma)$, and choose $m \geq 2\phi/\sigma$. Let $\mathbf{M}$ contain only the set of possible columns that have at most $m(\bar{\rho}+1)/2$ entries that are $+1$. (We can assume m was chosen so that this is an integer.) This completes our construction of $\mathbf{M}$. One can verify that ρ of matrix $\mathbf{M}$ obeys $\rho \leq \bar{\rho}$ (see [15] for details).

We will now describe our procedure for choosing weak classifiers, and then prove that this procedure always chooses edge values r_t within $[\bar{\rho}, \bar{\rho}+\sigma]$. As usual, for $t = 1$ we set $d_{1,i} = 1/m$ for all i. Let us describe the procedure to choose our weak classifier j_t, for iteration t. Without loss of generality, we reorder the training examples so that $d_{t,1} \geq d_{t,2} \geq \cdots \geq d_{t,m}$. We choose a weak classifier j_t that correctly classifies the first $\bar{i}$ training examples, where $\bar{i}$ is the smallest index such that $2\left(\sum_{i=1}^{\bar{i}} d_{t,i}\right) - 1 \geq \bar{\rho}$. That is, we correctly classify enough examples so that

the edge just exceeds $\bar{\rho}$. The maximum number of correctly classified examples, $\bar{i}$, will be at most $m(\bar{\rho}+1)/2$, corresponding to the case where $d_{t,1} = \cdots = d_{t,m} = 1/m$. Thus, the weak classifier we choose thankfully corresponds to a column of $\mathbf{M}$. The edge r_t is $r_t = 2\left(\sum_{i=1}^{\bar{i}} d_{t,i}\right) - 1 \geq \bar{\rho}$. We can now update AdaBoost's weight vector using the usual exponential rule. Thus, our description of the procedure is complete.

By definition, we have chosen the edge such that $\bar{\rho} \leq r_t$. We have only to show that $r_t \leq \bar{\rho} + \sigma$ for each t. The main step is to show $\phi = K_1 = K_t$ for all t, where:

$$K_t := \max\left\{\max_{i_1,i_2} \frac{d_{t,i_1}}{d_{t,i_2}}, \phi\right\}.$$

We will prove this by induction. For the base case $t = 1$, $K_1 = \max\{1, \phi\} = \phi$. In order to make calculations easier, we write AdaBoost's weight update as in [13]:

$$d_{t+1,i} = \left\{ \begin{array}{ll} \frac{d_{t,i}}{1+r_t} & \text{for} \quad i \leq \bar{i} \\ \frac{d_{t,i}}{1-r_t} & \text{for} \quad i > \bar{i} \end{array} \right\}.$$

Now for the inductive step. Assuming $\phi = K_t$, we will show that $K_{t+1} = K_t$, using the update rule above.

$$K_{t+1} \quad = \quad \left\{\frac{\max_{i_1} d_{t+1,i_1}}{\min_{i_2} d_{t+1,i_2}}, \phi\right\} = \max\left\{\frac{d_{t,1}}{d_{t,\bar{i}}}, \frac{d_{t,\bar{i}+1}}{d_{t,m}}, \frac{d_{t,1}}{d_{t,m}}\frac{1-r_t}{1+r_t}, \frac{d_{t,\bar{i}+1}}{d_{t,\bar{i}}}\frac{1+r_t}{1-r_t}, \phi\right\}.$$

By our inductive assumption, the ratios of $d_{t,i}$ values are all nicely bounded, i.e., $\frac{d_{t,1}}{d_{t,\bar{i}}} \leq K_t = \phi$, $\frac{d_{t,\bar{i}+1}}{d_{t,m}} \leq \phi$, and $\frac{d_{t,1}}{d_{t,m}} \leq \phi$. We have automatically $(1-r_t)/(1+r_t) \leq 1$. Since none of the first three terms can be greater than ϕ, they can thus be ignored. Since we have ordered the training examples, $\frac{d_{t,\bar{i}+1}}{d_{t,\bar{i}}} \leq 1$. If we can bound $(1+r_t)/(1-r_t)$ by ϕ, we will be done with the induction. We can bound the edge r_t from above, using our choice of $\bar{i}$. Namely, we chose $\bar{i}$ so that the edge exceeds $\bar{\rho}$ by the influence of at most one extra training example:

$$r_t \leq \bar{\rho} + 2\max_i d_{t,i} = \bar{\rho} + 2d_{t,1}. \tag{5.4}$$

Let us now upper bound $d_{t,1}$. By definition of K_t, we have $\frac{d_{t,1}}{d_{t,m}} \leq K_t$, and thus $d_{t,1} \leq K_t d_{t,m} \leq K_t/m \leq \phi\sigma/2\phi = \sigma/2$, where we have used $d_{t,m} = \min_i d_{t,i} \leq 1/m$ since the $\mathbf{d}_t$ vectors are normalized to 1, and $m \geq 2\phi/\sigma$ as specified. Thus, (5.4) yields $r_t \leq \bar{\rho} + 2\sigma/2 = \bar{\rho} + \sigma$. So,

$$\frac{1+r_t}{1-r_t} \leq \frac{1+\bar{\rho}+\sigma}{1-\bar{\rho}-\sigma} \leq \phi.$$

Thus, $K_{t+1} = \phi$. We have just shown that for this procedure, $K_t = \phi$ for all t.

Lastly, we note that since $K_t = \phi$ for all t, we will always have $r_t \leq \bar{\rho} + \sigma$, by the upper bound for r_t we have just calculated. □

6. Proof of Theorem 4.1

We drop the superscripts $^{\text{arc}}$ for the proof. In order to prove the convergence rate, we need the two lemmas below.

LEMMA 6.1. *(Progress at Every Iteration)*

$$g_{t+1} - g_t \geq \frac{\alpha_t(r_t - g_t)}{2s_{t+1}}.$$

The proof (see [15]) uses the concavity of the function tanh, the relation $\tanh(\gamma_t - \alpha_t) = \mu_t$ which holds for arc-gv by definition of the update α_t, and the recursive equation (5.2). Since the right hand side of Lemma 6.1 is non-negative, the sequence of g_t's is non-negative and non-decreasing; arc-gv makes progress according to the smooth margin.

LEMMA 6.2. *(Step Size Bound)*

$$\alpha_t \leq c_1 + c_2 s_t \;\; \textit{and} \;\; \alpha_t \leq c_1 + c_2 s_t, \;\; \textit{where} \;\; c_1 = \frac{\ln 2}{1-\rho} \;\; \textit{and} \;\; c_2 = \frac{\rho}{1-\rho}.$$

The proof of Lemma 5 of [14] is identical to the proof of this lemma; it follows from only the recursive equation (5.2) and the non-negativity of the g_t's.

Now for the proof of Theorem 4.1. Define $\Delta G(\boldsymbol{\lambda}) := \rho - G(\boldsymbol{\lambda})$. Since (2.1) states that $g_t \leq \mu(\boldsymbol{\lambda}_t)$, we know $0 \leq \rho - \mu(\boldsymbol{\lambda}_t) \leq \rho - g_t = \Delta G(\boldsymbol{\lambda}_t)$, and thus we need only to control how fast $\Delta G(\boldsymbol{\lambda}_t) \to 0$ as $t \to \infty$. That is, if g_t is within ϵ of the maximum margin ρ, so is the margin $\mu(\boldsymbol{\lambda}_t)$. Starting from Lemma 6.1,

$$\rho - g_{t+1} \leq \rho - g_t - \frac{\alpha_t}{2s_{t+1}}(r_t - \rho + \rho - g_t), \;\; \text{thus}$$

$$\Delta G(\boldsymbol{\lambda}_{t+1}) \leq \Delta G(\boldsymbol{\lambda}_t)\left[1 - \frac{\alpha_t}{2s_{t+1}}\right] - \frac{\alpha_t(r_t - \rho)}{2s_{t+1}} \leq \Delta G(\boldsymbol{\lambda}_{\tilde{1}}) \prod_{\ell=\tilde{1}}^{t}\left[1 - \frac{\alpha_\ell}{2s_{\ell+1}}\right]. \tag{6.1}$$

We stop the recursion at $\boldsymbol{\lambda}_{\tilde{1}}$, where $\boldsymbol{\lambda}_{\tilde{1}}$ is the coefficient vector at the first iteration where G is positive. Before we continue, we upper bound the product in (6.1) the same way as in [14, 15]:

$$\prod_{\ell=\tilde{1}}^{t}\left[1 - \frac{\alpha_\ell}{2s_{\ell+1}}\right] \leq \left[\frac{s_{\tilde{1}} + \ln 2}{s_{t+1} + \ln 2}\right]^{(1-\rho)/2}. \tag{6.2}$$

It follows from (6.1) and (6.2) that:

$$s_t \leq s_t + \ln 2 \leq (s_{\tilde{1}} + \ln 2)\left[\frac{\Delta G(\boldsymbol{\lambda}_{\tilde{1}})}{\Delta G(\boldsymbol{\lambda}_t)}\right]^{2/(1-\rho)}. \tag{6.3}$$

We now have an upper bound for s_t, and we will soon have a lower bound. Define $\Delta\mu(\boldsymbol{\lambda}_t) = \rho - \mu_t$:

$$\begin{aligned} \alpha_t \geq \tanh \alpha_t &= \tanh[\gamma_t - (\gamma_t - \alpha_t)] = \frac{\tanh\gamma_t - \tanh(\gamma_t - \alpha_t)}{1 - \tanh\gamma_t \tanh(\gamma_t - \alpha_t)} \\ &= \frac{r_t - \mu_t}{1 - r_t\mu_t} \geq \frac{\rho - \mu_t}{1} = \Delta\mu(\boldsymbol{\lambda}_t). \end{aligned}$$

Thus, we have:

$$s_{t+1} = s_{\tilde{1}} + \sum_{\ell=\tilde{1}}^{t} \alpha_\ell \;\; \geq \;\; s_{\tilde{1}} + \sum_{\ell=\tilde{1}}^{t} \Delta\mu(\boldsymbol{\lambda}_\ell) \geq s_{\tilde{1}} + (t - \tilde{1} + 1) \min_{\ell \in 1,\ldots,t} \Delta\mu(\boldsymbol{\lambda}_\ell).$$

or, changing the index and using $\min_{\ell \in 1,\ldots,t-1} \Delta\mu(\boldsymbol{\lambda}_\ell) \geq \min_{\ell \in 1,\ldots,t} \Delta\mu(\boldsymbol{\lambda}_\ell)$,

$$s_t \;\; \geq \;\; s_{\tilde{1}} + (t - \tilde{1}) \min_{\ell \in 1,\ldots,t} \Delta\mu(\boldsymbol{\lambda}_\ell).$$

Combining with (6.3), keeping in mind that $\Delta G(\boldsymbol{\lambda}_t) \geq \Delta\mu(\boldsymbol{\lambda}_t) \geq \min_{\ell\in 1,\ldots,t} \Delta\mu(\boldsymbol{\lambda}_\ell)$,

$$t - \tilde{1} \leq \frac{s_t}{\min_{\ell\in 1,\ldots,t} \Delta\mu(\boldsymbol{\lambda}_\ell)} \leq \frac{(s_{\tilde{1}} + \ln 2)\left[\Delta G(\boldsymbol{\lambda}_{\tilde{1}})\right]^{2/(1-\rho)}}{\left[\min_{\ell\in 1,\ldots,t} \Delta\mu(\boldsymbol{\lambda}_\ell)\right]^{[1+2/(1-\rho)]}},$$

which means that $\min_{\ell\in 1,\ldots,t} \Delta\mu(\boldsymbol{\lambda}_\ell) \geq \epsilon$ is possible only if $t \leq \tilde{1} + (s_{\tilde{1}} + \ln 2)\epsilon^{-(3-\rho)/(1-\rho)}$. If t exceeds this value, $\min_{\ell\in 1,\ldots,t} \Delta\mu(\boldsymbol{\lambda}_\ell) < \epsilon$. In other words, if t exceeds this value, then at some prior iteration t_{prior}, it is true that $\mu(\boldsymbol{\lambda}_{t_{\text{prior}}})$ was within ϵ of ρ. This concludes the proof.

7. Conclusions

In this work, we present fundamental convergence properties for two historically important boosting algorithms. For AdaBoost, which is a difficult algorithm to analyze in general, we have presented the case of "bounded edges" for which AdaBoost's convergence can be completely understood. Specifically, we give a fundamental convergence property with respect to the margin; if the edge values (which measure the performance of the weak learning algorithm) are bounded within a small interval, then a corresponding interval exists for AdaBoost's asymptotic margin. We use this theoretical result to conduct a set of controlled experiments showing a clear relationship between the margin and the generalization error, namely that a larger margin indicates a lower error in this setting. We also prove that for any given small interval, training data and a set of weak classifiers can be constructed such that the edges will fall into this interval. That is, we can coerce AdaBoost to converge within a small interval of any given margin. For our discussion of Breiman's arc-gv, we provide what we believe is the first convergence rate of arc-gv to a maximum margin solution.

References

[1] Leo Breiman. Arcing classifiers. *Ann. Statist.*, 26(3):801–849, 1998.

[2] Leo Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11(7):1493–1517, 1999.

[3] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proc. ICML*, 2006.

[4] Harris Drucker and Corinna Cortes. Boosting decision trees. In *Advances in Neural Information Processing Systems 8*, pages 479–485, 1996.

[5] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.*, 55(1):119–139, August 1997.

[6] Adam J. Grove and Dale Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1998.

[7] R. Meir and G. Rätsch. An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, LNCS, pages 119–184. Springer Verlag, 2003.

[8] J. R. Quinlan. Bagging, boosting, and C4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730, 1996.

[9] Gunnar Rätsch. *Robust Boosting via Convex Optimization: Theory and Applications.* PhD thesis, University of Potsdam, Department of Computer Science, Potsdam, Germany, 2001.

[10] Gunnar Rätsch and Manfred Warmuth. Maximizing the margin with boosting. In *Proceedings of the Fifteenth Annual Conference on Computational Learning Theory*, pages 334–350, 2002.

[11] Gunnar Rätsch and Manfred Warmuth. Efficient margin maximizing with boosting. *Journal of Machine Learning Research*, 6:2131–2152, December 2005.

[12] Lev Reyzin and Robert E. Schapire. How boosting the margin can also boost classifier complexity. In *Proceedings of the Twenty-third International Conference on Machine Learning*, 2006.

[13] Cynthia Rudin, Ingrid Daubechies, and Robert E. Schapire. The dynamics of AdaBoost: Cyclic behavior and convergence of margins. *Journal of Machine Learning Research*, 5:1557–1595, December 2004.

[14] Cynthia Rudin, Robert E. Schapire, and Ingrid Daubechies. Boosting based on a smooth margin. In *Proceedings of the Seventeenth Annual Conference on Computational Learning Theory*, pages 502–517, 2004.

[15] Cynthia Rudin, Robert E. Schapire, and Ingrid Daubechies. Analysis of boosting algorithms using the smooth margin function. Accepted, Annals of Statistics, 2007.

[16] Robert E. Schapire. The strength of weak learnability. In *30th Annual Symposium on Foundations of Computer Science*, pages 28–33, October 1989.

[17] Robert E. Schapire. The boosting approach to machine learning: An overview. In *Nonlinear Estimation and Classification*. Springer, 2003.

[18] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Statist.*, 26(5):1651–1686, October 1998.

[19] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

Columbia University, Center for Computational Learning Systems, Interchurch Center, 475 Riverside MC 7717, New York, NY 10115

E-mail address: rudin@cs.columbia.edu

Princeton University, Department of Computer Science, 35 Olden St., Princeton, NJ 08544

E-mail address: schapire@cs.princeton.edu

Princeton University, Program in Applied and Computational Mathematics, Fine Hall, Washington Road, Princeton, NJ 08544-1000

E-mail address: ingrid@math.princeton.edu

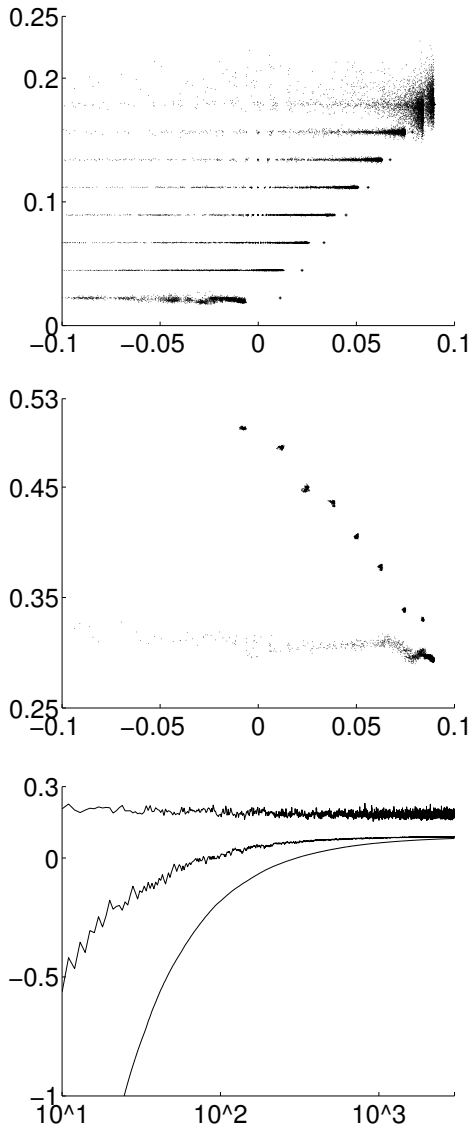

FIGURE 3. AdaBoost's probability of error on test data decreases as the margin increases. We computed 9 trials, namely, 8 trials of non-optimal AdaBoost, $\ell = 1, ..., 8$, and one trial of optimal AdaBoost (denoted via $\ell = 0$). For each non-optimal trial ℓ, a goal edge value $\overline{r_\ell}$ was manually pre-specified. For 3000 iterations of each trial, we stored the edge values $r_{\ell,t}$ and margins $\mu_{\ell,t}$ on the training set, along with the probability of error on a randomly chosen test set $e_{\ell,t}$. A - top figure: edge vs. margin. In each of the 9 trials, we plot $(\mu_{\ell,t}, r_{\ell,t})$ for iterations t that fall within the plot domain. Later iterations tend to give points nearer to the right in the plot. Additionally, dots have been placed at the points $(\Upsilon(\overline{r_\ell}), \overline{r_\ell})$ for $\ell = 1, ..., 8$. By Theorem 3.1, the asymptotic margin value for trial ℓ should be approximately $\Upsilon(\overline{r_\ell})$. Thus, AdaBoost's margins $\mu_{\ell,t}$ are converging to the pre-specified margins $\Upsilon(\overline{r_\ell})$. B - middle figure: probability of error versus margins. The lower scattered curve represents optimal AdaBoost; for optimal AdaBoost, we have plotted all $(\mu_{0,t}, e_{0,t})$ pairs falling within the plot domain. For clarity, we plot only the last 250 iterations for each non-optimal trial, i.e., for trial ℓ, there is a clump of 250 points $(\mu_{\ell,t}, e_{\ell,t})$ with margin values $\mu_{\ell,t} \approx \Upsilon(\overline{r_\ell})$. This plot shows that the probability of error decreases as the pre-specified margin increases. C - bottom figure: edges $r_{0,t}$ (top curve), margins $\mu_{0,t}$ (middle curve), and smooth margins (lower curve) versus number of iterations t for only the optimal AdaBoost trial.