Boosting with Diverse Base Classifiers

Sanjoy Dasgupta[1] and Philip M. Long[2]

[1] Department of Computer Science
University of California at San Diego
dasgupta@cs.ucsd.edu
[2] Genome Institute of Singapore
gislongp@nus.edu.sg

Abstract. We establish a new bound on the generalization error rate of the Boost-by-Majority algorithm. The bound holds when the algorithm is applied to a collection of base classifiers that contains a "diverse" subset of "good" classifiers, in a precisely defined sense. We describe cross-validation experiments that suggest that Boost-by-Majority can be the basis of a practically useful learning method, often improving on the generalization of AdaBoost on large datasets.

1 Introduction

Boosting [46, 14, 16] is an approach to training class prediction rules in which an algorithm is applied repeatedly on a variety of datasets constructed from the original dataset; an effort is made for each dataset to emphasize examples that were often classified incorrectly by rules output by previous invocations of the algorithm. After a number of rounds of this process, the class prediction rules returned (the "base classifiers") are often combined into a single rule by some kind of voting; for example, each base classifier can be assigned a weight, and the final classifier classifies an object as 1 if the total weight of the base classifiers that classify it as 1 is more than the total weight of the classifiers outputting a 0 prediction. For a wide variety of applied problems, the best algorithms known use boosting.

One interesting aspect of the behavior of boosting is that it appears to run counter to Occam's Razor, a principle that has played an important role in guiding the design of machine learning algorithms. Occam's Razor says that, all else being equal, an algorithm should prefer class prediction rules that are in some sense simple. Algorithms designed following this principle are viewed as balancing a classifier's fit to the data against its complexity.

Boosting seemed to contradict Occam's Razor because the generalization enjoyed by boosting algorithms was seen to improve as the number of rounds of boosting increased, even after the algorithm obtained zero training error [43, 10]. This improvement was seen despite the fact that the final classifier output by the boosting algorithm was getting more complex, without any accompanying decrease in training error.

Schapire, et al [47] provided an theoretical explanation of this phenomenon. This took the form of a bound on the generalization error of the boosted classifier in terms of the "margin." The margin by which a voting classifier correctly classifies an instance is the difference between the fraction of the weight voting correctly and the fraction voting incorrectly. Their bound can be paraphrased as saying that it is highly likely that if most examples in the training data are classified correctly with a large margin, then generalization will be good. They also demonstrated experimentally that AdaBoost, the main boosting algorithm, tends to improve the margin of the voting classifier during training. It has since been proved that some related algorithms maximize the margin [44, 45]. Improved bounds have also been obtained [24]. This analysis has had a substantial impact on the subsequent design and analysis of boosting algorithms.

Some experimental results suggested that it might be worth supplementing the margin analysis with alternative explanations of the generalization ability of boosting algorithms [10, 19, 36, 29, 30]. In the paper presenting the margin analysis, Schapire, et al also posed the problem of searching for alternative modes of analysis.

In this paper, we show theoretically that boosting is able to take advantage of situations in which the set of base classifiers is diverse, i.e. when there are many base classifiers that perform moderately well individually, but complement one another as sources of evidence of the correct classification of random objects. (The importance of diversity in the pool of base classifiers has been discussed in a number of papers, including [1, 34, 41, 3].)

To formalize this, we use the standard assumption that the training data, and any subsequent test data, is generated independently at random according to an underlying probability distribution $\mathbb{P}$ over instance-classification pairs. We further assume that the set H of n base classifiers used by the boosting algorithm contains a set H^* of k classifiers that are correct with probability at least $\frac{1}{2}+\gamma_*$. Finally, we assume that the random variables indicating whether the base classifiers in H^* are correct or not are mutually independent with respect to $\mathbb{P}$. We do not make any assumptions regarding the base classifiers in $H - H^*$; our analysis allows for them to depend on one another, and the classifiers in H^*, in a manner that is maximally confusing to the learning algorithm.

Note that the Hoeffding bound implies that a vote over the base classifiers in H^* is incorrect with probability at most $e^{-2\gamma_*^2 k}$. This is what can be achieved if H^* is known. We show that, with probability $1-\delta$, if γ_* is a constant, given m examples, the Boost-by-Majority algorithm [14] achieves accuracy

$$e^{-\Omega(k)} + O\left(\frac{k^{3/2}\log\frac{nk}{\delta}}{m}\right). \tag{1}$$

One can apply existing theory to show that an algorithm, essentially proposed in [4], that chooses a voting classifier to maximize the number of examples

correctly classified with a margin γ_*, which we call a MAM algorithm, achieves

$$e^{-\Omega(k)} + O\left(\frac{(\log^2 m)(\log n)\log\frac{1}{\delta}}{m}\right). \tag{2}$$

The Boost-by-Majority algorithm runs in polynomial time – in fact, for reasonable collections of base classifiers, it can be very fast. We do not address how to carry out the optimization for the MAM algorithm; it is not obvious to us how to do it efficiently.

Note that since one can always choose a smaller H^*, loosely speaking, the above bounds can be minimized over k. The second term will dominate unless k is polylogarithmic in m, so (1) and (2) are incomparable.

Note that any statistically consistent algorithm will approach accuracy at least as good as $e^{-2\gamma_*^2 k}$ with high probability as m approaches infinity. A number of algorithms related to boosting have been shown to be consistent [9, 22, 31–33, 11, 51].

In one of these papers, Lugosi and Vayatis [31] informally discussed the behavior of an algorithm related to AdaBoost in the setting of this paper. They pointed out that, as the number of examples approaches infinity, the accuracy of this algorithm approaches $e^{-2\gamma_*^2 k}$ with high probability. They did not work out a detailed bound on the error probability for a finite number m of examples; the most direct extension of their line of reasoning[3] leads to a bound, for constant γ_*, of

$$e^{-\Omega(k)} + e^{O(k)}\tilde{O}\left(\sqrt{\frac{1}{m}}\right).$$

One issue that arose during the course of performing our analysis surprised us. Our analysis leading to (1) concerns the version of Boost-by-Majority that performs Boosting-by-Filtering [14]. That is, in each round of boosting, the algorithm generates a dataset by randomly choosing a subset of the original dataset, where the probability of choosing an example depends on how many of the previously chosen base classifiers correctly classified the example. We started out trying to analyze algorithms, like AdaBoost, that evaluate base classifiers based on the total weight of the examples that are classified correctly, where the weight of an example is determined by the number of previously chosen base classifiers that got it correct. We planned to view this weighted sum as an estimate of the expected error with respect to a modified distribution over the whole domain. However, we were unable to prove (1) in this way. The trouble seemed to come at this error estimation step.

In fact, this appears to be an instance of a phenomenon that has already been discussed in the literature (see [27]). The problem of evaluating the accuracy of a base classifier in one of the later rounds of boosting can be described as estimating the expection of a random variable according to a distribution $\mathbb{Q}$, given a random sample drawn according to $\mathbb{P}$, where the function V such that

[3] In their notation, setting $\lambda = N$.

$V(x) = \mathbb{Q}(x)/\mathbb{P}(x)$ is known. Estimating the expectation of f as is done in the view of AdaBoost adopted in this paper, by drawing $x_1, ..., x_m$ according to $\mathbb{P}$, and then taking $\frac{1}{m}\sum_{i=1}^{m} Q(x_i)f(x_i)$ as the estimate, is called *importance sampling*, whereas performing this estimation as is done in Boosting-by-Filtering is called variously the *acceptance method*, the *rejection method*, and the *acceptance-rejection method*. When $\mathbb{Q}$ is quite different from $\mathbb{P}$ over much of the domain, as often happens in boosting, the estimates obtained through importance sampling are well known to suffer from high variance – the theoretical potential for this high variance is what prevented us from proving (1) for algorithms like AdaBoost that work this way.

Inspired by this observation, we experimentally evaluated a simple algorithm, similar to Boost-by-Majority, that we call BBM*. For most of the larger benchmark datasets we tried them on, BBM* generalized better than AdaBoost, providing preliminary evidence of the practical utility of boosting by filtering.

2 Preliminaries

2.1 The main model

Let $\mathcal{X}$ be a countable domain. A *class prediction rule* maps domain elements in $\mathcal{X}$ to classifications in $\{-1, 1\}$.

An algorithm is given access to a set H of n base class prediction rules. An unknown probability distribution $\mathbb{P}$ over $\mathcal{X} \times \{-1, 1\}$ is used to generate m examples $(x_1, y_1), ..., (x_m, y_m)$, which are passed to the algorithm, which uses them, together with the base classifiers H, to output a class prediction rule h.

We will analyze the Boost-by-Majority Algorithm[4] [14], which uses parameters α and T:

- Divide the m examples into T bins: put the first $\left\lfloor \frac{m}{2\sqrt{T((T-1)-0)}} \right\rfloor$ in bin 0, the next $\left\lfloor \frac{m}{2\sqrt{T((T-1)-1)}} \right\rfloor$ in bin 1, ..., the next $\left\lfloor \frac{m}{2\sqrt{T((T-1)-(T-2))}} \right\rfloor$ in bin $T-2$, and the remaining examples in bin $T-1$. Denote the indices of the examples in bin t by S_t.
- For rounds $t = 0, ..., T-1$
  - for each $i \in S_t$
    - let $r_{t,i}$ be the the number of previous base classifiers h_0,...,h_{t-1} that are correct on (x_i, y_i), and
    - $w_{t,i} = \binom{T-t-1}{\lfloor \frac{T}{2} \rfloor - r_{t,i}} (\frac{1}{2}+\alpha)^{\lfloor \frac{T}{2} \rfloor - r_{t,i}} (\frac{1}{2}-\alpha)^{\lceil \frac{T}{2} \rceil - t - 1 + r_{t,i}}$,
  - let $w_{t,\max} = \max_r \binom{T-t-1}{\lfloor \frac{T}{2} \rfloor - r} (\frac{1}{2}+\alpha)^{\lfloor \frac{T}{2} \rfloor - r} (\frac{1}{2}-\alpha)^{\lceil \frac{T}{2} \rceil - t - 1 + r}$ be the largest possible value that any $w_{t,i}$ could take,

[4] We have simplified the algorithm somewhat for the purposes of our analysis, but as the spirit of the algorithm is maintained, we refer to the modified algorithm also as Boost-by-Majority.

- apply the rejection method as follows, where $a_{t,i}$ is interpreted as indicating whether example i was accepted: for each $i \in S_t$,
  * choose $u_{t,i}$ uniformly from $[0,1]$,
  * let $a_{t,i} = \begin{cases} 1 \text{ if } u_{t,i} \leq \frac{w_{t,i}}{w_{t,\max}} \\ 0 \text{ otherwise.} \end{cases}$
- choose a base classifier h_t from H to maximize the number of examples in the filtered dataset that are classified correctly: $\{i \in S_t : a_{t,i} = 1 \text{ and } h(x_i) = y_i\}$.

– Output the classifier obtained by taking a majority vote over $h_0, ..., h_{T-1}$.

2.2 Correctness functions

For a class prediction rule h, its associated *correctness function* is an indicator function r_h that tells, for a given pair (x, y), whether it is the case that $h(x) = y$; r_h evaluates to 1 in that case, and 0 otherwise. When h is defined over a domain with an associated probability distribution, we will naturally refer to r_h as its correctness random variable.

2.3 Main result

Theorem 1. *Fix a constant $\gamma_* > 0$. Suppose the set H of base classifiers has a subset H^* of k base classifiers*

– *whose associated correctness random variables are mutually independent with respect to the underlying distribution $\mathbb{P}$, and*
– *each of which is correct with probability at least $1/2 + \gamma_*$ (again, with respect to $\mathbb{P}$).*

Then if the Boost-by-Majority Algorithm is run with $T = k$ and $\alpha = \frac{\gamma_*}{8}$, there are constants $c_1, c_2 > 0$ such that, for any underlying probability distribution $\mathbb{P}$, with probability at least $1 - \delta$, the output h of the Boost-by-Majority Algorithm applied to m random examples chosen independently according to $\mathbb{P}$ satisfies

$$\mathbb{P}(h(x) \neq y) \leq e^{-c_1 k} + \frac{c_2 k^{3/2} \log \frac{nk}{\delta}}{m}.$$

3 Some Lemmas

In this section, we establish some useful lemmas.

Since the rescaling factor for each example in each round of the Boost-by-Majority Algorithm is determined by the number of previously chosen base classifiers that classified the example correctly, we have the following.

Lemma 1. *Suppose the Boost-by-Majority algorithm is run on a dataset generated independently at random according to an underlying distribution $\mathbb{P}$. Then, for each round t, after conditioning on the examples seen before round t, the examples accepted by the algorithm in round t are mutually independent, and are distributed according to a probability distribution $\mathbb{P}_t$ over the whole of $\mathcal{X} \times \{-1, 1\}$ defined as follows.*

- $R_t(x,y) = |\{s < t : h_s(x) = y\}|$,
- $W_t(x,y) = \binom{T-t-1}{\lfloor \frac{T}{2} \rfloor - R_t(x,y)} (\frac{1}{2}+\alpha)^{\lfloor \frac{T}{2} \rfloor - R_t(x,y)} (\frac{1}{2}-\alpha)^{\lceil \frac{T}{2} \rceil - t - 1 + R_t(x,y)}$
- $\mathbb{P}_t(x,y) = W_t(x,y)\mathbb{P}(x,y)/Z_t$, *where* Z_t *is chosen so that* $\mathbb{P}_t$ *is a probability distribution.*

The following lemma will be used to show that Boost-by-Majority is often able to find accurate classifiers in rounds in which the distribution $\mathbb{P}_t$ is not too different from $\mathbb{P}$. It uses one known probabilistic method trick [42, 20, 47, 4].

Lemma 2. *Suppose* $\mathbb{P}$ *satifies the requirements of Theorem 1: there is a subset* H^* *of* k *elements of* H *whose correctness random variables are mutually independent with respect to* $\mathbb{P}$*, and each of which are correct with probability at least* $\frac{1}{2}+\gamma_*$*, where* $\gamma_* > 0$*. For any probability distribution* $\mathbb{Q}$ *such that for all* $(x,y) \in X \times \{-1,1\}$*,*

$$\mathbb{Q}(x,y) \leq \frac{\gamma_*}{3} e^{\gamma_*^2 k/2} \mathbb{P}(x,y),$$

there is a $g \in H^*$ *such that*

$$\mathbb{Q}(g(x) = y) \geq \frac{1}{2} + \frac{\gamma_*}{4}.$$

Proof. Let $g_1, ..., g_k$ (the "good" ones) be the elements of H^*. For each $i \in \{1, ..., k\}$, denote r_{g_i}, the correctness random variable for g_i, simply by r_i. The mutual independence of the r_i's with respect to $\mathbb{P}$, together with the Hoeffding bound, implies

$$\mathbb{P}\left(\frac{1}{k}\sum_{i=1}^{k} r_i < \frac{1}{2} + \frac{\gamma_*}{2}\right) \leq e^{-\gamma_*^2 k/2}. \tag{3}$$

Let us refer to the set of pairs $(x,y) \in \mathcal{X} \times \{-1,1\}$ on which $\frac{1}{k}\sum_{i=1}^{k} r_i \geq \frac{1}{2} + \frac{\gamma_*}{2}$ as the good *examples*, and call them U. The complement are the bad examples B.

First, we claim that for *any* probability distribution $\mathbb{R}$ for which $\mathbb{R}(B) = 0$, there is an $i \in \{1, ..., k\}$ such that $\mathbb{R}(r_i = 1) \geq \frac{1}{2} + \frac{\gamma_*}{2}$. For all examples in U, $\frac{1}{k}\sum_{i=1}^{k} r_i \geq \frac{1}{2} + \frac{\gamma_*}{2}$ so $\mathbf{E}_{(x,y)\sim\mathbb{R}}(\frac{1}{k}\sum_{i=1}^{k} r_i) \geq \frac{1}{2} + \frac{\gamma_*}{2}$ which implies $\frac{1}{k}\sum_{i=1}^{k} \mathbb{R}(r_i = 1) \geq \frac{1}{2} + \frac{\gamma_*}{2}$. So there is an i such that $\mathbb{R}(r_i = 1) \geq \frac{1}{2} + \frac{\gamma_*}{2}$.

The above claim implies that there is an i such that $\mathbb{Q}(g_i(x) = y|U) \geq \frac{1}{2} + \frac{\gamma_*}{2}$. Fix such an i. Then

$$\begin{aligned}
\mathbb{Q}(g_i(x) = y) &\geq \mathbb{Q}(g_i(x) = y|U)\mathbb{Q}(U) \\
&\geq \left(\frac{1}{2} + \gamma_*/2\right)\mathbb{Q}(U) \\
&= \left(\frac{1}{2} + \gamma_*/2\right)(1 - \mathbb{Q}(B)) \\
&\geq \left(\frac{1}{2} + \gamma_*/2\right)\left(1 - \frac{\gamma_*}{3} e^{\gamma_*^2 k/2}\mathbb{P}(B)\right)
\end{aligned}$$

$$\geq \left(\frac{1}{2} + \gamma_*/2\right)\left(1 - \frac{\gamma_*}{3}\right) \qquad \text{(by (3))}$$
$$\geq \frac{1}{2} + \gamma_*/4,$$

completing the proof. □

For the most part, Freund's original analysis [14] can take us the rest of the way to prove Theorem 1. For completeness, we provide the details of how.

Lemma 3. *Suppose the Boost-by-Majority is run with parameters α and T, and generates classifiers $h_0, ..., h_{T-1}$ for which*

$$\mathbb{P}_0(h_0(x) = y) \;=\; \frac{1}{2} + \gamma_0, ..., \mathbb{P}_{T-1}(h_{T-1}(x) = y) \;=\; \frac{1}{2} + \gamma_{T-1}.$$

Then, for a random element of $\mathbb{P}$, a majority vote over the predictions of the base classifiers $h_0, ..., h_{T-1}$ is incorrect with probability at most

$$e^{-2\alpha^2 T} + \sum_{t=0}^{T-1} (\alpha - \gamma_t) Z_t.$$

Proof. Define $B(t, R)$ recursively as follows:

$$B(T, R) = \begin{cases} 1 \text{ if } R \leq T/2 \\ 0 \text{ otherwise} \end{cases}$$
$$B(t, R) = \left(\frac{1}{2} - \alpha\right) B(t+1, R) + \left(\frac{1}{2} + \alpha\right) B(t+1, R+1).$$

The following is equivalent:

$$B(t, R) = \sum_{i=0}^{\lfloor T/2 \rfloor - R} \binom{T-t}{i} \left(\frac{1}{2} + \alpha\right)^i \left(\frac{1}{2} - \alpha\right)^{T-t-i}.$$

Define the potential Φ_t to be

$$\Phi_t = \sum_{(x,y) \in \mathcal{X} \times \{-1,1\}} \mathbb{P}(x, y) B(t, R_t(x, y)).$$

The Hoeffding bound implies

$$\Phi_0 \leq e^{-2\alpha^2 T}. \tag{4}$$

Freund [14, Lemma 3.7] proved that

$$\Phi_{t+1} = \Phi_t + (\alpha - \gamma_t) Z_t. \tag{5}$$

Finally, the probability with respect to $\mathbb{P} = \mathbb{P}_0$ that a majority vote over $h_0, ..., h_{T-1}$ is wrong can be rewritten as $\mathbb{P}_0(R_T(x, y) \leq T/2)$, which is Φ_T. Putting this together with (4) and (5) completes the proof. □

Lemma 4 (Lemma 3.9 of [14]). *For all iterations $t \leq T - 2$, $w_{t,\max} < \frac{2}{\sqrt{T-1-t}}$.*

4 Proof of Theorem 1

Recall that $T = k$ and $\alpha = \gamma_*/8$. Since the theorem is vacuously true if $4k < m$, we can assume without loss of generality that $m \geq 4k$, which means that the rejection method is applied to at least $\left\lfloor \frac{m}{4\sqrt{T((T-1)-t)}} \right\rfloor \geq \frac{m}{8\sqrt{T((T-1)-t)}}$ examples in each round $t \leq T-2$.

Let

$$\epsilon = \max\left\{ 4096\frac{T^{3/2}\left(\log\frac{nT}{2\delta}\right)}{\gamma_*^2 m}, \frac{3T}{\gamma_*}e^{-\gamma_*^2 T/2} \right\}. \tag{6}$$

Solving for m, we get

$$m \geq 4096\frac{T^{3/2}(\log\frac{nT}{2\delta})}{\gamma_*^2\epsilon}. \tag{7}$$

Lemma 5. *If $T = k$, for any t such that $Z_t > \epsilon/T$, with probability at least $1 - \delta/T$, $\gamma_t \geq \gamma_*/8 (= \alpha)$.*

Proof. We will give the details assuming $t \leq T-2$. The case $t = T-1$ can be proved similarly.

Let m_t be the number of examples accepted in round t. Since the probability that an example is accepted is $Z_t/w_{t,\max}$, the standard Chernoff bound implies the probability that $m_t < \frac{Z_t m}{16 w_{t,\max}\sqrt{T((T-1)-t)}}$ is at most

$$\exp\left(-\frac{Z_t m}{64 w_{t,\max}\sqrt{T((T-1)-t)}}\right).$$

Since, by assumption, $Z_t > \epsilon/T$, and, by Lemma 4, $w_{t,\max} < \frac{2}{\sqrt{T-1-t}}$, (7) implies that this probability is at most $\delta/(2T)$.

The definition of ϵ, (6), implies that

$$Z_t > \epsilon/T \geq \frac{3}{\gamma_*}e^{-\gamma_*^2 k/2}.$$

This implies that for all (x, y),

$$\mathbb{P}_t(x,y)/\mathbb{P}(x,y) \leq 1/Z_t \leq \frac{\gamma_*}{3}e^{\gamma_*^2 k/2}.$$

Applying Lemma 2, there is a base classifier $h_t^* \in H$ such that

$$\mathbb{P}_t(h_t^*(x) = y) \geq \frac{1}{2} + \frac{\gamma_*}{4}.$$

For each t, let $\hat{\mathbb{P}}_t$ be the empirical distribution over the examples (x_i, y_i) in the filtered dataset of round t, i.e. the examples such that $a_{t,i} = 1$. Then

$$\Pr(\gamma_t < \alpha)$$

$$
\begin{aligned}
&= \Pr\left(\gamma_t < \frac{\gamma_*}{8}\right) \\
&\leq \Pr\left(\hat{\mathbb{P}}_t(h_t^*(x) = y) - \mathbb{P}_t(h_t^*(x) = y) > \frac{\gamma_*}{16}\right. \\
&\qquad\qquad \left. \text{or } \ \mathbb{P}_t(h_t(x) = y) - \hat{\mathbb{P}}_t(h_t(x) = y) > \frac{\gamma_*}{16}\right) \\
&\leq \Pr\left(\text{There is an } h \in H,\ |\hat{\mathbb{P}}_t(h(x) = y) - \mathbb{P}_t(h(x) = y)| \ > \ \frac{\gamma_*}{16}\right).
\end{aligned}
$$

Applying Hoeffding bounds,

$$
\Pr(\gamma_t \leq \alpha) \leq 2n \exp\left(-\frac{\gamma_*^2 m_t}{128}\right).
$$

If $m_t \geq \frac{Z_t m}{16 w_{t,\max}\sqrt{T((T-1)-t)}}$, then since $Z_t > \epsilon/T$ and $w_{t,\max} \leq 2/\sqrt{((T-1)-t)}$, we have $m_t \geq \frac{\epsilon m}{32T^{3/2}}$, and the definition of m then implies $\Pr(\gamma_t \leq \alpha) \leq \frac{\delta}{2T}$, completing the proof. □

Let us return to proving Theorem 1. Lemma 5 implies that with probability at least $1-\delta$, for every t for which $Z_t > \epsilon/T$, $\gamma_t \geq \alpha$. Thus, applying Lemma 3, with probability at least $1-\delta$

$$
\begin{aligned}
&\Pr(\text{MAJORITY}(h_0, ..., h_{T-1}) \text{ incorrect}) \\
&\leq e^{-\frac{\gamma_*^2 k}{8}} + \sum_{t=0}^{T-1} (\alpha - \gamma_t) Z_t \\
&= e^{-\frac{\gamma_*^2 k}{8}} + \left(\sum_{t: Z_t \leq \epsilon/T} (\alpha - \gamma_t) Z_t\right) + \left(\sum_{t: Z_t > \epsilon/T} (\alpha - \gamma_t) Z_t\right) \\
&\leq e^{-\frac{\gamma_*^2 k}{8}} + \epsilon + 0,
\end{aligned}
$$

completing the proof. □

5 A margin-based bound

Recall that H is formally a set of $\{-1, 1\}$-valued functions. Let $\text{co}(H)$ be the set of all convex combinations of functions in H. In this section, we analyze the algorithm that chooses f from $\text{co}(H)$ to maximize the number of examples (x_i, y_i) for which $y_i f(x_i) \geq \gamma_*$, and outputs the classifier h_f defined by $h_f(x) = \text{sign}(f(x))$. In other words, this algorithm chooses weights with which each of the classifiers in H vote in order to maximize the number of training examples that are classified correctly with a margin of γ_*. Since it maximizes agreements with a margin, let us call such an algorithm a MAM algorithm.

Our analysis of this algorithm begins with the following lemma, which is an immediate consequence of Theorems 13.9, 12.8 and 14.20 of [4] (see also [2, 5]).

Lemma 6. *Fix a constant $\gamma_* > 0$. There are positive constants c_3 and c_4 such that, for any underlying distribution $\mathbb{P}$, if $(x_1, y_1), ..., (x_m, y_m)$ are drawn independently at random according to $\mathbb{P}$, then*

$$\Pr\left(\exists f \in \mathrm{co}(H),\ \mathbb{P}_{(x,y)}(h_f(x) \neq y) > 2\frac{|\{i : y_i f(x_i) < \gamma_*\}|}{m} + \beta\right)$$
$$\leq \exp(c_3(\log^2 m)\log n - c_4\beta m).$$

We will make use of the following standard Chernoff bounds.

Lemma 7 (see [40]). *Let $\hat{p}$ be the fraction of successes in m independent Bernoulli trials with success probability p. Then*

- *if $0 < \beta \leq 1$, $\Pr(\hat{p} > (1+\beta)p) \leq e^{-\beta^2 pm/3}$,*
- *if $\beta > 1$, $\Pr(\hat{p} > (1+\beta)p) \leq e^{-(1+\beta)\ln(1+\beta)pm/4}$.*

We then easily obtain the following.

Theorem 2. *Fix a constant $\gamma_* > 0$. There are positive constants c_5 and c_6 such that, for H, $\mathbb{P}$ and γ_* satisfying the requirements of Theorem 1, with probability $1-\delta$, the output h of an MAM algorithm applied to m random examples chosen independently according to $\mathbb{P}$ satisfies*

$$\mathbb{P}(h(x) \neq y) \leq e^{-c_5 k} + \frac{c_6(\log^2 m)(\log n)\log\frac{1}{\delta}}{m}.$$

Proof. First, we claim that it is likely that there is a voting classifier that correctly classifies all but a fraction $2e^{-\gamma_*^2 k/2} + \frac{8\ln\frac{2}{\delta}}{m}$ of the training examples correctly with a margin γ_*. (Let B be the event that this does not happen.) If $f(\cdot) = \frac{1}{k}\sum_{h\in H^*} h(\cdot)$, then the Hoeffding bound implies $\mathbb{P}(yf(x) < \gamma_*) \leq e^{-\gamma_*^2 k/2}$. Applying Lemma 7 then establishes that $\Pr(B) \leq 1 - \delta/2$ (use the $\beta = 1$ bound if $m \geq 3e^{\gamma_*^2 k/2}\ln\frac{2}{\delta}$, and $\beta > 1$ bound otherwise).

Applying Lemma 6 completes the proof. □

6 Experiments

The fact that we were only able to prove Theorem 1 using Boost-by-Majority made us wonder whether an algorithm like Boost-by-Majority might perform well in practice. In this section, we describe some preliminary experiments aimed at addressing this question.

Our experiments compare the performance of AdaBoost with an algorithm we call BBM*, which is like Boost-by-Majority, but with a few changes. Both algorithms were applied in conjunction with decision stumps, and the decision stump for each attribute was chosen to minimize the empirical error.

The differences between BBM* and Boost-by-Majority are as follows:

- When run for T rounds, instead of partitioning the training data into T disjoint parts to be used in the various rounds, BBM* uses all of the examples in each round. The rejection method is applied to choose a subset of the examples in each round in a manner analogous to Boost-by-Majority.
- If the number of examples accepted in a given round is less than 5, then, in BBM*, the round is skipped: no base classifier is added to the list of voters in that round. (This is similar to the practice Freund [14] analyzed: when the number of accepted examples is too small, add a base classifier that predicts randomly. The cutoff of 5 was chosen arbitrarily and not optimized, and the same value of 5 was used on all of the datasets.)
- If more than one attribute has a decision stump that minimizes training error on the filtered dataset in a given round, then an attribute is chosen uniformly at random from the list of minimizers.
- The parameter α is chosen using 5-fold cross-validation on the training data. The values $\{0.002, 0.005, 0.01, 0.02, 0.05\}$ are tried, and the value minimizing the cross-validation error is used. In case of a tie, the geometric mean of the values attaining the minimum is used.

In our experiments, both BBM* and AdaBoost were applied in conjunction with decision stumps, and both were run for 100 rounds. We evaluated the algorithms using the protocol of Dudoit, et al [13], in which the data is randomly split 100 times into a training set with 2/3 of the examples, and a test set with 1/3 of the examples. Both algorithms were evaluated on the same 100 training-test splits, and the average test-set error was tabulated. We applied both algorithms to a list of datasets from the UC Irvine repository previously used for evaluating AdaBoost [15], together with one microarray dataset called ER (see [49, 30] for a description).

Our results are summarized in Table 1. BBM* appears to significantly improve on the performance of AdaBoost on most of the larger datasets.

The code for these experiments is a modification of the code from [30]. The site

`http://giscompute.gis.nus.edu.sg/~plong/bbm_star`

has the new code.

7 Conclusion

We have provided a theoretical analysis of boosting that shows how a boosting algorithm can take advantage of a collection of base classifiers that contains a large, diverse collection of fairly good classifiers. Inspired by this analysis, we have investigated the practical utility of an algorithm like Freund's Boost-by-Majority algorithm, and found that, on some large datasets, it appears to perform better than AdaBoost. We have also showed that a better bound can be obtained by an algorithm that maximizes the number of examples classified correctly with a certain margin, but we have not shown how to efficiently perform this optimization.

Dataset	BBM*	AdaBoost	# attrs.	# examples
ER	20.0	18.1	7129	49
promoters	10.8	9.9	57	106
hepatitis	18.2	19.2	19	155
ionosphere	11.4	10.3	34	351
house	4.2	4.0	16	435
breast	3.4	4.5	9	699
pima	25.7	24.7	8	768
hypothyroid	0.86	1.01	25	3163
sick-euthyroid	2.5	3.1	25	3163
kr-vs-kp	3.0	4.3	36	3196

Table 1. Summary of our experimental comparison between the BBM* algorithm and AdaBoost when both are applied for 100 rounds in conjunction with decision stumps. On each dataset, the percentage of test examples misclassified by each of the algorithms, together with the number of attributes and number of examples in the dataset, are shown.

It is trivial to generalize our results to the case in which the correctness random variables of the good base classifiers H^* are *negatively associated*, say in the sense studied in Dubhashi and Ranjan's [12] paper. An analysis in which a limited amount of positive association among the errors of classifiers in H^* was allowed would be interesting.

All that we use about our assumption is that it implies that there is a convex combination f of the classifiers in H such that $\mathbb{P}(yf(x) \leq \gamma_*) \leq e^{-\gamma_*^2 k/2}$. Thus, more general theorems concerning this form of assumption are implicit in our analyses. This implies that our results can also be strengthened to apply when the *average* (instead of the maximum) error rate of the classifiers in H^* is at most $1/2 - \gamma_*$.

Directly applying (1.5) from [25] (see also [24]) leads to a bound, for the MAM algorithm, of

$$c\left(e^{-\gamma_*^2 k/2} + \frac{(1/\gamma_*)^{\frac{\log_2 n}{1+\log_2 n}}}{m^{-\frac{1+\frac{\log_2 n}{2}}{1+\log_2 n}}} + \frac{\ln\frac{1}{\delta}}{m}\right).$$

If γ_* is a constant and n is moderately large, the dependence on m is roughly as $1/\sqrt{m}$. However, it seems likely that some the techniques used in [24] can be applied to improve Theorem 2, at least by a factor of $\log m$.

It would be good to prove a bound like Theorem 2 for a provably fast algorithm. One promising avenue is to try to use boosting to do the optimization, possibly approximately, for an algorithm like MAM.

In our analysis, the parameters of the Boost-by-Majority algorithm were set as a function of k and γ^*. It would be nice to be able to prove a similar theorem for an algorithm that did not need to do this. A modification of the smooth boosting algorithm studied by Gavinsky [21, 23, 18] to use boosting-by-filtering

seems a good place to start. (Similarly, the MAM algorithm used knowledge of γ_*.)

Another question is whether the bounds of this paper, or better bounds, can be obtained by an algorithm that minimizes a convex function of the voting weights that is an upper bound on the number of misclassifications, as some boosting algorithms can be seen to do (see [8, 35, 17]). Recently, significant progress has been made on the analysis of such algorithms (see [31, 50, 6, 7]). Recent strong bounds obtained for Support Vector Machines [26, 37] using the PAC-Bayes methodology [39, 38] also raise hope for that technique to be profitably applied here. Either of these would result in guarantees with the flavor of the margin analysis, as well as for the framework of this paper, for the same, efficient, algorithm.

It also appears possible that improved analysis could be obtained with an algorithm like Boost-by-Majority. For example, can improved bounds be obtained for an algorithm like BBM* that, in each round, applies the rejection method on *all* the examples?

Finally, we view BBM* as a crude first step in investigation of the practical utility of the rejection method in the context of boosting. It appears possible that sophisticated hybrids of the rejection method and importance sampling, as have been developed for other applications (see [28]), might lead to significant improvements in practical performance for boosting algorithms. Another tantalizing possibility is that recent refinements to importance sampling that reduce the variance while remaining unbiased (see [48]) might have a role to play in boosting, both in theory and in practice.

8 Acknowledgements

We are grateful to Peter Bartlett, Gábor Lugosi, David McAllester, Partha Niyogi and Adai Ramasamy for helpful discussions and email messages, including in some cases pointers to the literature and in some cases helpful comments on previous versions of this paper. We would also like to thank anonymous referees for stimulating comments.

References

1. K. M. Ali and M. J. Pazzani. Error reduction through learning multiple descriptions. *Machine Learning*, 24:173–202, 1996.
2. N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the Association for Computing Machinery*, 44(4):616–631, 1997.
3. Y. Amit and G. Blanchard. Multiple randomized classifiers: MRCL, 2001. Manuscript.
4. M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

5. P. L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.
6. P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. Technical Report 638, Department of Statistics, U.C. Berkeley, 2003.
7. G. Blanchard, G. Lugosi, and N. Vayatis. On the rate of convergence of regularized boosting methods, 2003. Manuscript.
8. L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11(7), 1999.
9. L. Breiman. Some infinity theory for predictor ensembles. Technical Report 577, Statistics Department, UC Berkeley, 2000.
10. Leo Breiman. Arcing classifiers. *The Annals of Statistics*, 1998.
11. P. Bülmann and B. Yu. Boosting with the l2 loss: regression and classification. *Journal of the American Statistical Association*, to appear.
12. D. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *Random Structures & Algorithms*, 13(2):99–124, Sept 1998.
13. S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.
14. Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.
15. Y. Freund and R. Schapire. Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996.
16. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, 1995.
17. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337–407, 2000.
18. D. Gavinsky. Optimally-smooth adaptive boosting and application to agnostic learning. *Proceedings of the 13th International Workshop on Algorithmic Learning Theory*, 2002.
19. 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.
20. A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46:129–154, 1993.
21. Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *IEEE Symposium on Foundations of Computer Science*, pages 538–545, 1995.
22. W. Jiang. Process consistency for AdaBoost. *Annals of Statistics*, to appear.
23. Adam Klivans and Rocco A. Servedio. Boosting and hard-core sets. In *IEEE Symposium on Foundations of Computer Science*, pages 624–633, 1999.
24. V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30(1), 2002.
25. V. Koltchinskii and D. Panchenko. Complexities of convex combinations and bounding the generalization error in classification, 2003. Manuscript.
26. J. Langford and J. Shawe-Taylor. PAC-bayes and margins. *NIPS, 2002*.
27. J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.
28. J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.
29. P. M. Long. Minimum majority classification and boosting. *Proceedings of the The Eighteenth National Conference on Artificial Intelligence*, 2002.

30. P. M. Long and V. B. Vega. Boosting and microarray data. *Machine Learning*, to appear.
31. G. Lugosi and N. Vayatis. On the bayes-risk consistency of regularized boosting methods. *Annals of Statistics*, 2004. Preliminary version in COLT'02.
32. S. Mannor, R. Meir, and S. Mendelson. The consistency of boosting algorithms. Manuscript, 2001.
33. S. Mannor, R. Meir, and T. Zhang. The consistency of greedy algorithms for classification. *Proc. Fifteenth Annual Conference on Computational Learning Theory*, 2002.
34. Dragos D. Margineantu and Thomas G. Dietterich. Pruning adaptive boosting. In *Proc. 14th International Conference on Machine Learning*, pages 211–218. Morgan Kaufmann, 1997.
35. L. Mason, J. Baxter, P. L. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems 12*, pages 512–518. MIT Press, 2000.
36. Llew Mason, Peter L. Bartlett, and Jonathan Baxter. Improved generalization through explicit optimization of margins. *Machine Learning*, 38(3):243–255, 2000.
37. D. McAllester. Simplified PAC-Bayesian margin bounds. *Proceedings of the 2003 Conference on Computational Learning Theory*, 2003.
38. David A. McAllester. PAC-Bayesian model averaging. In *Proc. 12th Annu. Conf. on Comput. Learning Theory*, pages 164–170. ACM Press, New York, NY, 1999.
39. David A. McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.
40. R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
41. P. Niyogi, J.-B. Pierrot, and O. Siohan. On decorrelating classifiers and combining them, 2001. Manuscript, see `people.cs.uchicago.edu/~niyogi/decorrelation.ps`.
42. G. Pisier. Remarques sur un resultat non publi'e de B. Maurey. *Sem. d'Analyse Fonctionelle*, 1(12):1980–81, 1981.
43. J. Quinlan. Bagging, boosting and c4.5. In *Proceedings of the Thirteenth National Conference on Artifiicial Intelligence*, pages 725–730. AAAI/MIT Press, 1996.
44. G. Rätsch and M. K. Warmuth. Marginal boosting. *Proceedings of the Annual Conference on Computational Learning Theory*, 2002.
45. S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *NIPS*, 2002.
46. R. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.
47. Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.
48. F. Southey, D. Schuurmans, and A. Ghodsi. Regularized greedy importance sampling. *NIPS'02*.
49. M. West, et al. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl. Acad. Sci. USA*, 98(20):11462–11467, 2001.
50. T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, to appear.
51. T. Zhang and B. Yu. Boosting with early stopping: convergence and consistency. Technical Report 635, Statistics Department, UC Berkeley, 2003.