How Boosting the Margin Can Also Boost Classifier Complexity

Lev Reyzin LEV.REYZIN@YALE.EDU
Yale University, Department of Computer Science, 51 Prospect Street, New Haven, CT 06520, USA

Robert E. Schapire SCHAPIRE@CS.PRINCETON.EDU
Princeton University, Department of Computer Science, 35 Olden Street, Princeton, NJ 08540, USA

Abstract

Boosting methods are known not to usually overfit training data even as the size of the generated classifiers becomes large. Schapire et al. attempted to explain this phenomenon in terms of the margins the classifier achieves on training examples. Later, however, Breiman cast serious doubt on this explanation by introducing a boosting algorithm, arc-gv, that can generate a higher margins distribution than AdaBoost and yet performs worse. In this paper, we take a close look at Breiman's compelling but puzzling results. Although we can reproduce his main finding, we find that the poorer performance of arc-gv can be explained by the increased complexity of the base classifiers it uses, an explanation supported by our experiments and entirely consistent with the margins theory. Thus, we find maximizing the margins is desirable, but not necessarily at the expense of other factors, especially base-classifier complexity.

1. Introduction

The AdaBoost boosting algorithm (Freund & Schapire, 1997) and most of its relatives produce classifiers that classify by voting the weighted predictions of a set of base classifiers which are generated in a series of rounds. Thus, the size — and hence, naively, the apparent complexity — of the final combined classifier used by such algorithms increases with each new round of boosting. Therefore, according to Occam's razor (Blumer et al., 1987), the principle that less complex classifiers should perform better, boosting should suffer from overfitting; that is, with many rounds of boosting, the test error should increase as the final classifier becomes overly complex. Nevertheless, it has been observed by various authors (Breiman, 1998; Drucker & Cortes, 1996; Quinlan, 1996) that boosting often tends to be resistant to this kind of overfitting, apparently in defiance of Occam's razor. That is, the test error of AdaBoost often tends to decrease well after the training error is zero, and does not increase even after a very large number of rounds.[1]

Schapire et al. (1998) attempted to explain AdaBoost's tendency not to overfit in terms of the margins of the training examples, where the *margin* is a quantity that can be interpreted as measuring the confidence in the prediction of the combined classifier. Giving both theoretical and empirical evidence, they argued that with more rounds of boosting, AdaBoost is able to increase the margins, and hence the confidence, in the predictions that are made on the training examples, and that this increase in confidence translates into better performance on test data, even if the boosting algorithm is run for many rounds.

Although Schapire et al. backed up their arguments with both theory and experiments, Breiman (1999) soon thereafter presented experiments that raised important questions about the margins explanation. Following the logic of the margins theory, Breiman attempted to design a better boosting algorithm, called arc-gv, that would provably maximize the minimum margin of any training example. He then ran experiments comparing the performance of arc-gv and AdaBoost using CART decision trees pruned to a fixed number of nodes as base classifiers. He found that arc-gv did indeed produce uniformly higher margins than AdaBoost. However, contrary to what was apparently predicted by the margins theory, he found that his new

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[1] However, in some of these cases, the test error has been observed to increase slightly after an *extremely* large number of rounds (Grove & Schuurmans, 1998).

algorithm arc-gv performed worse on test data than AdaBoost in almost every case. Breiman concluded rather convincingly that his experiments put the margins explanation into serious doubt and that a new understanding is needed.

In this paper, we take a close look at these compelling experiments to try to determine if they do in fact contradict the margins theory. In fact, the theory that was presented by Schapire et al. states that the generalization error of the final combined classifier can be upper bounded by a function that depends not only on the margins of the training examples, but also on the number of training examples and the complexity of the base classifiers (where complexity might, for instance, be measured by VC-dimension or description length). Breiman was well aware of this dependence on the complexity of the base classifiers and attempted to control for this factor in his experiments by always choosing decision trees of a fixed size. However, in our experiments, we find that there still remain important differences between the trees chosen by AdaBoost and arc-gv. Specifically, we find that the trees produced using arc-gv are considerably deeper, both in terms of maximum and average depth of the leaves. Intuitively, such deep trees are more prone to overfitting, and indeed, it is clear that the space of decision trees of a given size is much more greatly constrained when a bound is placed on the depth of the leaves. Furthermore, we find experimentally that the deeper trees generated by arc-gv are measurably more prone to overfitting than those of AdaBoost. The use of depth as a measure of tree complexity was also suggested in the work of Mason, Bartlett and Golea (2002) who worked on finding more refined ways of measuring the complexity of a decision tree besides its overall size.

Thus, we argue that the trees found by arc-gv have topologies that are more complex in terms of their tendency to lead to overfitting, and that this increase in complexity accounts for arc-gv's inferior performance on test data, an argument that is consistent with the margins theory.

We then consider the use of other base classifiers, such as decision stumps, whose complexity can be more tightly controlled. We again compare the performance of AdaBoost and arc-gv, and again find that AdaBoost is superior, despite the fact that base classifiers of equivalent complexity are being used, and despite the fact that arc-gv tends to obtain a higher minimum margin than AdaBoost. Nevertheless, on close inspection, we see that the bounds presented by Schapire et al. are in terms of the *entire* distribution of margins, not just the *minimum* margin. When this overall margin distribution is examined, we find that although arc-gv obtains a higher minimum margin, the margin distribution as a whole is very much higher for AdaBoost. Thus, again, these experiments do not appear to contradict the margins theory.

In sum, our experiments explore the complex interplay between margins, base classifier complexity and sample size that helps to determine how well a classifier performs. We believe that understanding this interaction better might help us to design better algorithms. In a sense, our results confirm Breiman's point that maximizing margins is not enough; we also need to think about the other factors, especially base classifier complexity, and how that can be driven up by an over-aggressive attempt to increase the margins. Our results also explore the interplay between minimum margin and the overall margins distribution as seen in the way that arc-gv only increases the minimum margin, but AdaBoost sometimes seems to do a better job with the overall distribution.

Our paper focuses only on Breiman's arc-gv algorithm for maximizing margins although others have been proposed, for instance, by Rätsch and Warmuth (2002), Grove and Schuurmans (1998) and Rudin, Schapire and Daubechies (2004). Moreover, Mason, Bartlett and Golea (2004) were able to show how the direct optimization of margins could indeed lead to improved performance. We also focus only on the theoretical bounds of Schapire et al., although these have been greatly improved, for instance, by Koltchinskii and Panchenko (2002). Overviews on boosting are given by Schapire (2002) and Meir and Rätsch (2003).

After reviewing the margins theory in Section 2, we begin our study in Section 3 with experiments intended to replicate those of Breiman. In Section 4, we then present evidence that arc-gv produces higher margins by using more complex base classifiers and that its poorer performance is consistent with the margins theory. In Section 5, we try to control the complexity of the base classifiers but find that this prevents arc-gv from having a uniformly higher margins distribution.

2. Algorithms and Theory

Boosting algorithms combine moderately inaccurate prediction rules and take their weighted majority vote to form a single classifier. On each round, a boosting algorithm generates a new prediction rule to use and then places more weight on the examples classified incorrectly. Hence, boosting constantly focuses on classifying correctly the examples that are the hardest

Given: $(x_1, y_1), \ldots, (x_m, y_m)$
where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train base learner using distribution D_t.
- Get base classifier $h_t : X \to \{-1, +1\}$.
- Choose $\alpha_t \in \mathbb{R}$.
- Update:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

  where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final classifier:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Figure 1. A generic algorithm equivalent to both AdaBoost and arc-gv, depending on how α_t is selected.

to classify.

Figure 1 presents a generic algorithm that is equivalent to both AdaBoost and arc-gv, depending on the choice of α_t. Specifically, AdaBoost sets α to be

$$\alpha_t = \frac{1}{2} \ln \frac{1 + \gamma_t}{1 - \gamma_t}$$

where γ_t is the so-called *edge* of h_t:

$$\gamma_t = \sum_i D_t(i) y_i h_t(x_i)$$

which is linearly related to h_t's weighted error.

AdaBoost greedily minimizes a bound on the training error of the final classifier. In particular, as shown by Schapire and Singer (1999), its training error is bounded by $\prod_t Z_t$, so, on each round, it chooses h_t and sets α to minimize Z_t, the normalizing factor.

Freund and Schapire (1997) derived an early bound on the generalization error of boosting, showing that

$$\Pr\left[H(x) \neq y\right] \leq \hat{\Pr}[H(x) \neq y] + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right)$$

where $\Pr[\cdot]$ denotes probability over the distribution that was assumed to have generated the training examples, $\hat{\Pr}[\cdot]$ denotes the empirical probability on the training sample, and d is the VC-dimension of the space of all possible base classifiers. However, this bound becomes very weak as the number of rounds T increases, and predicts that AdaBoost will quickly overfit with only a moderate number of rounds. Early experiments (Breiman, 1998; Drucker & Cortes, 1996; Quinlan, 1996), however, showed just the opposite, namely, that AdaBoost tends not to overfit.

Schapire et al.(1998) attempted to explain why boosting often does not overfit using the concept of margins on the training examples. The margin of example (x, y) depends on the votes $h_t(x)$ with weights α_t of all the hypotheses:

$$\text{margin}(x, y) = \frac{y \sum_t \alpha_t h_t(x)}{\sum_t \alpha_t}.$$

The magnitude of the margin represents the strength of agreement of the base classifiers, and its sign indicates whether the combined vote produces a correct prediction. Using the margins, Schapire et al. proved a bound not dependent on the number of boosting rounds. They showed that for any θ, the generalization error is at most

$$\hat{\Pr}[\text{margin}(x, y) \leq \theta] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right). \qquad (1)$$

We can notice that this margins bound depends most heavily on the margins near the bottom of the distribution, since having generally high smallest margins allows θ to be small without $\hat{\Pr}[\text{margin}(x, y) \leq \theta]$ getting too large.

Following this logic, Breiman (1999) designed arc-gv to greedily maximize the minimum margin. Arc-gv follows the same algorithm as AdaBoost, except for setting α_t differently:

$$\alpha_t = \frac{1}{2} \log \frac{1 + \gamma_t}{1 - \gamma_t} - \frac{1}{2} \log \frac{1 + \varrho_t}{1 - \varrho_t}$$

where ϱ_t is the minimum margin over all training examples of the combined classifier up to the current round:

$$\varrho_t = \min_i \left(y_i \frac{\sum_{s=1}^{t-1} \alpha_s h_s(x_i)}{\sum_{s=1}^{t-1} \alpha_s}\right)$$

(It is understood that $\varrho_1 = 0$.)

Arc-gv has the property[2] that its minimum margin converges to the largest possible minimum margin,

[2]Meir and Rätsch (2003) claimed they can only prove this property when taking ϱ_t to be the maximum minimum margin over all previous rounds in the equation above; we nevertheless decided to use Breiman's original formulation of arc-gv.

Table 1. Dataset sizes for training and test.

	cancer	ion	ocr 17	ocr 49	splice
training	630	315	1000	1000	1000
test	69	36	5000	5000	2175

provided that the edges are sufficiently large, as will be the case if the base classifier with largest edge is selected on every round. Thus, the margin theory would appear to predict that arc-gv's performance should be better than AdaBoost's, although as we here explore, there are other factors at play.

3. Breiman's Experiments

Breiman (1999) showed that it is possible for arc-gv to produce a higher margins distribution and yet perform worse. He ran AdaBoost and arc-gv for 100 rounds using pruned CART decision trees as base classifiers. Each such tree was created by generating the full CART tree and pruning it to the best (i.e., minimum weighted error) k-leaf subtree. Breiman's most compelling results were for trees of size $k = 16$ where he found that the margins distributions are uniformly higher for arc-gv than for AdaBoost.

We begin our study by replicating his results. However, unlike Breiman, we did not see the margins of arc-gv being significantly higher until we ran the algorithms for 500 rounds. Since Breiman's critique of the margins theory is strongest when the difference in the margins distributions is clear, we focus only on the 500-round case with $k = 16$.

We considered the following datasets: breast cancer, ionosphere, splice, ocr17, and ocr49, all available from the UCI repository. These include the same natural datasets as Breiman, except the sonar dataset, since it only includes 208 data points and thereby produces high variance in experiment. The splice dataset was modified to collapse the two splice categories into one to create binary-labeled data. Also, ocr17 and ocr49 contain randomly chosen subsets of the NIST database of handwritten digits consisting only of the digits 1 and 7, and 4 and 9 (respectively); in addition, the images have been scaled down to 14×14 pixels, each with only four intensity levels. Table 1 shows the number of training and test examples used in each. The stark differences in the training and test sizes among the datasets occur because we used the same random splits Breiman used for ionosphere and breast cancer, but the additional datasets we used had many more data points, which allowed us to use larger sets for

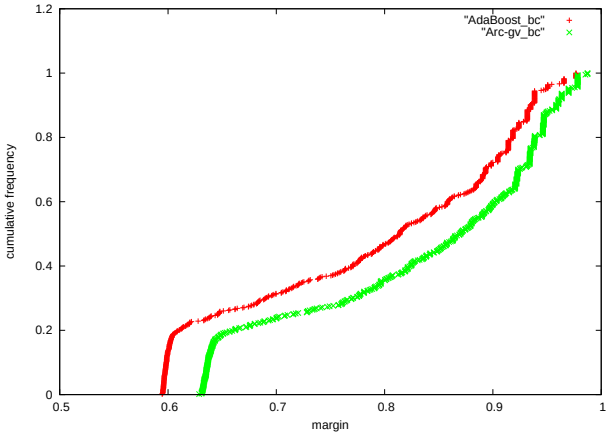

Figure 2. Cumulative margins for AdaBoost and arc-gv for the breast cancer dataset after 500 rounds of boosting.

Table 2. Test errors, averaged over 10 trials, of AdaBoost and arc-gv, run for 500 rounds using CART decision trees pruned to 16 leaf nodes as base classifiers.

	cancer	ion	ocr 17	ocr 49	splice
AdaBoost	2.46	3.46	0.96	2.04	3.18
arc-gv	3.04	7.69	1.76	2.38	3.45

test data. In running our experiments, we followed Breiman's technique of choosing an independently random subset for training data of sizes specified in the table. All experiments were repeated on ten random partitions of the data, and, in most cases, the results were averaged.

Figure 2 shows the cumulative margins distribution after 500 rounds for both AdaBoost and arc-gv on the breast cancer dataset. As observed by Breiman, arc-gv does indeed produce higher margins. These distributions are representative of the margins distributions for the rest of the datasets.

Table 2 shows the test errors for each algorithm. Again, in conformity with Breiman, we see that the test errors of arc-gv are indeed higher than those of AdaBoost.

To further visualize what is happening during the running of these algorithms, we plotted both the test error and minimum margin as a function of the number of rounds in Figures 3 and 4. These results seem to be in direct contradiction to the margins theory.

Table 3. Test errors, minimum margins, and tree depths, averaged over 10 trials, of AdaBoost and arc-gv, run for 500 rounds using CART decision trees pruned to 16 leaf nodes as base classifiers. (For 100 rounds, we also saw arc-gv producing deeper trees on average.)

	test error		minimum margin		tree depth	
	arc-gv	AdaBoost	arc-gv	AdaBoost	arc-gv	AdaBoost
breast cancer	3.04	2.46	0.64	0.61	9.71	7.86
ionosphere	7.69	3.46	0.97	0.77	8.89	7.23
ocr 17	1.76	0.96	0.95	0.88	7.47	7.41
ocr 49	2.38	2.04	0.53	0.49	7.39	6.70
splice	3.45	3.18	0.46	0.42	7.12	6.67

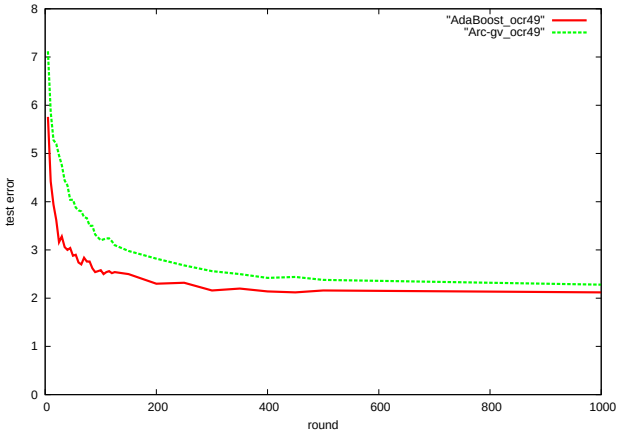

Figure 3. Test errors for AdaBoost and arc-gv for the ocr49 dataset as a function of the number of rounds of boosting.

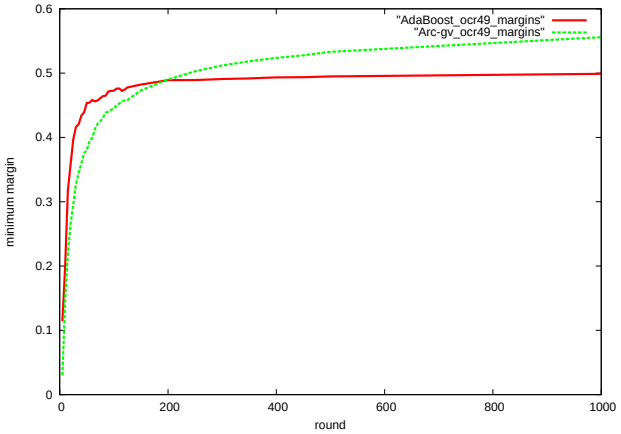

Figure 4. Minimum margins for AdaBoost and arc-gv for the ocr49 dataset as a function of the number of rounds of boosting.

4. Tree Complexity

Can these results be reconciled with the margins explanation? In fact, according to Eq. (1), there are factors other than the minimum margin that need to be considered. Specifically, the generalization error of the combined classifier depends both on the margins it generates, the size of the training sample, and on the complexity of the base classifiers. Since the size of the sample is the same for both arc-gv and AdaBoost, after recording the margins, we should examine the complexity of the base classifiers.

How can we measure the complexity of a decision tree? The most obvious measure is the number of leaves in the tree, which, like Breiman, we are already controlling by always selecting trees with exactly 16 leaves. However, even among all trees of fixed size, we claim that there remain important topological differences that affect the tendency of the trees to overfit. In particular, deeper trees make predictions based on a longer sequence of tests and therefore intuitively tend to be more specialized than shallow trees and thus more likely to overfit.

In fact, arc-gv generates significantly deeper trees than AdaBoost. Table 3 shows the average depths of the trees (measured by the maximum depth of any leaf) in addition to the minimum margin and error rates for each algorithm. We also measured the running average of the tree complexity of both algorithms as the number of rounds increased. The pattern in Figure 5 is representative of the results for most datasets. In this figure, we can see that at the beginning of boosting, the depths of the trees generated by AdaBoost converge downward to a value, while the depths of the trees generated by arc-gv continue to increase for about 200 rounds before leveling off to a higher value. It is evident that while arc-gv has higher margins and

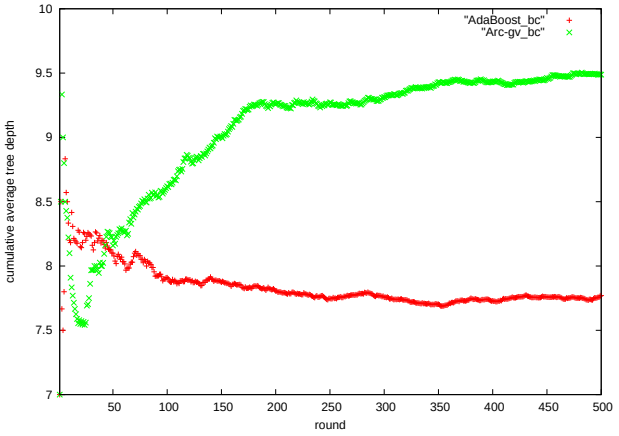

Figure 5. Cumulative average of decision tree depth for AdaBoost and arc-gv for the breast cancer set for 500 rounds of boosting.

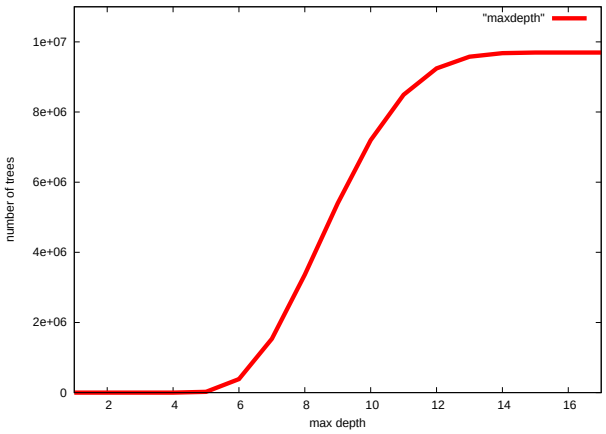

Figure 6. The number of tree topologies of depth at most d, as a function of d.

Table 4. Percent test and training errors per generated tree, and their differences, averaged over all CART decision trees generated in 500 rounds of boosting, over 10 trials.

	AdaBoost			arc-gv		
	test	train	**diff**	test	train	**diff**
cancer	13.2	9.7	**3.5**	10.4	6.3	**4.1**
ion	19.8	10.9	**8.9**	12.5	2.6	**9.9**
ocr 17	5.6	3.7	**1.9**	2.6	0.6	**2.0**
ocr 49	24.8	21.1	**3.7**	21.9	17.8	**4.1**
splice	27.7	23.4	**4.3**	23.9	19.2	**4.7**

higher error, it also produces, on average, deeper trees.

Referring back to the bound in Eq. (1), we can upper bound the VC-dimension d of a finite space of base classifiers $\mathcal{H}$ by $\lg|\mathcal{H}|$. Thus, measuring complexity is essentially a matter of counting how many trees there are of bounded depth. Clearly, the more tightly bounded is the depth, the more constrained is the space of allowable trees, and the smaller will be the complexity measure $\lg|\mathcal{H}|$. This can be seen in Figure 6 which shows the number of 16-leaf tree topologies of depth at most d, as a function of d.

So we are claiming that a possible explanation for the better performance of arc-gv despite its higher margins is that it achieves them by choosing from a greater set of base classifiers. By the bound in Eq. (1), we can see that the higher depths of arc-gv trees can be affecting the generalization error even if the margins explanation holds.

In general, we expect the difference between training and test errors to be greater when classifiers are selected from a larger or more complex class. Thus, as a further indication that the deeper trees generated by arc-gv are more likely to cause overfitting, we can directly measure the difference between the test error and (unweighted) training error for the trees generated by each algorithm. In Table 4, we can see that this difference is substantially higher for arc-gv than for AdaBoost in each of the datasets. This adds to the evidence that arc-gv is producing higher margins by using trees which are more complex in the sense that they have a greater tendency to overfit.[3]

The margins explanation basically says that when all other factors are equal, higher margins result in lower error. Given, however, that arc-gv tends to choose trees from a larger class, its higher test error no longer qualitatively contradicts the margin theory.

5. Controlling Classifier Complexity

Knowing that arc-gv should produce a higher minimum margin in the limit, and observing that with CART trees, arc-gv produces a uniformly higher distribution than AdaBoost, we wished to fix the complexity of the classifiers both algorithms produce. The

[3]While it is curious that the test errors of the individual trees generated by arc-gv are on average lower than those for AdaBoost, it does not necessarily follow that the *combination* of trees generated by arc-gv should perform better than that produced by AdaBoost. For example, as will be seen in Section 5, decision stumps can work quite well as base classifiers while individually having quite large test and training errors.

Table 5. Test errors, minimum margins, and average margins averaged over 100 trials, of AdaBoost and arc-gv, run for 100 rounds using decision stumps as weak learners.

	test error		minimum margin		average margin	
	arc-gv	AdaBoost	arc-gv	AdaBoost	arc-gv	AdaBoost
cancer	4.15	4.29	-.01	-.06	.07	.27
ionosphere	10.27	9.58	.01	.03	.09	.20
ocr 17	1.12	1.10	.03	.06	.14	.36
ocr 49	6.38	6.28	-.02	-.07	.05	.20
splice	7.22	6.79	-.01	-.07	.06	.21

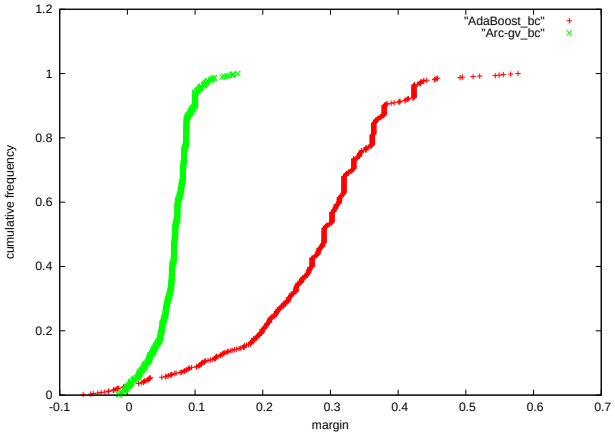

Figure 7. Cumulative margins for AdaBoost and arc-gv for the breast cancer dataset after 100 rounds of boosting on decision stumps.

margins theory tells us that if arc-gv still continued to produce higher margins, it should also perform better. However, if we could see that arc-gv, with some class of weak learners, gets higher margins without generating higher depth trees and still performs worse, it would put the margins theory into serious doubt.

A natural class to look at is decision stumps, which are commonly used as base classifiers in boosting and all have the same complexity by most any measure. Yet, looking at sample margins distributions that AdaBoost and arc-gv generate, in Figure 7, we can see that while arc-gv usually does have a larger minimum margin, it does not have a higher margins distribution overall. In fact, if we look at the average margins, AdaBoost's are uniformly higher, and once again AdaBoost on average performs better than arc-gv. These results are in Table 5.[4]

[4]If arc-gv and AdaBoost run for more rounds, their margins distributions begin to converge, as do their test errors. Hence, in the few data sets where AdaBoost has slightly higher minimum margin after 100 rounds, this difference disappears when boosting is run longer.

Table 6. Percent test and training errors per generated stump, and their differences, averaged over all decision stumps generated in 100 rounds of boosting, over 10 trials.

	AdaBoost			arc-gv		
	test	train	**diff**	test	train	**diff**
cancer	40.7	40.5	**0.2**	41.8	41.7	**0.1**
ion	42.4	41.4	**1.0**	42.7	41.8	**0.9**
ocr 17	34.1	33.9	**0.2**	34.5	34.2	**0.2**
ocr 49	42.4	42.0	**0.4**	43.3	42.9	**0.4**
splice	42.5	41.9	**0.6**	43.2	42.7	**0.5**

This result is both surprising and insightful. We would have expected arc-gv to have uniformly higher margins once more, but this time have lower test error. Yet, it seems that in the case where arc-gv could not produce more complex trees, it sacrificed on the margins distribution as a whole to have an optimal minimum margin in the limit. Knowing this, the margins theory would no longer predict arc-gv to perform better, and it does not.

This is because the margins bound, in Eq. (1), depends on setting θ to be as low as possible while keeping the *probability* of being less than θ low. So if the margins of AdaBoost overtake the margins of arc-gv at the lower cumulative frequencies, then the theory would predict AdaBoost to perform better. This is exactly what happens.

For comparison to Table 4, we give in Table 6 the differences between the test and training errors of individual decision stumps generated in 100 rounds of AdaBoost and arc-gv. Consistent with theory, the differences in these test and training errors for individual stumps are much smaller than they are for CART

trees, reflecting the lower complexity or tendency to overfit of stumps compared to trees. Since these differences are nearly identical for AdaBoost and arc-gv, this also suggests that the stumps generated by the two algorithms are roughly of the same complexity.

6. Discussion

In this paper, we have shown an alternative explanation for arc-gv's poorer performance that is consistent with the margins theory. We can see that while having higher margins is desirable, we must pay attention to other factors that can also influence the generalization error of the classifier.

Our experiments with decision stumps show us that it may be fruitful to consider boosting algorithms that greedily maximize the average or median margin rather than the minimum one. Such an algorithm may outperform both AdaBoost and arc-gv.

Finally, we leave open an interesting question. We have tried to keep complexity constant using base classifiers other than decision stumps, and in every instance we have seen AdaBoost generate higher average margins. Is there a base classifier that has constant complexity, with which arc-gv will have an overall higher margins distribution than AdaBoost? If such a base learner exists, it would be a good test of the margins explanation to see whether arc-gv would have lower error than AdaBoost as we predict. However, it is also possible that unless arc-gv "cheats" on complexity, it cannot generate overall higher margins than AdaBoost.

Acknowledgments

This material is based upon work supported by the National Science Foundation under grant numbers CCR-0325463 and IIS-0325500. We also thank Cynthia Rudin for helpful discussions.

References

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1987). Occam's razor. *Information Processing Letters*, *24*, 377–380.

Breiman, L. (1998). Arcing classifiers. *The Annals of Statistics*, *26*, 801–849.

Breiman, L. (1999). Prediction games and arcing classifiers. *Neural Computation*, *11*, 1493–1517.

Drucker, H., & Cortes, C. (1996). Boosting decision trees. *Advances in Neural Information Processing Systems 8* (pp. 479–485).

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

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

Koltchinskii, V., & Panchenko, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, *30*.

Mason, L., Bartlett, P. L., & Golea, M. (2002). Generalization error of combined classifiers. *Journal of Computer and System Sciences*, *65*, 415–438.

Meir, R., & Rätsch, G. (2003). An introduction to boosting and leveraging. In S. Mendelson and A. Smola (Eds.), *Advanced lectures on machine learning (lnai2600)*, 119–184. Springer.

Quinlan, J. R. (1996). Bagging, boosting, and C4.5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (pp. 725–730).

Rätsch, G., & Warmuth, M. (2002). Maximizing the margin with boosting. *15th Annual Conference on Computational Learning Theory* (pp. 334–350).

Rudin, C., Schapire, R. E., & Daubechies, I. (2004). Boosting based on a smooth margin. *17th Annual Conference on Learning Theory* (pp. 502–517).

Schapire, R. E. (2002). The boosting approach to machine learning: An overview. *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. *The Annals of Statistics*, *26*, 1651–1686.

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