The Annals of Statistics, 32(1), February, 2004.

A Discussion of:
"Process Consistency for AdaBoost" by Wenxin Jiang
"On the Bayes-risk consistency of regularized boosting methods" by Gábor Lugosi and Nicolas Vayatis
"Statistical Behavior and Consistency of Classification Methods based on Convex Risk Minimization" by Tong Zhang

Yoav Freund
Mitsubishi Electric Research Laboratories
201 Broadway
Cambridge, MA 02139
freund@merl.com

Robert E. Schapire
Department of Computer Science
Princeton University
35 Olden Street
Princeton, NJ 08544
schapire@cs.princeton.edu

The notion of a boosting algorithm was originally introduced by Valiant in the context of the "probably approximately correct" (PAC) model of learnability [19]. In this context boosting is a method for provably improving the accuracy of any "weak" classification learning algorithm. The first boosting algorithm was invented by Schapire [16] and the second one by Freund [2]. These two algorithms were introduced for a specific theoretical purpose. However, since the introduction of AdaBoost [5], quite a number of perspectives on boosting have emerged. For instance, AdaBoost can be understood as a method for maximizing the "margins" or "confidences" of the training examples [17]; as a technique for playing repeated matrix games [4, 6]; as a linear or convex programming method [15]; as a functional gradient-descent technique [8, 13, 14, 3]; as a technique for Bregman-distance optimization in a broader framework that includes logistic regression [1, 10, 12]; and finally as a stepwise model-fitting method for minimization of the exponential loss function, an approximation of the negative log binomial likelihood [7]. The current papers add to this list of perspectives, giving a view of boosting that is very different from its original interpretation and analysis as an algorithm for improving the accuracy of a weak learner. These many different points of view add to the richness of the theory of boosting, and are enormously helpful in the practical design of new or better algorithms for machine learning and statistical inference.

Originally, boosting algorithms were designed expressly for classification. The goal in this setting is to accurately predict the classification of a new example. Either the prediction is correct, or it is not. There is no attempt made to estimate the conditional probability of each class. In practice, this sometimes is not enough since we may want to have some sense of how likely our prediction is to be correct, or we may want to incorporate numbers that look like probabilities into a larger system.

Later, Friedman, Hastie and Tibshirani [7] showed that AdaBoost can in fact be used to estimate such probabilities, arguing that AdaBoost approximates a form of logistic regression. They and others [1] subsequently modified AdaBoost to explicitly minimize the loss function associated with logistic regression, with the intention of computing such estimated probabilities. In one of the current papers, Zhang vastly generalizes this approach showing that conditional probability estimates $P\{y|x\}$ can be obtained when minimizing any smooth convex loss function, not just exponential loss or negative log binomial likelihood. Moreover, he relates the loss to a specific Bregman distance between the true conditional probability and its estimate. This fascinating result leads one to wonder how special the exalted log likelihood loss function really is for this task when apparently any convex function will do.

It seems that most if not all of the consistency results in these papers depend on the ability of boosting-like methods to estimate probabilities. That is, this work tends to divide the inference process into two steps: (1) estimate the conditional probability of y given x, and (2) use this estimate to make a prediction, for example, select the class with highest estimated conditional probability. Although, as noted above, this can be very useful in some applications, in other cases, we really are only interested in being able to make accurate predictions with no opportunity to hedge with a probability estimate. In this case, there is no need to estimate conditional probabilities. Such estimates are in no way necessary for classification. For instance, such estimates are not used when analyzing boosting in terms of the margins of the training examples [11, 17], nor in the theory of support-vector machines [20]. It is perhaps inevitable in the quest for consistent learning algorithms that we end up thinking about conditional probability estimates. But if the goal is classification accuracy, then we may be seeking something that is more than we really need. This is Vapnik's basic message: don't try to estimate probabilities (or conditional probabilities) if your goal is classification; simply try to minimize the empirical error and use uniform convergence bounds to estimate the out-of-sample performance.

These three papers also all seem to require an assumption of the denseness of the estimating class. Again, if the goal is consistency, then such an assumption seems unavoidable. Unfortunately, this can be a rather strong assumption. For instance, using decision stumps apparently does not satisfy the denseness requirement. Decision trees probably do satisfy this requirement, but there is no efficient method for provably finding the best decision tree on a given dataset. Denseness means that the approximating class must be very rich, rich enough to approximate nearly any function. Lacking additional assumptions it seems that this precludes the possibility of inferring the label of any instance that is not in the training set. Thus, the need for regularization. This unfortunately adds a degree of complexity to the practical application of these algorithms. Moreover, AdaBoost usually seems to work fine without regularization, bringing into question its necessity (though raising the possibility of it benefiting from its use).

In most applications, we know full well that the true distribution is far from any distribution in our class. For example, nobody using HMM's for speech analysis really thinks that speech can be synthesized by these HMM's. Are there other modes of analysis that do not require such strong assumptions? Given a "reasonable" class, but one that does not admit zero approximation error, what can be said about how well these algorithms perform?

Although interesting and important, the analyses given in these papers do not seem to offer insight as to why boosting and support-vector machines are effective in higher dimensions, a phenomenon that is perhaps better captured by the respective margins theories. Consistency does not seem to be related to the effectiveness of an algorithm in high dimensions. For instance, k-nearest neighbor algorithms are known to be consistent, but are also known to suffer considerably from the curse of dimensionality [9].

Both Zhang and Lugosi and Vayatis carry out their analysis only with regard to the loss function that they are studying. In other words, they do not consider at all the algorithm that is used to minimize that loss function. However, in studying a learning algorithm like AdaBoost, the loss function alone cannot tell us the whole story. For instance, suppose the data is linearly separable so that there exists a set of weights $w_1, \ldots, w_N$ and a set of base classifiers $g_1, \ldots, g_N$ such that, for each training example (x_i, y_i),

$$y_i \sum_j w_j g_j(x_i) > 0,$$

i.e., y_i is equal to the sign of $f(x_i) = \sum w_j g_j(x_i)$. AdaBoost attempts to minimize the exponential loss

$$\sum_i \exp(-y_i f(x_i)).$$

Clearly, if we multiply each weight w_j by a large positive constant c, then this loss will quickly be driven to zero. Thus, the fact that AdaBoost minimizes the exponential loss only tells us that it finds a separating hyperplane (with which it can drive the exponential loss to zero). It does not tell us anything about *which* hyperplane was selected, and it is well known that we can expect some hyperplanes to be much better than others (witness the success of support-vector machines). So it is not enough to look only at the loss function — we also need to consider the mechanics of the specific algorithm that is being used.

Exponential loss is in terms of the *unnormalized* margin $yf(x)$, whereas the margins theory [17] is about the *normalized* margin (in which we divide f by the sum of the weights of the base classifiers). In the example of linearly separable data above, minimizing exponential loss implies maximizing the unnormalized margins by forcing all of them to approach (positive) infinity. As noted above, this tells us nothing about which separating hyperplane was selected. On the other hand, AdaBoost is known to approximately maximize the *normalized* margins, a property that does very strongly constrain the separating hyperplane that is selected, and that, it can be argued, goes far in explaining why boosting is more effective than choosing just any old hyperplane.

The comments in Section 6 of Lugosi and Vayatis are quite amusing. It has previously been observed that intuitively AdaBoost and other boosting algorithms attempt to force the weak classifiers to behave as if they were independent. Indeed, Lugosi and Vayatis's comments can be generalized to the case where the weak classifiers are not independent: In this case, if the t-th weak classifier h_t has error p on the distribution D_t on which it was trained (which will automatically be true if they are independent as in the Lugosi and Vayatis paper) then the error $L(f)$ of the resulting combined classifier will again be

$$\left(2\sqrt{p(1-p)}\right)^N.$$

In fact, there is another boosting algorithm, called the boost-by-majority algorithm [2], that gives a bound on the error that is *not* a Chernoff bound, but is instead an exact binomial tail:

$$\sum_{i=0}^{N/2} \binom{N}{i} p^{N-i}(1-p)^i.$$

Understanding the properties of this algorithm in the frameworks employed in these papers would certainly be an interesting challenge.

More broadly, all this points to a strong connection between probability theory and game theory. This is spelled out beautifully by Schafer and Vovk [18].

References

[1] Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1/2/3), 2002.

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

[3] Yoav Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3):293–318, June 2001.

[4] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 325–332, 1996.

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

[6] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[7] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, April 2000.

[8] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), October 2001.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Verlag, 2001.

[10] Jyrki Kivinen and Manfred K. Warmuth. Boosting as entropy projection. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 134–144, 1999.

[11] V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1), February 2002.

[12] Guy Lebanon and John Lafferty. Boosting and maximum likelihood for exponential models. In *Advances in Neural Information Processing Systems 14*, 2002.

[13] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[14] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems 12*, 2000.

[15] Gunnar Rätsch, Manfred Warmuth, Sebastian Mika, Takashi Onoda, Steven Lemm, and Klaus-Robert Müller. Barrier boosting. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 170–179, 2000.

[16] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[17] 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, October 1998.

[18] Glenn Shafer and Vladimir Vovk. *Probability and Finance, it's only a game!* Wiley, 2001.

[19] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

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