[4] BREIMAN, L. (2000). Some infinity theory for predictor ensembles. Technical Report 577, Dept. Statistics, Univ. California, Berkeley.
[5] BÜHLMANN, P. and YU, B. (2003). Boosting with L_2-loss: Regression and classification. *J. Amer. Statist. Assoc.* **98** 324–339.
[6] FREUND, Y. and SCHAPIRE, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* **55** 119–139.
[7] FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion). *Ann. Statist.* **28** 337–407.
[8] LESHNO, M., LIN, YA. V., PINKUS, A. and SCHOCKEN, S. (1993). Multilayer feedforward networks with a non-polynomial activation function can approximate any function. *Neural Networks* **6** 861–867.
[9] LUGOSI, G. and VAYATIS, N. (2004). On the Bayes-risk consistency of regularized boosting methods. *Ann. Statist.* **32** 30–55.
[10] MANNOR, S., MEIR, R. and ZHANG, T. (2002). The consistency of greedy algorithms for classification. In *Proc. 15th Annual Conference on Computational Learning Theory. Lecture Notes in Comput. Sci.* **2375** 319–333. Springer, New York.
[11] ROCKAFELLAR, R. T. (1970). *Convex Analysis*. Princeton Univ. Press.
[12] RUDIN, W. (1987). *Real and Complex Analysis*, 3rd ed. McGraw-Hill, New York.
[13] SCHAPIRE, R. E., FREUND, Y., BARTLETT, P. and LEE, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Statist.* **26** 1651–1686.
[14] SCHAPIRE, R. E. and SINGER, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning* **37** 297–336.
[15] STEINWART, I. (2002). Support vector machines are universally consistent. *J. Complexity* **18** 768–791.
[16] VAPNIK, V. N. (1998). *Statistical Learning Theory*. Wiley, New York.
[17] WAHBA, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelpfia.
[18] ZHANG, T. (2001). A leave-one-out cross validation bound for kernel methods with applications in learning. In *Proc. 14th Annual Conference on Computational Learning Theory* 427–443. Springer, New York.

IBM T. J. WATSON RESEARCH CENTER
P.O. BOX 218
YORKTOWN HEIGHTS, NEW YORK 10598
USA
E-MAIL: tzhang@watson.ibm.com

DISCUSSION

BY PETER L. BARTLETT, MICHAEL I. JORDAN AND JON D. MCAULIFFE

University of California, Berkeley

The authors have contributed three significant papers that provide, among other insights, an understanding of the consistency of several "large margin" methods for pattern classification. In two-class classification, the aim is to find a function $f : \mathcal{X} \to \mathbb{R}$ that accurately predicts a binary response variable $Y \in \{\pm 1\}$ using the covariate $X \in \mathcal{X}$, in the sense that $R(f) = \mathbf{E}\ell(Yf(X))$, the risk of the

thresholded function, is minimized. Here, $\ell(z)$ denotes the indicator function of the event $z \leq 0$. *Large margin classification methods* use some loss function $\phi : \mathbb{R} \to \mathbb{R}$, typically convex, and seek a function f from some class $\mathcal{F}$ that minimizes the ϕ-risk, $R_\phi(f) = \mathbf{E}\phi(Yf(X))$, that is, the expected loss evaluated at the margin $Yf(X)$. These methods typically minimize the empirical ϕ-risk, $\hat{R}_\phi(f)$, or a regularized version thereof. Many successful pattern classification methods fall in this class, including AdaBoost and other greedy algorithms for forming ensembles of classifiers, and support vector machines. We can categorize them according to the loss function ϕ, the class of functions $\mathcal{F}$ and the algorithm used to approximately minimize R_ϕ.

The three papers in this issue demonstrate the consistency of various methods of this kind.

- The consistency result in the paper by Zhang applies to several loss functions and concerns kernel methods, which choose a function f from a reproducing kernel Hilbert space $\mathcal{H}$ of functions on $\mathcal{X}$ to minimize a regularized empirical ϕ-risk,

  $$\hat{R}_\phi(f) + C\|f\|_{\mathcal{H}},$$

  where $\|\cdot\|_{\mathcal{H}}$ is the Hilbert space norm. This is equivalent (for some λ) to choosing f from the function class

  $$\mathcal{F}_k(\mathcal{H}, \lambda) = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq \lambda\}$$

  so as to minimize $\hat{R}_\phi(f)$.
- The paper by Lugosi and Vayatis considers the loss function $\phi(\alpha) = \exp(-\alpha)$, the function class

  $$\mathcal{F}_b(\mathcal{G}, \lambda) = \left\{\sum_i \alpha_i g_i : \|\alpha\|_1 \leq \lambda, g_i \in \mathcal{G}\right\},$$

  where $\mathcal{G} \in \{\pm 1\}^{\mathcal{X}}$ has finite VC-dimension and an algorithm that minimizes empirical ϕ-risk. The AdaBoost algorithm is similar, but without the constraint on the coefficients.
- The paper by Jiang also considers the exponential loss function, the function class

  $$\mathcal{F}_b(k) = \left\{\sum_{i=1}^{k} \alpha_i g_i : g_i \in \mathcal{G}\right\},$$

  and the AdaBoost algorithm, which chooses the α_i, g_i sequentially, to greedily minimize empirical ϕ-risk.

We can identify three key steps in proving consistency results of this kind. The first involves a "comparison theorem," relating the excess risk $R(f) - R^*$ to the excess ϕ-risk, $R_\phi(f) - R_\phi^*$. Here, R^* is the Bayes risk, that is, the infimum over

all measurable f of $R(f)$, and $R_\phi^* = \inf_f R_\phi(f)$ is the analogous quantity for the ϕ-risk. A result of this kind is present in all three papers: Zhang's Theorem 2.1 gives an explicit inequality relating the two excess risks; Lugosi and Vayatis' Lemma 5 gives a limiting result; and Jiang's Lemma 1 gives a related comparison, via the $L_2(P)$ distance between f and $f_\phi^* = \arg\min_f R_\phi(f)$.

The second and third steps are more conventional in consistency proofs. The second step is to show that the functions used by the method are rich enough to approximate f_ϕ^*, the measurable function that minimizes ϕ-risk. As formulated above, this involves showing that

$$\bigcup_{\lambda>0} \mathcal{F}_k(\mathcal{H}, \lambda), \qquad \bigcup_{\lambda>0} \mathcal{F}_b(\mathcal{G}, \lambda) \quad \text{and} \quad \bigcup_{k>0} \mathcal{F}_b(k)$$

are sufficiently rich.

The third step is to choose a sequence of subsets $\mathcal{F}_n \subseteq \mathcal{F}$ with suitably restricted complexity as a function of the sample size n, so that the ϕ-risk of the estimated $\hat{f}_n \in \mathcal{F}_n$ converges to the minimal value, $\inf_{f\in\mathcal{F}_n} R_\phi(f)$. For example, in the cases considered in these three papers, the set $\mathcal{F}_n$ is defined as the set of combinations of k_n functions from $\mathcal{G}$, or the set of combinations of functions from $\mathcal{G}$ with the coefficient vector having one-norm no more than λ_n, or a ball of radius λ_n in an RKHS $\mathcal{H}$. (In the last case, $\hat{f}_n$ is chosen to minimize a combination of the empirical ϕ-risk and a regularization term involving the RKHS norm.)

This third step is a little more involved in the case of Jiang's consistency result, since that result involves an algorithm that does not minimize an objective function involving the empirical risk. Thus, it is essential to show that, under certain conditions, the algorithm finds a good function quickly.

It is interesting to consider what properties of the loss function ϕ allow comparison theorems, and hence consistency results, for large margin methods in general. Jiang's result is for the exponential loss function, and the proof exploits a smoothness assumption on the joint probability distribution that ensures that the optimal f_ϕ^* is continuous. Lugosi and Vayatis assume that ϕ is differentiable, strictly convex, monotonic, and has a certain limiting behavior. Zhang assumes that ϕ satisfies three conditions:

1. For any $\eta \neq 1/2$, any minimizer α^* of the conditional ϕ-risk, $\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)$ has the same sign as $\eta - 1/2$. Thus, a pointwise minimization of the conditional ϕ-risk leads to a function that gives the correct sign everywhere.
2. ϕ is convex.
3. The minimal conditional ϕ-risk,

$$H(\eta) = \inf_{\alpha\in\mathbb{R}} \big(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\big),$$

decreases polynomially with $|1/2 - \eta|$.

Note that the first condition is implied by Lugosi and Vayatis' assumptions (as can be verified by a short calculation), and thus holds a fortiori for the exponential function studied by Jiang. The condition is clearly the weakest possible condition that can be imposed on ϕ if we are to obtain consistency—if the minimizer of ϕ-risk yields the wrong sign at a given point, then it is easy to concoct a probability distribution that has zero excess ϕ-risk but nonzero excess risk. Surprisingly, it turns out that this condition is not only necessary but is also sufficient for obtaining a general comparison theorem—no other conditions are needed. We provide a brief overview of this result here; see [1] for a detailed presentation.

We begin by defining the following functional transform of a loss function ϕ:

DEFINITION 1. Given $\phi : \mathbb{R} \to [0, \infty)$, define the function $\tilde{\psi} : [0, 1] \to [0, \infty)$ by

$$\tilde{\psi}(\theta) = H^-\left(\frac{1+\theta}{2}\right) - H\left(\frac{1+\theta}{2}\right),$$

where

$$H(\eta) = \inf_{\alpha \in \mathbb{R}} \big(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\big),$$

$$H^-(\eta) = \inf_{\alpha\,:\,\alpha(2\eta-1)\le 0} \big(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\big).$$

The *ψ-transform* is defined to be the function $\psi : [0, 1] \to [0, \infty)$ that is the convex closure of $\tilde{\psi}$.

Note that it is straightforward to compute the ψ-transform for all of the examples of loss functions ϕ studied in the three papers in this issue.

The importance of the ψ-transform is shown by the following theorem.

THEOREM 2. *For any nonnegative loss function ϕ, any measurable $f : \mathcal{X} \to \mathbb{R}$ and any probability distribution on $\mathcal{X} \times \{\pm 1\}$,*

$$\psi\big(R(f) - R^*\big) \le R_\phi(f) - R_\phi^*.$$

This theorem establishes a general quantitative relationship between the excess ϕ-risk and the excess risk.

For this relationship to be useful in particular applications we need to show that ψ has particular properties—properties that arise from conditions that are imposed on ϕ. In particular, let us introduce the condition described above—that pointwise minimization of the conditional ϕ-risk leads to a function that gives the correct sign. We express this condition in the following way:

DEFINITION 3. We say that ϕ is *classification-calibrated* if, for any $\eta \ne 1/2$,

$$H^-(\eta) > H(\eta).$$

Equivalently, ϕ is classification-calibrated if for any sequence (α_i) such that

$$\lim_{i\to\infty} \{\eta\phi(\alpha_i) + (1-\eta)\phi(-\alpha_i)\} = H(\eta),$$

we have

$$\lim_{i\to\infty} \operatorname{sign}\big(\alpha_i(\eta - 1/2)\big) = 1.$$

In particular, if the infimum $H(\eta)$ is achieved at a minimizing value α^*, then this value must have the correct sign. Thus, this condition is essentially an elaboration of Zhang's first condition. As pointed out by Lin [2], it can be viewed as a variant of Fisher consistency that is appropriate for classification.

We have the following result:

THEOREM 4. *The following conditions are equivalent*:

1. ϕ *is classification-calibrated*.
2. *For any sequence* (θ_i) *in* $[0, 1]$,

$$\psi(\theta_i) \to 0 \qquad \text{if and only if } \theta_i \to 0.$$

3. *For every sequence of measurable functions* $f_i : \mathcal{X} \to \mathbb{R}$ *and every probability distribution* P,

$$R_\phi(f_i) \to R_\phi^* \qquad \text{implies } R(f_i) \to R^*.$$

Thus we see that we obtain a meaningful general comparison theorem under the weakest possible condition on the loss function ϕ. In addition, it can be shown that for a given ϕ, the ψ-transform is optimal in the sense that everywhere on its domain, the bound given by Theorem 2 cannot be improved in general.

Note in particular that we have not assumed that ϕ is convex. If we do assume that ϕ is convex, then we can say more—in particular, the function $\tilde{\psi}$ in Definition 1 is then necessarily closed and convex, and thus the ψ-transform is specified directly via the variational representation $\psi(\theta) = H^-((1+\theta)/2) - H((1+\theta)/2)$. Moreover, if ϕ is convex, then it is possible to show that it is classification-calibrated if and only if it is differentiable at 0 and $\phi'(0) < 0$.

The comparison theorem in Theorem 2 and the analogous comparison theorems in the three papers in this issue suggest a general framework for studying pattern classification methods that involve a surrogate loss function. It is common to view the excess risk as a combination of an estimation term and an approximation term:

$$R(f) - R^* = \left(R(f) - \inf_{g\in\mathcal{F}} R(g)\right) + \left(\inf_{g\in\mathcal{F}} R(g) - R^*\right).$$

However, choosing a function with risk near minimal over a class $\mathcal{F}$—that is, finding an f for which the estimation term above is close to zero—is, in a minimax setting, equivalent to the problem of minimizing empirical risk. For

typical classes $\mathcal{F}$ of interest, this problem is computationally infeasible. Even worse, for the function classes typically used by boosting and kernel methods, the estimation term in this expression does not converge to zero for the minimizer of the empirical risk. On the other hand, the comparison theorems we are considering suggest splitting the upper bound on excess risk into an estimation term and an approximation term:

$$\begin{aligned}\psi\big(R(f) - R^*\big) &\le R_\phi(f) - R_\phi^* \\ &= \Big(R_\phi(f) - \inf_{g\in\mathcal{F}} R_\phi(g)\Big) + \Big(\inf_{g\in\mathcal{F}} R_\phi(g) - R_\phi^*\Big).\end{aligned} \tag{1}$$

We can view the function ψ provided by the comparison theorem as quantifying the penalty incurred by using the surrogate loss function ϕ in place of the 0–1 loss, and linking the excess risk to the approximation error and estimation error associated with the ϕ-risk.

In many cases it is possible to minimize the ϕ-risk efficiently over a convex class $\mathcal{F}$ and, hence, find an $f \in \mathcal{F}$ for which this upper bound on risk is near minimal. This holds despite the fact that finding an $f \in \mathcal{F}$ with near-minimal risk is typically computationally infeasible.

Another interesting question raised by Theorem 2 and by the papers in this issue is that of convergence rates. Zhang's paper makes a start in this direction for kernel methods, and this is continued in his more recent work with Mannor and Meir concerning boosting methods [3]. Tsybakov [4] has considered empirical risk minimization in pattern classification problems with low noise—specifically, where the P_X-probability that $P(Y = 1|X)$ is near $1/2$ is small. He showed that the risk of the empirical minimizer converges to its minimal value surprisingly quickly in these cases. It turns out that, under Tsybakov's low noise condition, the relationship between excess risk and excess ϕ-risk presented in Theorem 2 can be improved [1]. In that case, if the loss function ϕ is uniformly convex and $\mathcal{F}$ is convex, then the excess risk converges to its minimal value [the approximation error term in (1)] surprisingly quickly.

The problem of classification has been a fruitful domain in which to explore connections between statistical and computational science. Efficient algorithms can be designed to solve large-scale classification problems by exploiting tools from convex optimization, and the statistical consequences of using these tools are beginning to be understood. The three papers in this issue represent significant progress on the general problem of incorporating considerations of computational complexity in statistical theory, providing hints of general tradeoffs between statistical accuracy and computational resources that are only beginning to be explored.

REFERENCES

[1] BARTLETT, P. L., JORDAN, M. I. and MCAULIFFE, J. D. (2003). Convexity, classification, and risk bounds. Technical Report 638, Dept. Statistics, Univ. California, Berkeley.

[2] LIN, Y. (2001). A note on margin-based loss functions in classification. Technical Report 1044r, Dept. Statistics, Univ. Wisconsin.

[3] MANNOR, S., MEIR, R. and ZHANG, T. (2002). The consistency of greedy algorithms for classification. In *Proc. 15th Annual Conference on Computational Learning Theory. Lecture Notes in Comput. Sci.* **2375** 319–333. Springer, New York.

[4] TSYBAKOV, A. (2004). Optimal aggregation of classifiers in statistical learning. *Ann. Statist.* **32** 135–166.

P. L. BARTLETT
M. I. JORDAN
DIVISION OF COMPUTER SCIENCE
AND DEPARTMENT OF STATISTICS
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720-3860
USA
E-MAIL: bartlett@stat.berkeley.edu
jordan@stat.berkeley.edu

J. D. MCAULIFFE
DEPARTMENT OF STATISTICS
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720-3860
USA
E-MAIL: jon@stat.berkeley.edu

THE GOLDEN CHAIN

BY PETER J. BICKEL AND YA'ACOV RITOV

University of California, Berkeley and Hebrew University of Jerusalem

And through the palpable obscure find out / His uncouth way.

J. Milton, *Paradise Lost*.

Jiang, Lugosi and Vayatis, and Zhang, in part explicitly and in part implicitly, have done a great deal in explaining the nature of boosting from a statistical point of view.

The problem all consider is that of finding classifiers that approximate the Bayes classifier using only a training sample (X_i, Y_i), $i = 1, \ldots, n$, $(X_i, Y_i) \sim (X, Y)$, with $Y = \pm 1$ (for simplicity). The Bayes classifier is described as $\operatorname{sgn}(F_p(X))$, where $F_p(X) = q \circ \log(p[Y = 1|X]/P[Y = -1|X])$, for any strictly increasing function q with $q(0) = 0$.

The methods of approximation discussed by these and previous authors cited in their papers have the common setting that the approximating values are $\operatorname{sgn}(\hat{F}(X))$, where $\hat{F} \in \tilde{\mathcal{F}} \equiv \bigcup_{k=1}^{\infty} \mathcal{F}_k$, $\mathcal{F}_k = \{\sum_{j=1}^{k} \lambda_j h_j : h_1, \ldots, h_k \in \mathcal{H}, \lambda_1, \ldots, \lambda_k \in R\}$ and $\mathcal{H}$ is a set of base classifiers, $h : \mathcal{X} \to \{-1, 1\}$.

All methods are based on the following two observations:

(i) Given W convex, $W = \mathbb{R} \to \mathbb{R}^+$. Then, at least formally, if $\tilde{\mathcal{F}}$ is rich enough and P denotes expectation, then $F_p = \arg\min P\, W(Y F(X))$ as above. The

validity of this identity is studied extensively by Zhang who relates it to minimizing the Bregman divergence between F and F_p. The function $W(t) = e^t$ corresponds to classical AdaBoost, while $W(t) = -2t + t^2$ is "L_2 boosting." See [5] and [9].

(ii) One "optimizes" $P_n W(YF(X))$ over $\tilde{\mathcal{F}}$, where P_n is the empirical distribution of (X_i, Y_i), $i = 1, \ldots, n$, in the same way to obtain $\hat{F}$. The classical prescription of Breiman [3] is to optimize greedily starting at $F_0 \equiv 1$ using the Gauss–Southwell approach moving from $\mathcal{F}_m$ to $\mathcal{F}_{m+1}$ on the mth step.

Unfortunately, as is made fairly explicit in these papers, unless P is discrete, $\inf_{F \in \tilde{\mathcal{F}}} P_n W(YF(X)) = 0$, and optimizing to the bitter end leads to overfitting.

Jiang shows for classical AdaBoost that, under some conditions, given convergence of the population algorithm, it is possible to stop the sample algorithm early and achieve consistency, that is, convergence to the Bayes classifier. Lugosi and Vayatis and Zhang separately show that by regularizing, effectively changing what is being optimized, convergence to the Bayes classifier is possible quite generally and obtain rates for their procedures. Such approaches via sieves have already been considered by Baraud [1] for "L_2 boosting."

We see four distinct questions:

(i) When are greedy algorithms consistent in the population case?

(ii) When does early stopping in the sample case lead to a consistent procedure?

(iii) How can early stopping be implemented by cross-validation?

(iv) How can one directly modify the greedy algorithm, retaining its simple sequential structure and yet achieve optimal rate upon stopping suitably?

In our remark we address points (i) and (ii). Point (ii) is treated separately by Bickel and Ritov [2], Zhang and Yu [8] and Bühlmann [4] and (iv) is in progress.

1. Weak consistency. Here is a very general framework.

Let $\Theta_1 \subset \Theta_2 \subset \cdots$ be a sequence of sets contained in a separable metric space with metric ρ, $\Theta = \overline{\bigcup \Theta_m}$, where $\bar{}$ denotes closure. Let K be a target function and $\vartheta_\infty = \arg\min_\Theta K(\vartheta)$. Let $\Pi_m : \Theta_{m+1} \to \Theta_m$. Finally, let K_n be a sample-based approximation of K. We assume the following:

A1. For any m, $\vartheta_0 : M$, $\Theta_m \cap \{\vartheta : \rho(\vartheta, \vartheta_0) < M\}$ is compact. Let $K : \Theta \to \mathbb{R}$ and assume that $\vartheta_\infty = \arg\min_{\vartheta \in \Theta} K(\vartheta)$ is unique.

A2. K is strictly convex and $K(\vartheta) \le K(\vartheta') \Rightarrow \rho(\vartheta, \vartheta_\infty) \le A\rho(\vartheta', \vartheta_\infty)$ for some $A < \infty$.

A3. If $\rho(\vartheta_m, \vartheta_0) \to \infty$ for some, and hence all ϑ_0, then $K(\vartheta_m) \to \infty$.

Let $\Pi_m : \Theta_m \to 2^{\Theta_{m+1}}$ be a sequence of point to set ρ-continuous mappings, where distance between sets is defined as $\rho(A, B)$, the Hausdorff distance between

the closures of A and B, and define the following algorithm generating a sequence $\bar{\vartheta}_m \in \Theta_m$, $m = 1, 2, \ldots$, given an initial point ϑ_0:

(i) $\bar{\vartheta}_{m+1} \in \Pi_m(\bar{\vartheta}_m)$.
(ii) $K(\bar{\vartheta}_{m+1}) = \inf_{\vartheta \in \Pi_m(\bar{\vartheta}_m)} K(\vartheta)$.

Suppose:

A4. If $\{\vartheta_m\}$ is defined as above with any initial ϑ_0, then $\rho(\vartheta_m, \vartheta_\infty) \to 0$.

In boosting, given P, $\Theta = \{F(X), F \in \tilde{\mathcal{F}}\}$, ρ is a metric of convergence in probability, $\Theta_m = \{\sum_{j=1}^m \lambda_j h_j, h_j \in \mathcal{H}\}$ and $\Pi_m(F) = \{F + \lambda h, \lambda \in R, h \in \mathcal{H}\}$. Moreover, $K(F) = EW(YF(X))$.

Now suppose $K_n(\cdot)$ is a sequence of random functions on Θ such that:

A5. K_n is convex and $\sup\{|K_n(\vartheta) - K(\vartheta)| : \vartheta \in \Theta_m,\ \rho(\vartheta, \vartheta_m) < M\} \xrightarrow{P} 0$ for all finite m, M, ϑ_0.

In boosting, $K_n(F) = n^{-1} \sum_{i=1}^n W(Y_i F(X_i))$ and A5 corresponds to requiring that $\{W(YF(X)) : F \in \Theta_m,\ \rho(F, F_0) \le M\}$ is uniformity class for LLN for P, for instance, a VC class. Bühlmann [4], Zhang and Yu [8] and Bickel and Ritov [2] discuss such conditions in different degrees of generality.

The sequence $\{\bar{\vartheta}_m\}$ is the golden chain we try to follow using the obscure information in the sample. Define $\hat{\vartheta}_{m,n}$ by the following:

(i) $\hat{\vartheta}_{m+1,n} \in \Pi_m(\hat{\vartheta}_{m,n})$.
(ii) If $\vartheta' \in \Pi_m(\hat{\vartheta}_{m,n})$, then $K_n(\hat{\vartheta}_{m+1,n}) \le K_n(\vartheta')$ and, in case of equality, also $\rho^*(\vartheta_{m+1,n}, \vartheta_0) \le \rho^*(\vartheta', \vartheta_0)$ for some metric ρ^* such that $\rho(\vartheta_m, \vartheta_0) \to \infty \Rightarrow \rho^*(\vartheta_m, \vartheta_0) \to \infty$.

The purpose of introducing ρ^* is to avoid an unnecessarily large norm of the estimate. In boosting ρ^* can be any metric like the $L_2(\mu)$ metric where μ has fatter tails than P.

THEOREM 1.1. *Under* A1–A5 *there exists a sequence* $\{m_n\}$ *such that*

$$\rho(\vartheta_{m_m,n}, \vartheta_\infty) \xrightarrow{P} 0.$$

PROOF. Consider $\hat{\vartheta}_{1,n}$. By definition,

$$K_n(\hat{\vartheta}_{1,n}) \le \min\{K_n(\vartheta_0), K_n(\bar{\vartheta}_1)\}. \tag{1}$$

However, for large enough M, we get from A3 that $\inf_{\vartheta \in \Theta_1, \rho(\vartheta, \vartheta_0) = M} K(\vartheta) > K(\vartheta_0)$. By A5 we obtain that also

$$P\left(\inf_{\vartheta \in \Theta_1, \rho(\vartheta, \vartheta_0) = M} K_n(\vartheta) > K(\vartheta_0)\right) \to 1. \tag{2}$$

Convexity of K_n, (1) and (2) imply that $\rho(\hat{\vartheta}_{1,n})$ is bounded. But then strict convexity of K and uniform convergence imply that

$$\rho(\hat{\vartheta}_{1,n}, \bar{\vartheta}_1) \xrightarrow{P} 0. \tag{3}$$

We continue now to $\hat{\vartheta}_{2,n}$. Since K is continuous, (3) implies that $\inf_{\vartheta \in \Pi_1(\hat{\vartheta}_1)} K(\vartheta) \xrightarrow{P} K(\bar{\vartheta}_2)$. Applying the same argument as for $\hat{\vartheta}_{1,n}$, we get $\rho(\hat{\vartheta}_{2,n}, \bar{\vartheta}_2)$ is bounded, and since K is continuous and strictly convex, we get again that $\rho(\hat{\vartheta}_{2,n}, \hat{\vartheta}_2) \xrightarrow{P} 0$. By induction, we obtain that $\rho(\hat{\vartheta}_{m,n}, \bar{\vartheta}_m) \xrightarrow{P} 0$ for every m.

Let $m_n = \sup\{m : P(\rho(\hat{\vartheta}_{m,n}, \bar{\vartheta}_m) < m^{-1}) < m^{-1}\}$. Then $m_n \to \infty$ and $\rho(\hat{\vartheta}_{m_n,n}, \bar{\vartheta}_{m_n}) \xrightarrow{P} 0$. Apply A4 to conclude the proof. □

Results based on this theorem cannot give an estimate of the speed of convergence of $\hat{\vartheta}_{m_n,n}$ to ϑ_∞, since the $\{m_n\}$ are not known. As we have mentioned, regularization can yield such rates but in all cases we are left with a sequence $\{\hat{\vartheta}_{1,n}, \hat{\vartheta}_{2,n}, \dots\}$ of procedures for which we need to select a stopping time τ on the basis of the data such that $\hat{\vartheta}_{\tau,n}$ behaves well. A natural comparison is to the oracle stopping time W, such that $EK(\hat{\vartheta}_{W,n}) = \min_m EK(\hat{\vartheta}_{m,n})$. In the next section we give a general result guaranteeing that $K(\hat{\vartheta}_{\tau,n}) \approx EK(\hat{\vartheta}_{W,n})$ in the context of classification. We shall show how this result may be applied to the regularized variants of boosting elsewhere.

2. The beauty of the test-bed. The boosting algorithm can be stopped appropriately if there are available good data driven bounds on the sample error. However, it is more practical to use some type of cross-validation. Here is a general result.

Assume that the observations are i.i.d. from $Z = (Y, X_1, X_2, \dots) = (Y, \mathbf{X})$, where $Y \in \{-1, 1\}$. The task is to find a function $\vartheta(\mathbf{X})$, such that $P(Y\vartheta(\mathbf{X}) > 0)$ is maximized. The sample is divided into a main sample, $Z_1, \dots, Z_n$, and a test-bed $Z_1^T, \dots, Z_k^T$. The main sample is used to derive a sequence of classifiers $\hat{\vartheta}_1, \hat{\vartheta}_2, \dots$. The test data is used to pick $\hat{\vartheta}_\tau$ as the classifier to be used, where

$$\tau = \arg\min_{m<M} \sum_{j=1}^{k} \mathbf{1}\big(Y_j^T \hat{\vartheta}_m(\mathbf{X}_j^T) > 0\big).$$

An oracle constrained to use rules of the form $\operatorname{sgn}(Y^T \hat{\vartheta}_m(\mathbf{X}^T))$ would use

$$W = \arg\min_{m<M} P\big(Y^T \hat{\vartheta}_m(\mathbf{X}) > 0 | \hat{\vartheta}_i\big).$$

Let $\eta_m = P(Y^T \hat{\vartheta}_m(\mathbf{X}) > 0 | \hat{\vartheta}_m(\cdot))$, $m = 1, 2, \dots$. The following assumption will be used:

ASSUMPTION S (Similarity of the good classifiers). With probability converging to 1, one of the following holds for every $m < K$:

1. $(\log M)^{-1}\sqrt{k}(\eta_m - \eta_W) > b_n$, for some $b_n \to \infty$.
2. $P(\hat{\vartheta}_m(\mathbf{X})\hat{\vartheta}_W(\mathbf{X}) < 0 | \hat{\vartheta}_m(\cdot), \hat{\vartheta}_W(\cdot)) < a_n$, for some $a_n \to 0$. Moreover, there is a monotone nondecreasing function $\Psi(\cdot)$, $\Psi(0) = 0$, such that

$$E\Big(Y\big(\mathbf{1}(\hat{\vartheta}_W(\mathbf{X}) > 0) - \mathbf{1}(\hat{\vartheta}_m(\mathbf{X}) > 0)\big) | \hat{\vartheta}_m(\cdot), \hat{\vartheta}_W(\cdot)\Big)$$
$$\leq \Psi\left(\frac{E(Y(\mathbf{1}(\hat{\vartheta}_W(\mathbf{X}) > 0) - \mathbf{1}(\hat{\vartheta}_m(\mathbf{X}) > 0)) | \hat{\vartheta}_m(\cdot), \hat{\vartheta}_W(\cdot))}{\sqrt{P(\hat{\vartheta}_m(\mathbf{X})\hat{\vartheta}_W(\mathbf{X}) > 0 | \hat{\vartheta}_m(\cdot), \hat{\vartheta}_W(\cdot))}}\right).$$

We essentially require that all procedures with close to optimal performance are similar.

THEOREM 2.1. *Let Assumption* S *hold. Then* $\eta_\tau = \eta_W + o_p(\Psi(\sqrt{\log M/k}))$.

PROOF. Let the two sets of indices postulated in Assumption S be S_1 and S_2, respectively. Since the estimates

$$k^{-1}\sum_{j=1}^{k} \mathbf{1}\big(Y_j^T \vartheta_m(\mathbf{X}_j^T) > 0\big), \qquad m = 1, \ldots, M,$$

have a uniform error bound of $\log(M)/\sqrt{k}$, we have $\tau \notin S_1$. Hence, with probability converging to 1, the test-bed stopping time is minimizing

$$U_m = k^{-1}\sum_{j=1}^{k}\Big(\mathbf{1}\big(Y_j^T \vartheta_W(\mathbf{X}_j^T) > 0\big) - \mathbf{1}\big(Y_j^T \vartheta_m(\mathbf{X}_j^T) > 0\big)\Big) \tag{4}$$

over $m \in S_2$. But the sum in (4) is of $\{-1, 0, 1\}$ i.i.d. random variables, which are 0 with high probability. Let p_m and q_m be the conditional probabilities (conditioned on the main sample) that a given term in the sum is 1 or -1, respectively. Then

$$EU_m = p_m - q_m,$$
$$\operatorname{Var} U_m = \big(1 + o(1)\big)(p_m + q_m)/k.$$

Hence, with probability converging to 1,

$$\eta_W - \eta_\tau = \max\left\{p_m - q_m : m \in S_2, \sqrt{\frac{k}{\log M}}\frac{p_m - q_m}{\sqrt{p_m + q_m}} < 1\right\}$$
$$\leq \max\left\{\Psi\left(\frac{p_m - q_m}{\sqrt{p_m + q_m}}\right) : m \in S_2, \Psi\left(\frac{p_m - q_m}{\sqrt{p_m + q_m}}\right) < \Psi\left(\sqrt{\frac{\log M}{k}}\right)\right\}$$
$$\leq \Psi\left(\sqrt{\frac{\log M}{k}}\right). \qquad \square$$

REFERENCES

[1] BARAUD, Y. (2002). Model selection for regression on a random design. *ESAIM Probab. Statist.* **6** 127–146.
[2] BICKEL, P. J. and RITOV, Y. (2003). Boosting and other iterative procedures. Unpublished manuscript.
[3] BREIMAN, L. (2000). Some infinity theory for predictor ensembles. Technical Report 577, Dept. Statistics, Univ. California, Berkeley.
[4] BÜHLMANN, P. (2002). Consistency for L_2Boosting and matching pursuit with trees and tree-type basis functions. Preprint. Available from stat.ethz.ch/~buhlmann/bibliog.html.
[5] BÜHLMANN, P. and YU, B. (2003). Boosting with the L_2 loss: Regression and classification. *J. Amer. Statist. Assoc.* **98** 324–339.
[6] FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion). *Ann. Statist.* **28** 337–407.
[7] MALLAT, S. and ZHANG, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **41** 3397–3415.
[8] ZHANG, T. and YU, B. (2003). Boosting with early stopping: Convergence and consistency. Technical Report 635, Dept. Statistics, Univ. California, Berkeley. Available from www.stat.berkeley.edu/~binyu/publications.html.

DEPARTMENT OF STATISTICS
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720-3860
USA
E-MAIL: bickel@stat.berkeley.edu

DEPARTMENT OF STATISTICS
HEBREW UNIVERSITY
91905 JERUSALEM
ISRAEL
E-MAIL: yaacov.ritov@huji.ac.il

DISCUSSION

BY PETER BÜHLMANN AND BIN YU

ETH Zürich and University of California, Berkeley

Jiang, Lugosi and Vayatis, and Zhang ought to be congratulated for their different works on the original AdaBoost algorithm with early stopping (Jiang), an ℓ_1-penalized version of boosting (Lugosi and Vayatis) and a convex minimization method which can be viewed as an ℓ_2-penalized version of boosting (Zhang).

1. A motivation for combining trees with boosting. The interesting and common part of all three papers is that Bayes risk consistency can be achieved by a linear or convex combination of simple classifiers. The most prominent examples, exhibiting good performance in a variety of datasets, are combinations of small or moderate-sized classification trees. Each of the trees is low-dimensional, but by linear or convex addition of trees we obtain a combined classifier whose complexity is (much) larger.

A problem with single classification trees is that they are often inflexible or cannot be constructed large enough for optimal classification performance. We

show in Figure 1 the test set misclassification loss (0–1 loss) for $n = 100$ i.i.d. realizations of (X, Y) in the model

$$X = (X_1, \ldots, X_{10}) \sim \text{Uniform}([-0.5, 0.5]^{10}), \qquad Y \in \{-1, 1\} \text{ with } \mathbb{P}[Y = 1] = p(X),$$
$$\text{logit}(p(x)) = \log\big(p(x)/\big(1 - p(x)\big)\big) = 50 \sum_{j=1}^{10} x_j. \tag{1}$$

The trees are constructed in a greedy way (as usual) optimizing the Gini index fitting criterion. We tuned the size of a classification tree by the minimal number of observations that fall into the terminal nodes, and the largest trees are constructed under the constraint that there are at least two observations per terminal node. We see in Figure 1 that on average, the largest classification trees have about 10 or 11 terminal nodes. We also see that the test set error is smallest at our largest tree, but we cannot make the trees larger (more complex) to potentially decrease the test set error (we could enlarge them a bit by requiring at least one observation per terminal node, but this turns out to be rather unstable with low predictive power). This has to do with two things: first, it is the constrained nature of trees with splits parallel to coordinate axes; second, a greedily constructed classification tree is restrictive and hence involves much fewer degrees of freedom (less complexity) than when constructed in a nongreedy way. Regarding the first issue, other proposals with splits that are not parallel to axes have been proposed; compare [8]; the second issue is more difficult, but recently some progress has been made in constructing trees in a more exhaustive, less greedy way [7]. The second remedy is nontrivial and with much higher computational costs.

Perhaps a conceptually simpler way, if we are concerned only with good classification performance, is given by boosting (AdaBoost which may be read as "ad a boost"), which "boosts" a single classification tree to make it more flexible and more complex. Figure 1 also shows how much the test set error could be improved by using LogitBoost (with the log-likelihood loss function) with stumps, namely by about 30%. Thus, from a practical point of view, linear or convex combinations of trees overcome the limitation of "bounded" complexity of single trees. Moreover, as we understand from rigorous results in the L_2-boosting case with squared error loss [3], the increase of complexity occurs in a very gradual fashion (much slower than counting the number of terms), which allows adaptation to problems of different complexity. Last but not least, boosting has also been found to have excellent performance in a wide range of real datasets. The papers under discussion justify such combination procedures which seem to act intelligently with the curse of dimensionality.

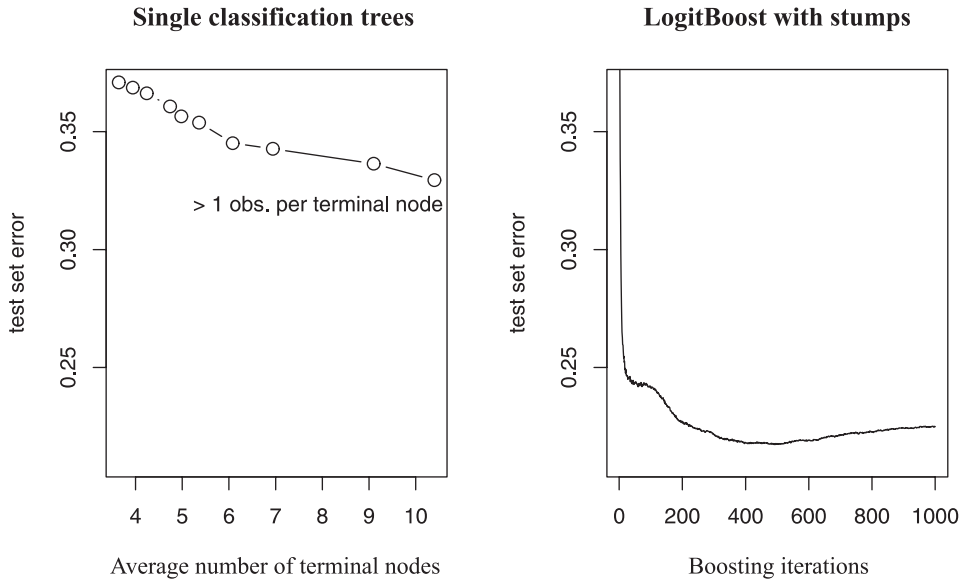

FIG. 1. *Test set misclassification errors in model* (1). Left panel: *Classification trees with varying minimal numbers of observations per terminal node, displayed as a function of average number of terminal nodes*; *the lower right circle corresponds to classification trees with at least two observations per terminal node*. Right panel: *LogitBoost with stumps as a function of boosting iterations*.

2. Boosting (with early stopping) versus regularized boosting. Jiang analyzes the original AdaBoost algorithm with early stopping, whereas versions of regularized boosting are considered by Lugosi and Vayatis (ℓ_1-constrained boosting) and also Zhang (ℓ_2-penalized boosting).

Computational advantage of boosting (*with early stopping*). The original boosting scheme specifies explicitly the numerical algorithm for optimization to be greedy, in contrast to many other classical statistical estimation schemes which are defined through an ideal optimization of an objective function. And we believe this original version of boosting (with early stopping) has an important computational advantage for coping with high-dimensional complex datasets having dimension of the predictor in the thousands. We think that it is exactly for such problems where boosting (using a learner which does variable selection) plays a significant role, since more traditional methods become very difficult to use and tune; for the latter, forward variable selection is still feasible, but assigning various smoothing parameters for selected predictors or terms is very difficult (see also the end of this section).

ℓ_1-constrained boosting is Lasso. The ℓ_1-constrained boosting algorithm proposed by Lugosi and Vayatis can be understood as seeking a combination

of base learners with an ℓ_1-constraint on the combination weights, that is, one minimizes the empirical risk $A_n(f)$ under the constraint $\sum_{j=1}^{N} w_j \leq \lambda$ (notation as in Lugosi and Vayatis). This is best known in the statistics community as the Lasso method ([11] or also as basis pursuit [4]) in signal processing.

Efficient computation of Lasso or basis pursuit is in general a nontrivial issue ([4] and [10]). A notable point is that Lasso solutions are usually *not* computed using greedy algorithms which are in danger of being overly greedy and can get stuck in suboptimal solutions. Lugosi and Vayatis use in their examples the MarginBoost.L_1 algorithm from [9]. It is a gradient descent, greedy forward method, very similar to boosting, which normalizes the ℓ_1-norm of the weights along the way. Interestingly, this MarginBoost.L_1 algorithm can be used for many base learners and is not restricted to specialized problems like linear regression or expansions from an over-complete dictionary. Lugosi and Vayatis do not discuss to what extent the MarginBoost.L_1 algorithm yields approximate solutions to Lasso-type problems or whether the MarginBoost.L_1 algorithm corresponds exactly to the ℓ_1-constrained boosting for which theoretical results are proven by Lugosi and Vayatis. In particular, at first sight, it seems that the greedy nature of the MarginBoost.L_1 algorithm used by Lugosi and Vayatis for their regularized boosting is in conflict with the nongreedy Lasso algorithms in [4] and [10].

Using the LARS (least angle regression) algorithm for finite linear regression models, Efron, Hastie, Johnstone and Tibshirani [5] recently made a connection between Lasso and boosting with infinitesimal shrinkage factor (or ε-boosting), or equivalently, linking nongreedy linear programming algorithms for Lasso with greedy, gradient descent methods for boosting with infinitesimal steps: ε-boosting (or "stagewise" as called by Efron, Hastie, Johnstone and Tibshirani [5]) adds normalized base learners to the current fit by an infinitesimal amount ε (but fixed among the boosting iterations). Under some positive cone conditions for the predictor variables, Efron, Hastie, Johnstone and Tibshirani [5] show that Lasso and ε-boosting are equivalent. In practice, the ε or infinitesimal amount of shrinkage has to be chosen as a small constant as has been advocated by Friedman [6]. However, it is worth noting that ε-boosting is not the same as MarginBoost.L_1, but we believe they are closely related.

Although this connection in the finite predictor (or finite base learner) case is intriguing, it is unclear how to generalize the LARS algorithm to the infinite base learner case. [One such example is trees, although for a given n, there are only finitely many possible trees with any fixed number of terminal nodes and split points in the middle between observations. However, this finite number is already equal to dim.(predictor) $\cdot (n-1)+1$ for stumps and even much bigger for larger trees; asymptotically, it is infinite.] This infinite base learner scenario is the most relevant to the success of boosting with empirical datasets because the base learner fitted at each step is taken from a pool of infinitely many base learners. For this case, we believe boosting (with small steps) provides the most flexible solution and, in some sense, generalizes LARS from the finite learner case. It is

interesting to note that the convergence analysis of Zhang and Yu [12] suggests that small step sizes are necessary for the convergence of the boosting algorithm as the iteration goes to infinity. For good statistical performance, however, we almost always stop before convergence and we believe that boosting is, in general, different from ℓ_1-constrained boosting or MarginBoost.L_1. This difference can also been seen in the experiments provided by Lugosi and Vayatis on AdaBoost and MarginBoost.L_1.

ℓ_2-regularization is Ridge. Zhang proposes a convex combination of base learners: his way of estimation and regularization is via the more established ℓ_2-penalty. Because of the ℓ_2-penalty, his algorithm can be viewed as a Ridge method. In general, the solution is not expected to be sparse. On the other hand, boosting with a base learner that does variable selection can be shown to have the interesting feature to do variable selection *and* assign varying amounts of degrees of freedom to different selected variables (e.g., in a linear model) or terms in an expansion (e.g., in fitting an additive model). The same holds for Lasso, which is also reflected by the "equivalence" of ε-boosting and Lasso [5].

Adaptivity of boosting (*with early stopping*). We have shown in [3] that boosting with the squared error loss function, which we called L_2Boost, adapts to higher-order smoothness for curve estimation in nonparametric regression. For example, when using cubic smoothing splines as base learners with a fixed conventional smoothing parameter λ_0, L_2Boost with a suitable number of boosting iterations achieves the minimax optimal MSE rates over Sobolev classes. Even though we are using a cubic smoothing spline as a base learner, L_2Boost achieves a faster MSE rate than $O(n^{-4/5})$ (the optimal rate for the Sobolev class of degree 2) if the underlying true function is in the Sobolev space of degree larger than 2 (essentially more than twice differentiable). With non-boosted smoothing splines, we would only get the minimax optimal MSE rates when knowing the smoothness of the underlying function. Thus, L_2Boost has the interesting theoretical property of adapting automatically to higher-order smoothness, and interestingly, this is achieved by a greedy forward algorithm!

Because of the connection between the ℓ_2-penalized convex combination algorithm of Zhang, when used with the squared error loss and the classical smoothing splines, we doubt that this adaptivity holds for Zhang's ℓ_2-regularized boosting algorithm. It remains to be seen whether the ℓ_1-constrained boosting has this adaptivity, but we conjecture that it does due to its connection to ε-boosting.

3. Final remarks. Jiang solved the problem of consistency for original boosting with early stopping which we think is a very effective statistical methodology and at the same time computationally feasible for high-dimensional data-sets. Breiman [1] pointed already at the issue of consistency for AdaBoost but Jiang was the first to prove consistency of AdaBoost. Since we believe that

boosting (with early stopping) is very useful in general, we have followed up on Jiang's work. In [2], consistency of L_2Boost (with early stopping) is proved for regression or probability estimation in classification (which is more general than Bayes risk consistency). More recently, Zhang and Yu [12] showed consistency of boosting with early stopping under general loss functions.

Lugosi and Vayatis present elegant consistency theorems which work under "minimal" assumptions. Since they analyze ℓ_1-constrained boosting, we may think that their result also hints at consistency for Lasso-type methods in classification.

Zhang's work has an interesting part on implementing loss functions for classification, providing consistency for Ridge-type methods in classification.

In summary, the three papers under discussion present some important recent understanding of boosting, as a result of the joint efforts of the statistics and the machine learning communities. We believe that this interaction of statistics and machine learning is bearing or will bear fruit on understanding many other procedures such as support vector machines and independent component analysis.

REFERENCES

[1] BREIMAN, L. (2004). Population theory for boosting ensembles. *Ann. Statist.* **32** 1–11.

[2] BÜHLMANN, P. (2002). Consistency for L_2Boosting and matching pursuit with trees and tree-type basis functions. Preprint. Available from stat.ethz.ch/~buhlmann/bibliog.html.

[3] BÜHLMANN, P. and YU, B. (2003). Boosting with the L_2 loss: Regression and classification. *J. Amer. Statist. Assoc.* **98** 324–339.

[4] CHEN, S. S., DONOHO, D. L. and SAUNDERS, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.* **20** 33–61.

[5] EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression. *Ann. Statist.* To appear.

[6] FRIEDMAN, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29** 1189–1232.

[7] GEMAN, D. and JEDYNAK, B. (2001). Model-based classification trees. *IEEE Trans. Inform. Theory* **47** 1075–1082.

[8] KIM, H. and LOH, W.-Y. (2001). Classification trees with unbiased multiway splits. *J. Amer. Statist. Assoc.* **96** 589–604.

[9] MASON, L., BAXTER, J., BARTLETT, P. and FREAN, M. (2000). Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers* (A. J. Smola, P. L. Bartlett, B. Schölkopf and D. Schuurmans, eds.) 221–247. MIT Press.

[10] OSBORNE, M. R., PRESNELL, B. and TURLACH, B. A. (2000). On the lasso and its dual. *J. Comput. Graph. Statist.* **9** 319–337.

[11] TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58** 267–288.

[12] ZHANG, T. and YU, B. (2003). Boosting with early stopping: Convergence and consistency. Technical Report 635, Dept. Statistics, Univ. California, Berkeley. Available from www.stat.berkeley.edu/~binyu/publications.html.

SEMINAR FÜR STATISTIK
ETH-ZENTRUM LEO C17
CH-8092 ZÜRICH
SWITZERLAND
E-MAIL: buhlmann@stat.math.ethz.ch

DEPARTMENT OF STATISTICS
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720-3860
USA
E-MAIL: binyu@stat.berkeley.edu

DISCUSSION

BY JEROME FRIEDMAN, TREVOR HASTIE, SAHARON ROSSET, ROBERT TIBSHIRANI AND JI ZHU

Stanford University

1. Introduction. We congratulate the authors for their interesting papers on boosting and related topics. Jiang deals with the asymptotic consistency of AdaBoost. Lugosi and Vayatis study the convex optimization of loss functions associated with boosting. Zhang studies the loss functions themselves. Their results imply that boosting-like methods can reasonably be expected to converge to Bayes classifiers under sufficient regularity conditions (such as the requirement that trees with at least $p + 1$ terminal nodes are used, where p is the number of variables in the model). An interesting feature of their results is that whenever data-based optimization is performed, some form of regularization is needed in order to attain consistency. In the case of AdaBoost this is achieved by stopping the boosting procedure early, whereas in the case of convex loss optimization, it is achieved by constraining the L_1 norm of the coefficient vector. These results reiterate, from this new perspective, the critical importance of regularization for building useful prediction models in high-dimensional space. This is also the theme of the remainder of our discussion.

Since the publication of the AdaBoost procedure by Freund and Schapire [6], there has been a flurry of papers seeking to answer the question: why does boosting work? Since AdaBoost has been generalized in different ways by different authors, the question might be better posed as: what are the aspects of boosting that are the key to its good performance?

2. Our view: boosting performs a high-dimensional Lasso. We would like to present our current view of boosting here. In recent years, a new paradigm has emerged in flexible function fitting. There are three ingredients:

- A large dictionary $\mathcal{D}$ of basis functions for representing the function, typically as a linear expansion $f(x) = \sum_{h_\ell \in \mathcal{D}} h_\ell(x)\beta_\ell$.
- A loss function $L(Y, f(X))$ appropriate for the problem, for example, for regression or classification.
- A regularizer $J(\beta)$ to control the size of the coefficients in the model.

One then fits the model by minimizing the sum over the data

$$\sum_{i=1}^{N} L\big(y_i, f(x_i)\big) + \lambda J(\beta), \tag{1}$$

where λ is a tuning parameter that controls the trade-off between average loss and penalty. If constructed appropriately, the resulting problem is convex and hence

can be solved by convex optimization methods. Support vector machines fall into this paradigm: they use an L_2 penalty, a piecewise-linear ("hinge") loss function and a basis dictionary generated by a positive definite kernel. Although such bases can have infinite dimension, the "kernel trick" results in a finite representation and simplifies the optimization [12].

Boosting methods use adaptively constructed basis functions and a forward stagewise procedure to build the model. In [9] we showed that AdaBoost fits an additive model in its basis functions, using a particular *exponential* loss function. This framework led to alternative and potentially better forms of boosting, by allowing the use of other loss functions and improvements in the forward stagewise procedure ([9] and [7]).

In this work we noticed that slowing down the procedure through shrinkage—a kind of *slow learning*—always seemed to help. This led us to our current view of boosting. We think of the forward stagewise procedure as a numerical device for approximating a sequence of solutions to (1) when $J(\beta)$ is an L_1 penalty. The sequence is obtained by continuously relaxing the parameter λ. Chapter 10 of [10] has a discussion of this point. More recently, Efron, Hastie, Johnstone and Tibshirani [5] proved a result in the simplified framework of least squares regression. Given a centered outcome variable $Y = \{y_i\}_1^n$ and standardized predictors $X_j = \{x_{ij}\}_1^n$, $j = 1, 2, \ldots p$, consider the following forward-stagewise procedure for estimating the coefficients $\beta = \{\beta_j\}_1^p$:

1. Start with $\beta_j = 0$ for all j, and the residual $r = Y$.
2. Find the predictor X_j most correlated with r, and increment its coefficient β_j by some small amount ε in the direction of this correlation,

$$\beta_j \leftarrow \beta_j + \varepsilon \cdot \text{sign}[\text{corr}(r, X_j)].$$

   Adjust r accordingly,

$$r \leftarrow r - X_j \cdot \varepsilon \cdot \text{sign}[\text{corr}(r, X_j)].$$

3. Repeat step 2 many times.

We call this "incremental forward stagewise regression." If this procedure is run for many steps, it eventually reaches the full least squares solution (modulo the granularity in ε). But more interestingly, we show in [5] that the resulting coefficient profiles approximate the solution to an L_1-constrained regression ("Lasso") $\beta(\lambda) = \arg\min \sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$. That is, the profiles of the coefficients resulting from incremental forward stagewise regression look much like the lasso solutions $\beta(\lambda)$, as λ is varied from $+\infty$ (maximum constraint) to 0 (no constraint). Figure 1 shows an example, taken from [10].

What does this have to do with boosting? Take as basis functions the set of all possible regression trees that can be grown from the given features. Suppose we want to compute the lasso path of solutions. This cannot be done directly since the number of trees is so large. Instead, take the incremental forward stagewise

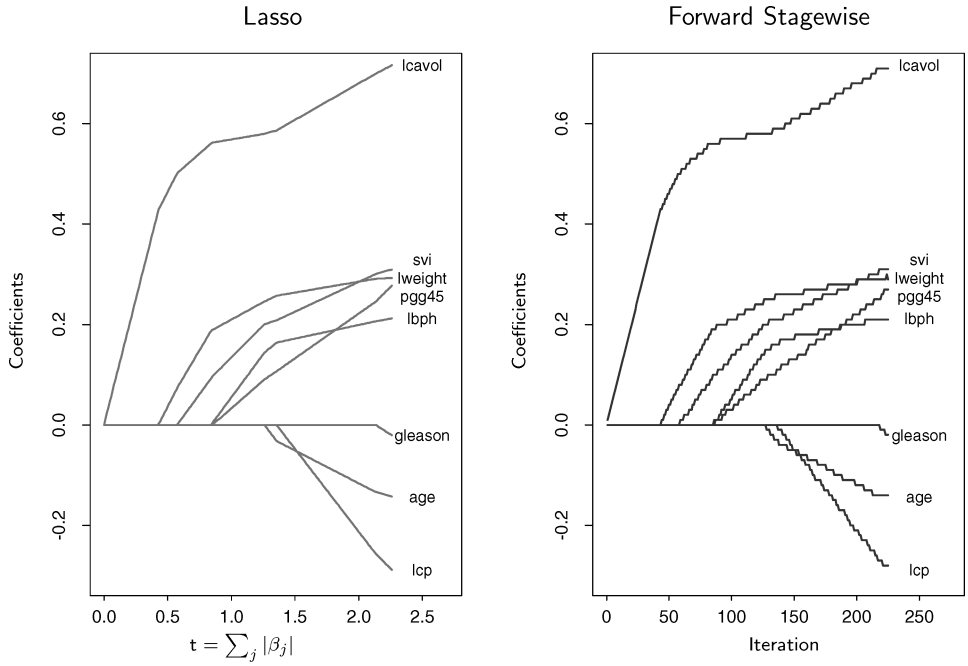

FIG. 1. *Profiles of estimated coefficients from linear regression, for the prostate data studied in Chapter* 3 *of* [10]. *The left panel shows the results from the Lasso, for different values of the bound parameter* $t = \sum_j |\beta_j|$. *The right panel shows the results of the incremental forward stagewise linear regression algorithm, using* $M = 250$ *consecutive steps of size* $\varepsilon = 0.01$.

regression and replace the predictors X_j with basis functions that are the set of all possible regression trees that can be grown from the given features. The least squares boosting procedure of Friedman [7] looks like the following:

1. Start with $F(x) = 0$ and the residual $r = Y$.
2. Fit a tree $f(x)$ to the outcome r, increment $F(x)$ with a shrunken version of $f(x)$,

$$F(x) \leftarrow F(x) + \varepsilon f(x),$$

and update r,

$$r \leftarrow r - \varepsilon f(x).$$

3. Repeat step 2 many times.

Now in step 2 when we fit a tree to r, we are approximately finding the tree (among all possible trees) that is most correlated with r. Hence least squares boosting can be viewed as a numerically savvy way of carrying out incremental forward stagewise regression on the space of regression trees. The latter, in turn, is an approximate way of computing the lasso path in this space.

For simplicity, our discussion has focussed on least-squares boosting. It also applies to other forms of boosting that use different loss functions [8], for example, AdaBoost, which is based on exponential loss [11].

3. The "bet on sparsity" principle. Now for any of this to be of practical importance, there must be an inherent reason (other than the ease of implementation) to prefer an L_1 penalty, to say an L_2 penalty, for these kinds of problems. Suppose we have 10K data points and our model is a linear combination of a million trees. Suppose also that the true population coefficients of these trees arose from a Gaussian distribution. Then we know that in a Bayesian sense the best predictor would be a ridge regression; that is, we should use an L_2 rather than an L_1 penalty when fitting the coefficients. On the other hand, if there are only a small number (e.g., 1000) of nonzero true coefficients, the Lasso (L_1 penalty) will work better. We think of this as a *sparse* scenario, while the first case (Gaussian coefficients) as *dense*. Note however that in the dense scenario, although the L_2 penalty is best, neither method does very well since there is too little data from which to estimate such a large number of nonzero coefficients. This is the *curse of dimensionality* taking its toll. In a sparse setting, we can potentially do well with the L_1 penalty, since the number of nonzero coefficients is small. The L_2 penalty fails again.

In other words, use of the L_1 penalty follows what we call the *bet on sparsity* principle for high-dimensional problems:

> *Use a procedure that does well in sparse problems, since no procedure does well in dense problems.*

These comments need the following moderation:

- For any given application, the degree of sparseness/denseness depends on the unknown true target function and the chosen dictionary $\mathcal{D}$.
- The notion of sparse vs. dense is relative to the size of the training data set and/or the signal-to-noise ratio (SNR). Larger training sets allow us to estimate coefficients with smaller standard errors. Likewise in situations with large SNR, we can identify more nonzero coefficients with a given sample size than in situations where the SNR is smaller.
- The size of the dictionary plays a role as well. Increasing the size of the dictionary may lead to a sparser representation for our function, but the search problem becomes more difficult.

Figure 2 illustrates these points in the context of linear regression. The details are given in the caption. Note that we are not using the training data to select λ, but rather are reporting the best possible behavior for each method in the different scenarios. The L_2 penalty performs poorly everywhere. The Lasso performs reasonably well in the only two situations where it can (sparse coefficients). As expected the performance gets worse as the SNR decreases and as the model becomes denser.

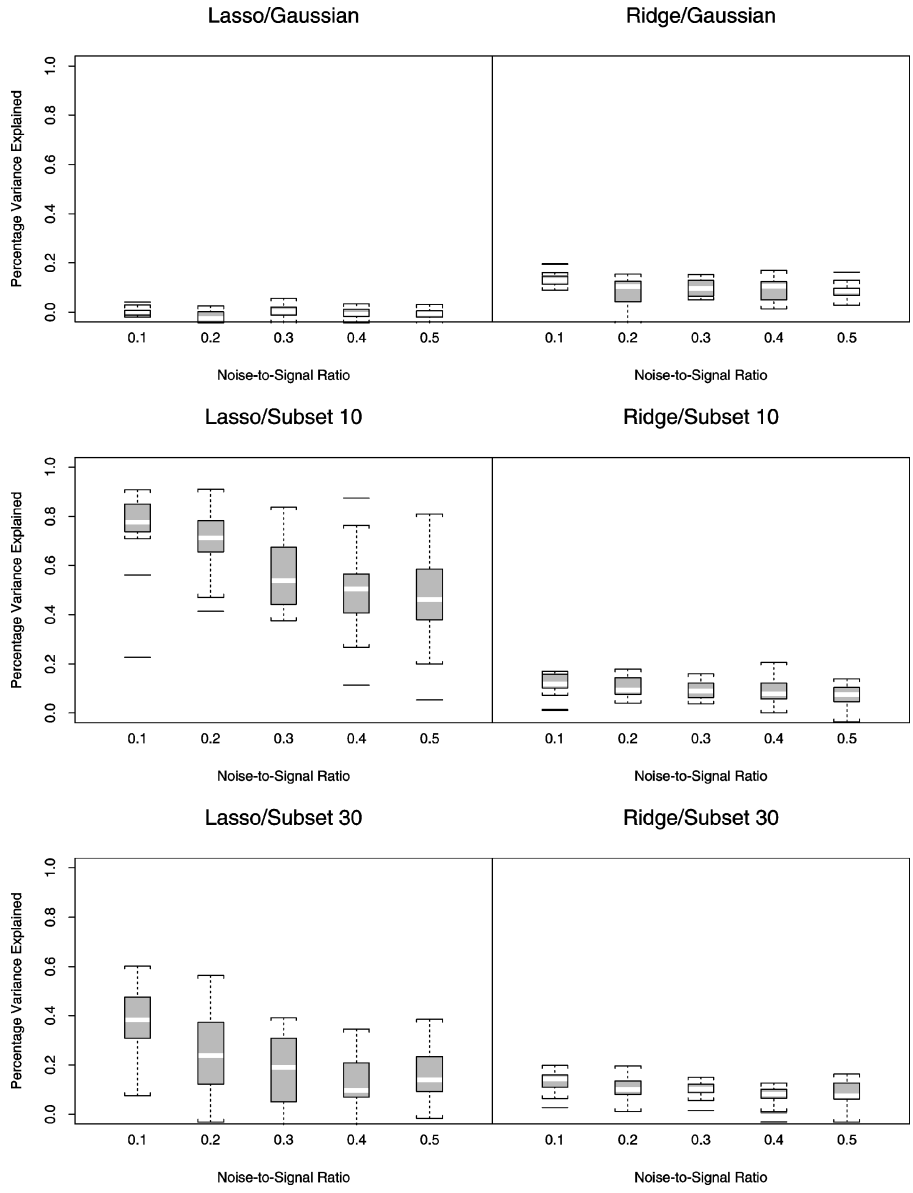

FIG. 2. *Simulations that show the superiority of the* L_1 (*Lasso*) *penalty over* L_2 (*Ridge*) *in regression. Each run has* 50 *observations with* 300 *independent Gaussian predictors. In the top row all* 300 *coefficients are nonzero, generated from a Gaussian distribution. In the middle row, only* 10 *are nonzero generated from a Gaussian, and the last row has* 30 *nonzero. In each case the coefficients are scaled so that the signal variance* $\mathrm{var}(X^T\beta)$ *is* 1. *The noise variance varies from* 0.1 *to* 0.5 (*noise to signal ratio*). *Lasso is used in the left column, Ridge in the right. In both cases we used a series of* 100 *values of* λ, *and picked the value that minimized the theoretical test error. In the figures we report the percentage variance explained* (*in terms of mean squared error*), *displayed as boxplots over* 20 *realizations for each combination.*

These empirical results are supported by a large body of theoretical results [1–4] that support the superiority of L_1 estimation in sparse settings.

REFERENCES

[1] DONOHO, D. and ELAD, M. (2002). Optimally-sparse representation in general (non-orthogonal) dictionaries via l_1 minimization. Technical report, Dept. Statistics, Stanford Univ.

[2] DONOHO, D. and JOHNSTONE, I. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81** 425–455.

[3] DONOHO, D., JOHNSTONE, I., HOCH, J. and STERN, A. (1992). Maximum entropy and the nearly black object (with discussion). *J. Roy. Statist. Soc. Ser. B.* **54** 41–81.

[4] DONOHO, D., JOHNSTONE, I., KERKYACHARIAN, G. and PICARD, D. (1995). Wavelet shrinkage: Asymptopia? (with discussion). *J. Roy. Statist. Soc. Ser. B.* **57** 301–369.

[5] EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression. *Ann. Statist.* To appear.

[6] FREUND, Y. and SCHAPIRE, R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning*: *Proc. 13th International Conference* 148–156. Morgan Kaufmann, San Francisco.

[7] FRIEDMAN, J. (1999). Greedy function approximation: The gradient boosting machine. Technical report, Dept. Statistics, Stanford Univ.

[8] FRIEDMAN, J. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29** 1189–1232.

[9] FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion). *Ann. Statist.* **28** 337–407.

[10] HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2001). *The Elements of Statistical Learning*: *Data Mining*, *Inference*, *and Prediction*. Springer, New York.

[11] ROSSET, S., ZHU, J. and HASTIE, T. (2002). Boosting as a regularized path to a maximum margin classifier. Technical report, Dept. Statistics, Stanford Univ.

[12] VAPNIK, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.

DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4065
USA
E-MAIL: hastie@stanford.edu

DISCUSSION

BY VLADIMIR KOLTCHINSKII

University of New Mexico

1. Boosting: numerical and statistical aspects. These three interesting papers explore (from somewhat different points of view) convergence properties of boosting methods of binary classification. It has become common to interpret these methods as minimization of empirical risk with an asymmetric loss function

(often chosen to be convex) that penalizes heavily for incorrect classification and even penalizes to some extent for correct classification with a small classification margin. Such a choice of the loss function allows one to improve the accuracy of approximation of the true risk by the empirical risk and to make empirical risk minimization computationally tractable (at the same time). It also explains the fact that classifiers obtained using these methods tend to have large positive margins. Boosting type algorithms search for large margin classifiers in the convex hull of a given base class of functions, the empirical risk minimization problem being solved using a functional version of iterative gradient descent method (which can be also interpreted as a stagewise fitting of additive logistic regression). The most famous representative of this class of algorithms is AdaBoost invented by Freund and Schapire several years ago. The properties of AdaBoost and subsequent boosting type algorithms, including their superb generalization ability and relative immunity to overfitting, are not as surprising now as they used to be when the algorithms were first suggested and tested in numerous experiments, but the theoretical explanation of these properties is still far from being complete. One of the difficulties with their analysis is related to the fact that these methods combine techniques of numerical optimization with techniques of statistical estimation, and therefore the analysis requires a subtle combination of the tools coming from both areas.

The problem of Bayes risk consistency of boosting methods was looked at by Leo Breiman a couple of years ago [1]. He eliminated the statistical part of the question by assuming that the amount of training data is infinite. This (not very realistic) assumption allowed him to explore approximation properties of a population version of AdaBoost giving a solution of approximation and numerical analysis parts of the problem (showing the convergence of the algorithm to the Bayes risk). From a somewhat different point of view, the numerical analysis part was also explored by Mason, Baxter, Bartlett and Frean [5]. The current paper of Tong Zhang (and some other related papers of this author) takes this line of research further. His main concern is a thorough study of the approximation error (in the current paper) and convergence properties of optimization algorithms involved in boosting in some other papers; see [7] and [8].

The paper of Lugosi and Vayatis, on the contrary, does not take into account the iterative nature of boosting algorithms, but views them as methods of precise minimization of the empirical risk and studies regularized versions of these algorithms (with the restrictions imposed on the sums of the weights of base classifiers). The main goal here is to prove consistency of regularized boosting (i.e., convergence a.s. of the generalization error of classifiers produced by the algorithm to the Bayes risk as the amount of the training data tends to infinity).

Jiang's idea is different. His version of regularization of AdaBoost is based on early stopping of the algorithm. He shows that the number of rounds of AdaBoost can be chosen (depending on the sample size) in such a way that the classifier output by the algorithm is consistent. In this context, I would like also to mention

the paper of Mannor, Meir and Zhang [4] and the more recent paper of Zhang and Yu [8] that develop Jiang's idea in a different fashion, that is, much closer to the work of Lugosi and Vayatis.

2. A simple proof of consistency. In this section (which will be much more formal than the previous one), I would like to explain my own understanding of the work of Lugosi and Vayatis (and of some elements of the work of Zhang). In particular, I would like to show a very easy proof of a slightly generalized version of Theorem 1 of Lugosi and Vayatis (based on empirical processes bounds from [2] they used as well). I will have to introduce some notation.

Let (X, Y) be a random couple in $S \times \{-1, 1\}$, P being the distribution of (X, Y) and Π being the distribution of X. For a classifier $g : S \mapsto \{-1, 1\}$,

$$L(g) := P\{(x, y) : y \neq g(x)\}$$

denotes its generalization error. The Bayes risk L^* is the minimum of $L(g)$ over all measurable classifiers $g : S \mapsto \{-1, 1\}$. This minimum is attained at the Bayes classifier

$$g^*(x) := I\big(\eta(x) > 0\big) - I\big(\eta(x) \leq 0\big),$$

where $\eta(x) = \mathbb{E}(Y|X = x)$ is the regression function. The following representation is well known:

$$L(g) - L^* = \int_{\{g=1\} \triangle \{g^*=1\}} |\eta| \, d\Pi. \tag{1}$$

Given a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ of i.i.d. copies of (X, Y) [defined on some probability space $(\Omega, \Sigma, \mathbb{P})$], let P_n denote the empirical distribution based on the sample. A sequence of classifiers $\hat{g}_n$ (also based on the sample) is called consistent iff $L(\hat{g}_n) \to L^*$ as $n \to \infty$ a.s.

Consider a class $\mathcal{F}$ of functions $f : S \mapsto \mathbb{R}$. The complexity of function classes used in learning problems is often characterized by the following quantity, called the Gaussian complexity:

$$G_n(\mathcal{F}) := \mathbb{E} \sup_{f \in \mathcal{F}} \left| n^{-1} \sum_{i=1}^{n} g_i f(X_i) \right|,$$

where $\{g_i\}$ are i.i.d. standard normal r.v.'s independent of $\{X_i\}$.

Let $\ell : \mathbb{R} \mapsto (0, +\infty)$ be a continuous loss function with $\ell(u) \to 0$ as $u \to +\infty$ and $\ell(u) \to +\infty$ as $u \to -\infty$. In what follows we denote $(\ell \bullet f)(x, y) := \ell(yf(x))$, $(x, y) \in S \times \{-1, 1\}$. Boosting type methods are often viewed as functional gradient descent algorithms of minimizing the empirical risk $P_n(\ell \bullet f)$ over $f \in \mathcal{F} := \operatorname{conv}(\mathcal{H})$, where $\mathcal{H}$ is a base class of functions.

Let

$$Q(u, \tau) := \ell(u)\frac{1 + \tau}{2} + \ell(-u)\frac{1 - \tau}{2}, \qquad u \in \mathbb{R}, \tau \in [-1, 1].$$

We assume in what follows that, for any $\tau \in (-1, 1)$, $u \mapsto Q(u, \tau)$ has the unique minimal point $m(\tau)$:

$$\min_{u \in \mathbb{R}} Q(u, \tau) = Q(m(\tau), \tau).$$

We also set $m(1) = +\infty$, $m(-1) = -\infty$ [the uniqueness of the minimum $m(\tau)$ holds, e.g., when ℓ is strictly convex]. Clearly, $u_n \to m(\tau)$ iff $Q(u_n, \tau) \to Q(m(\tau), \tau)$ [by uniqueness of $m(\tau)$ and continuity of Q]. Finally, we assume that $m(\tau) > 0$ (< 0) iff $\tau > 0$ (< 0). This condition holds if $\ell(u) > \ell(0) > \ell(-u)$ for all $u < 0$.

The following representation is easily proved by conditioning:

$$P(\ell \bullet f) = \int_S Q(f, \eta)\, d\Pi. \tag{2}$$

It immediately implies that the function $f^*(x) := m(\eta(x))$, $x \in S$, minimizes the risk $P(\ell \bullet f)$ over the class of all measurable functions $f : S \mapsto \mathbb{R}$.

Let now $\mathcal{F}_n$ be a sequence of classes of functions on S and let $\hat{f}_n \in \mathcal{F}_n$ denote a function that minimizes the empirical risk $P_n(\ell \bullet f)$ over $\mathcal{F}_n$. Let

$$\hat{g}_n(x) := I\big(\hat{f}_n(x) > 0\big) - I\big(\hat{f}_n(x) \le 0\big).$$

Suppose that all the functions in $\mathcal{F}_n$ are bounded by a constant $C_n > 0$ and let M_n denote the sup-norm and L_n the Lipschitz constant of ℓ on the interval $[-C_n, C_n]$.

THEOREM 1. *Suppose that*

$$\inf_{f \in \mathcal{F}_n} P(\ell \bullet f) \to P(\ell \bullet f^*) \qquad \text{as } n \to \infty, \tag{3}$$

$$\frac{M_n \sqrt{\log n}}{\sqrt{n}} \to 0 \quad \text{and} \quad L_n G_n(\mathcal{F}_n) \to 0 \qquad \text{as } n \to \infty. \tag{4}$$

Then $\{\hat{g}_n\}$ is consistent.

Let

$$\Delta_n := \sup_{f \in \mathcal{F}_n} \big|(P_n - P)(\ell \bullet f)\big|.$$

We will use the following empirical process bound that was involved in the proofs of many statements in [2] (see Theorems 1 and 2 there):

$$\mathbb{P}\left\{\Delta_n \ge \sqrt{2\pi} L_n G_n(\mathcal{F}_n) + \frac{t M_n}{\sqrt{n}}\right\} \le e^{-2t^2}, \quad t > 0. \tag{5}$$

We also use a well-known characterization of convergence in measure: a sequence of functions converges in measure iff it is possible to extract from any of its subsequences a subsequence that converges a.s.

PROOF OF THEOREM 1. Using conditions (4), bound (5) immediately implies that $\Delta_n \to 0$ $\mathbb{P}$-a.s. (just take $t_n := \sqrt{\log n}$ and use the Borel–Cantelli lemma). Since

$$P(\ell \bullet \hat{f}_n) - \inf_{f \in \mathcal{F}_n} P(\ell \bullet f) \le P_n(\ell \bullet \hat{f}_n) - \inf_{f \in \mathcal{F}_n} P_n(\ell \bullet f) + 2\Delta_n = 2\Delta_n,$$

we also have, using (3), $P(\ell \bullet \hat{f}_n) \to P(\ell \bullet f^*)$, $\mathbb{P}$-a.s. Representation (2) yields

$$\int_S \bigl[Q(\hat{f}_n, \eta) - Q(m(\eta), \eta)\bigr] d\Pi \to 0 \qquad \text{as } n \to \infty, \ \mathbb{P}\text{-a.s.},$$

which implies (since the integrand is nonnegative) that $\mathbb{P}$-a.s. $Q(\hat{f}_n, \eta) \xrightarrow{\Pi} Q(m(\eta), \eta)$. Recall that $m(\eta)$ is the unique minimum of $u \mapsto Q(u, \eta)$ and $Q(u, \eta)$ is continuous with respect to u. Therefore, by the characterization of convergence in measure, $\hat{f}_n \xrightarrow{\Pi} m(\eta)$. It remains to apply formula (1),

$$L(\hat{g}_n) - L^* = \int_{\{\hat{f}_n \le 0\} \triangle \{\eta \le 0\}} |\eta| \, d\Pi = \int_S \eta_n \, d\Pi,$$

where

$$\eta_n := |\eta| \bigl(I(\hat{f}_n \le 0, \eta > 0) + I(\hat{f}_n > 0, \eta \le 0)\bigr),$$

and to recall that $\eta > 0$ (≤ 0) iff $m(\eta) > 0$ (≤ 0). Therefore, the convergence $\hat{f}_n \xrightarrow{\Pi} m(\eta)$ implies $\eta_n \xrightarrow{\Pi} 0$ (everything is happening $\mathbb{P}$-a.s.!). Thus, by dominated convergence, $L(\hat{g}_n) - L^* \to 0$, implying the statement. $\square$

To apply Theorem 1 in a standard framework of boosting type algorithms, take $\mathcal{F}_n := C_n \mathcal{F}$, where $\mathcal{F} := \operatorname{conv}(\mathcal{H})$, $\mathcal{H}$ is a Π-Donsker class (in particular, it can be a VC-class) uniformly bounded by 1, and $C_n \to \infty$. Then

$$G_n(\mathcal{F}_n) \le C_n G_n\bigl(\operatorname{conv}(\mathcal{H})\bigr) = C_n G_n(\mathcal{H}) \le \frac{K C_n}{\sqrt{n}},$$

since for Π-Donsker classes $G_n(\mathcal{H}) \le \frac{K}{\sqrt{n}}$ with some constant $K > 0$. Therefore, condition (4) is satisfied as soon as $\frac{L_n C_n}{\sqrt{n}} \to 0$ and $\frac{M_n \sqrt{\log n}}{\sqrt{n}} \to 0$, which immediately implies one of the main results of Lugosi and Vayatis (their Theorem 1). One could expect that many other consistency results (e.g., for kernel machines) should follow easily from Theorem 1. Also, one can replace the assumption that $\hat{f}_n$ is a precise minimizer of $P_n(\ell \bullet f)$ over $\mathcal{F}_n$ by a weaker assumption that it is an approximate minimizer (for instance, an output of an iterative minimization algorithm after a certain number of iterations). In combination with the result of Zhang [7] on convergence rates of such optimization procedures for convex loss functions this leads to a consistency result for a version of boosting with an early stopping, somewhat in the spirit of Jiang's paper, but different (see also [4]).

3. What is next? Clearly, convergence rates of boosting type methods to the Bayes risk is a very important problem to look at. Some preliminary results are easy to obtain based on the estimates used in the proof of consistency (see, e.g., [4]). However, these rates are rather slow and it is well known that the convergence rates in classification problems can be very fast (can approach n^{-1} in the zero error case, i.e., when $L^* = 0$). More surprisingly, recent results of Mammen and Tsybakov [3] and subsequent results of Tsybakov [6] showed us that very fast convergence rates can also occur in the case when $L^* > 0$ [under special conditions on the distribution function of $|\eta(X)|$]. Tsybakov also defined adaptive classifiers for which these rates are attained. They are based on the empirical risk minimization over δ-nets chosen in the families of possible classifiers. Since the cardinalities of these δ-nets grow exponentially with the sample size, the algorithm becomes computationally intractable. It is reasonable to try to replace Tsybakov's algorithm with a boosting type algorithm, searching for a good classifier in the convex hull of a properly chosen base class (recent work of Bartlett, Jordan and McAuliffe deals with this type of problem). However, reproducing Tsybakov's convergence rates for boosting type methods will not be the end of the story. In my view, the main difficulty one has to deal with in classification problems is that there is no unique way to define the complexity of the problem, but rather a variety of possible ways. For many years, VC-dimension and related quantities have been viewed as very natural measures of complexity of classes of functions (sets) involved in the problem. The discovery of support vector machines and boosting methods changed this view rather dramatically. It became clear that relevant complexity measures are more complicated, for instance, they should take into account classification margins. The comprehensive nonparametric theory of classification will have to study how the optimal convergence rates are related to various notions of complexity brought into the learning theory in the recent years.

REFERENCES

[1] BREIMAN, L. (2000). Some infinity theory for predictor ensembles. Technical Report 577, Dept. Statistics, Univ. California, Berkeley.

[2] KOLTCHINSKII, V. and PANCHENKO, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *Ann. Statist.* **30** 1–50.

[3] MAMMEN, E. and TSYBAKOV, A. (1999). Smooth discrimination analysis. *Ann. Statist.* **27** 1808–1829.

[4] MANNOR, S., MEIR, R. and ZHANG, T. (2002). The consistency of greedy algorithms for classification. In *Proc. 15th Annual Conference on Computational Learning Theory. Lecture Notes in Comput. Sci.* **2375** 319–333. Springer, New York.

[5] MASON, L., BAXTER, J., BARTLETT, P. and FREAN, M. (2000). Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers* (A. Smola, P. Bartlett, B. Schölkopf and D. Schuurmans, eds.) 221–247. MIT Press.

[6] TSYBAKOV, A. (2004). Optimal aggregation of classifiers in statistical learning. *Ann. Statist.* **32** 135–166.

[7] ZHANG, T. (2002). Sequential greedy approximation for certain convex optimization problems. Technical Report RC22309, T. J. Watson Research Center, IBM, Yorktown Heights, NY.

[8] ZHANG, T. and YU, B. (2003). Boosting with early stopping: Convergence and consistency. Technical Report 635, Dept. Statistics, Univ. California, Berkeley. Available from www.stat.berkeley.edu/~binyu/publications.html.

DEPARTMENT OF MATHEMATICS
AND STATISTICS
UNIVERSITY OF NEW MEXICO
ALBUQUERQUE, NEW MEXICO 87131-1141
USA
E-MAIL: vlad@math.unm.edu

DISCUSSION

BY YOAV FREUND AND ROBERT E. SCHAPIRE

Columbia University and Princeton University

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 [3, 7, 13, 14]; 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 [8]. 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 [8] 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] and [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: do not 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 HMMs for speech analysis really thinks that speech can be synthesized by these HMMs. 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 exist 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),

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

that is, 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

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

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 tth 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 Shafer and Vovk [18].

REFERENCES

[1] COLLINS, M., SCHAPIRE, R. E. and SINGER, Y. (2002). Logistic regression, AdaBoost and Bregman distances. *Machine Learning* **48** 253–285.

[2] FREUND, Y. (1995). Boosting a weak learning algorithm by majority. *Inform. and Comput.* **121** 256–285.

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

[4] FREUND, Y. and SCHAPIRE, R. E. (1996). Game theory, on-line prediction and boosting. In *Proc. 9th Annual Conference on Computational Learning Theory* 325–332. ACM Press, New York.

[5] FREUND, Y. and SCHAPIRE, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* **55** 119–139.

[6] FREUND, Y. and SCHAPIRE, R. E. (1999). Adaptive game playing using multiplicative weights. *Games Econom. Behav.* **29** 79–103.

[7] FRIEDMAN, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29** 1189–1232.

[8] FRIEDMAN, J. H., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion). *Ann. Statist.* **28** 337–407.

[9] HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2001). *The Elements of Statistical Learning*: *Data Mining*, *Inference*, *and Prediction*. Springer, New York.
[10] KIVINEN, J. and WARMUTH, M. K. (1999). Boosting as entropy projection. In *Proc. 12th Annual Conference on Computational Learning Theory* 134–144. ACM Press, New York.
[11] KOLTCHINSKII, V. and PANCHENKO, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *Ann. Statist.* **30** 1–50.
[12] LEBANON, G. and LAFFERTY, J. (2002). Boosting and maximum likelihood for exponential models. In *Advances in Neural Information Processing Systems* **14**. MIT Press, Cambridge, MA.
[13] MASON, L., BAXTER, J., BARTLETT, P. and FREAN, M. (2000a). Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers* (A. J. Smola, P. L. Bartlett, B. Schölkopf and D. Schuurmans, eds.) 221–247. MIT Press, Cambridge, MA.
[14] MASON, L., BAXTER, J., BARTLETT, P. and FREAN, M. (2000b). Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems* **12**. MIT Press, Cambridge, MA.
[15] RÄTSCH, G., WARMUTH, M., MIKA, S., ONODA, T., LEMM, S. and MÜLLER, K.-R. (2000). Barrier boosting. In *Proc. 13th Annual Conference on Computational Learning Theory* 170–179. Morgan Kaufmann, San Francisco.
[16] SCHAPIRE, R. E. (1990). The strength of weak learnability. *Machine Learning* **5** 197–227.
[17] SCHAPIRE, R. E., FREUND, Y., BARTLETT, P. and LEE, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Statist.* **26** 1651–1686.
[18] SHAFER, G. and VOVK, V. (2001). *Probability and Finance*. *It's Only a Game*! Wiley, New York.
[19] VALIANT, L. G. (1984). A theory of the learnable. In *Proc. 16th Annual ACM Symposium on Theory of Computing* 436–445. ACM Press, New York.
[20] VAPNIK, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.

CENTER FOR COMPUTATIONAL LEARNING SYSTEMS
COLUMBIA UNIVERSITY MC 4750
500 WEST 120TH STREET
NEW YORK, NEW YORK 10027
USA

DEPARTMENT OF COMPUTER SCIENCE
PRINCETON UNIVERSITY
35 OLDEN STREET
PRINCETON, NEW JERSEY 08544
USA
E-MAIL: schapire@cs.princeton.edu

REJOINDER

BY WENXIN JIANG

Northwestern University

1. Comments and discussion. I thank the discussants for their insightful comments and new contributions, and thank Jon A. Wellner for arranging this discussion. I also congratulate the authors of the other papers under discussion and thank them for their significant works that are both independent and also collaborative in a general sense. My paper is a small step built on important previous works of other people: Freund and Schapire [7] invented

the popular AdaBoost algorithm. Brieman [2] and Schapire and Singer [18] identified the exponential criterion in relation to AdaBoost. Friedman, Hastie and Tibshirani [9] found the minimizer of the exponential criterion in the population case. Breiman [3] solved the difficult convergence problem for AdaBoost in the population case, showing that the iterations indeed approach the right minimum.

When I started to work on this topic a few years ago, AdaBoost seemed a mysterious and interesting puzzle. Although it performs well and is often resistant against overfitting, I soon found that in all analytic examples I can work out, AdaBoost can overfit when run for a sufficiently long time, for example, of order $t \sim n^2 \log n$, where n is the sample size; see [12] and [13]. [Of course, overfitting has been noticed in experiments or anticipated conceptually by other people as well, e.g., Grove and Schuurmans [10], Mason, Baxter, Bartlett and Frean [16], Friedman, Hastie and Tibshirani [9] and Bickel (private communication).] So when Breiman brought to my attention his paper proving that the population version ($n = \infty$, roughly speaking) of AdaBoost does *not* overfit and is consistent at $t \to \infty$, there seemed to be an apparent contradiction that I had to resolve.

To find a compromise, I regard Breiman's situation as $t \prec n$ (since n is already infinite), while the situations considered in [13] are $t \succ n$. The situations are different. It then occurs to me that Breiman's result suggests that a consistent AdaBoost solution may be obtained in the finite sample situation as well, if t is chosen to increase with n at a rate that is *not too fast*, so as to prevent the overfitting situation that I considered before.

Starting with this conjecture, I wrote up this note. It was originally intended to be a short communication, since I was not satisfied with the restrictive framework, conditions and results. However, to my relief, I noticed follow-up works that have made significant improvements in several directions. So now I think this short communication is at least very successful in this regard, for example, to induce other works that are better and more comprehensive.

Several interesting new results are described in the discussions. Bartlett, Jordan and McAuliffe obtain a general comparison theorem with necessary and sufficient conditions relating the consistency in prediction and the consistency in minimizing the "working" cost function. Bickel and Ritov, in a very efficient way, outline a consistency proof for boosting with truncation with a general cost function, and justify the use of cross-validation for implementing the truncation. Bühlmann and Yu point out the computational advantage of the truncation method and introduce some of their promising work in this direction that is independent of Bickel and Ritov: Bühlmann's work on consistency of L_2Boost, and Zhang and Yu's work on convergence and consistency of boosting with a general cost function.

Friedman, Hastie, Rosset, Tibshirani and Zhu investigate the close relationship between boosting and L_1 penalty and show how this might benefit in the case of "sparsity." This connection was also discussed by Bühlmann and Yu and both discussions refer to the interesting work by Efron, Hastie, Johnstone and Tibshirani [1] on boosting with infinitesimal steps. Koltchinskii provides an alternative proof

for the consistency result in Lugosi and Vayatis' paper under discussion. Freund and Schapire raise several interesting points that are common to all papers under discussion. I will only focus on a few of their remarks. I am sure the other authors can provide better replies and I will rely on them to respond to the other points.

Freund and Schapire rightly pointed out that the consistency results do not seem to explain the good finite sample performance of the boosting algorithms when handling high-dimensional data. They recommend explanations from margin theory. On the other hand, most other discussants implicitly or explicitly (e.g., Bartlett, Jordan and McAuliffe; Koltchinskii) suggest future work in studying convergence rates. I also think the study of convergence rates is more promising. The currently available margin bounds do not compare to the Bayes error and typically cannot be tight in the case of noisy data.

Freund and Schapire also raised the question of regularization of AdaBoost: is it unnecessary or potentially beneficial? I tend to agree with the second choice. Especially in noisy cases, regularized variants have been reported to lead to improvements. See, for example, [4, 8, 14, 16].

2. Future directions. Future efforts are most effective if both experimental people and theorists (they can be the same persons of course) collaborate very closely. Analytic studies alone can reveal some insights but are often limited to idealized cases. Experimental studies sometimes fail to generate information that might be important in reaching a good understanding.

For (an old) example, as far as I know, there is still no complete understanding of the most mysterious behavior of AdaBoost: *In some situations the training error becomes zero*, *but the prediction error still continues to decrease.* Partial explanation was made in [17] based on semiempirical upper bounds. Theoretical studies obtained exact solutions only in the one-dimensional case, where it is proved that AdaBoost with trees generates zero training error in finite time and converges to a nearest neighbor rule (see [12]). On the other hand, in higher dimensions a rule that fits the training sample perfectly can be very different from the nearest neighbor rule. *After a perfect fit on the training sample*, *does the prediction error approach something that is about the same as the nearest neighbor error*, *or not as good, or magically better*? Apparently, only in the last possibility will this mystery be worth studying, for otherwise one could use a nearest neighbor rule to do as good a job or better. However, in the experimental results that reported such a mystery, in no case was the noise level or nearest neighbor error reported. The reporting of the nearest neighbor error in such cases with zero training error could help us understand when such a mystery will occur (in noisy or noiseless cases), and whether this mystery is worth studying (whether it will beat the natural benchmark, the nearest neighbor error). A coordinated effort in both experiments and theory seems needed.

As many discussants point out, another promising direction is the study of convergence rates for variants of boosting algorithms, possibly regularized, in

various combinations of base learners and situations of data (e.g., noisy or not, sparseness or denseness, as Friedman, Hastie, Rosset, Tibshirani and Zhu commented). There are already some preliminary steps made in this direction, for example, [5, 12, 15]. *Studying the convergence rates in various interesting situations for various methods could further our knowledge on when boosting will work well and when it can be improved and how.* In the next section I will try to explain these points by a simple example.

3. A simple example. The following example involves a regularization method that averages over AdaBoost predictions from several subsamples. It is motivated from slightly diffferent bag-boosting schemes described by Bühlmann and Yu [4] and Krieger, Long and Wyner [14], and is closer to the latter reference. I became interested in this regularization scheme due to the good performance reported in the experiments of Krieger, Long and Wyner, and due to the modularity of its implementation. Again, I can only obtain analytic results in the idealized one-dimensional case described below.

Consider a setup similar to Section 5.2 of [12], $X \sim \text{Unif}[0, 1]$. The base hypothesis space is the space of "stumps" or a more general space which contains piecewise constant hypotheses, having splits chosen from mid-data points. The regularization scheme involves averaging subsample predictions as follows.

ALGORITHM (Averaged AdaBoost from subsamples).

(i) Divide the training sample $(X_i, Z_i)_1^n$ randomly into K subsamples of size m. (For convenience we assume $n = Km$.)

(ii) For subsample $k = 1, \ldots, K$, run AdaBoost t-steps and define the resulting prediction rule (at any $x \in [0, 1]$) as $\hat{z}_k^{(t)}(x)$.

(iii) Compute the average of these predictions $\bar{z}_K^{(t)} = K^{-1} \sum_{k=1}^K \hat{z}_k^{(t)}$ and use $\hat{Z}_K^t(x) \equiv \text{sgn}(\bar{z}_K^{(t)}(x))$ to predict the value of the unknown label Z for a future observation with $X = x$.

We will study cases with large t's so that AdaBoost already overfits the individual subsamples, and investigate how averaging over K subsample results remedies the overfit. Here K provides additional freedom for regularizing AdaBoost and measures the level of regularization (nonregularized AdaBoost has $K = 1$). We will consider what convergence rate of the resulting prediction can be achieved in the following three situations. Denote the (conditional) *probability function* $\pi(x) = P(Z = 1|X = x)$.

(A) (Noiseless with finite number of jumps). $\pi(x) \in \{0, 1\}$ for all x and is piecewise constant with at most J jumps. (J is a positive integer.)

(B) (Lipschitz). $|\pi(x) - \pi(x')| < D\varepsilon$ whenever $|x - x'| < \varepsilon$ for all small ε.

(C) (Finite number of finite "sign-changes"). $|\pi(x) - 0.5| \geq \delta$ for all x, for some $\delta \in (0, 0.5]$. Also, $\operatorname{sgn}\{\pi(x^-) - 0.5\} \neq \operatorname{sgn}\{\pi(x^+) - 0.5\}$ for at most J locations of x. (J is a positive integer.)

PROPOSITION (Averaged AdaBoost and convergence rates). *Denote* $L = P[\hat{Z}_K^t(X) \neq Z]$ *as the prediction error,* $L^* = E\min\{\pi(X),\ 1 - \pi(X)\}$ *as the Bayes error. The following results hold for all* $t \geq 2m^2\log(m+1)$, *where* $m = n/K$ *is the size of the subsamples and* t *is the number of boosting steps.*

(a1) (Remark 3(c), Jiang [11]). *For the noiseless class* (A), *taking* $K = 1$ (*no subsampling*) *leads to*

$$L - L^* \leq 2Jn^{-1}\log n\{1 + o_n(1)\}.$$

(a2) (Theorem 3, Jiang [12]). *In the general noisy situations, however, taking* $K = 1$ *for our current case of large* t *can lead to inconsistency*:

$$L - L^* = E\big[2|\pi(X) - 0.5|\min\big\{\pi(X),\ 1 - \pi(X)\big\}\big] + o_n(1).$$

(b1) *For the Lipschitz case* (B), *denote the "noise level"* $E\operatorname{var}(Y|X) = \sigma^2$ *and assume* $\sigma > 0$ [*here* $Y = (Z+1)/2$ *is the binary response*]. *Let the "signal to noise ratio"* $\mathrm{SNR} = D/\sigma$ *where* D *is the Lipschitz constant. Then, taking* $K \approx (n/\mathrm{SNR})^{2/3}$ *leads to*

$$L - L^* \leq 2\sigma(n/\mathrm{SNR})^{-1/3}\log(n/\mathrm{SNR})^{1/3}\{1 + o_n(1)\}.$$

(b2) *For smooth cases in* (B) *with continuous derivative* $\pi'(\cdot)$, *result* (b1) *can be strengthened by taking* $\mathrm{SNR} = E|\pi'(X)|/\sigma$ (*assume* $E|\pi'(X)| > 0$) *in the formulas of* K *and* $L - L^*$.

(c) *For the possibly noisy case* (C) *with a finite number of finite "sign-changes," if we take* $K \approx (2\delta^2)^{-1}\log n$, *we have*

$$L - L^* \leq 0.5J\delta^{-4}n^{-1}(\log n)^2\{1 + o_n(1)\}.$$

REMARKS.

1. These results show that *different data situations entail different regularization strategies and allow different convergence rates*. An important indicator for characterizing various situations is the noise level, which can be defined in several ways: as the average conditional variance (which is essentially the nearest neighbor error), as the average conditional standard deviation or as the Bayes error. When $\pi \in \{0, 1\}$, all such measures should indicate zero noise.
2. In the noiseless case (A), boosting without regularization, $K = 1$, is already near optimal, that is, achieves a rate $n^{-1}\log n$ that is within $\log n$ of the minimax rate n^{-1} for noiseless learning (see, e.g., [6], Theorem 14.1 or [11], Proposition 4).

3. In the noisy situations, nonregularized boosting is generally inconsistent. However, for noisy cases in (B), regularization with averaged subsample predictions can achieve a near optimal rate $n^{-1/3}\log n$, within $\log n$ of the minimax rate $n^{-1/3}$ for a Lipschitz family (see, e.g., [19]).
4. In the noisy case described in (C), a better rate $n^{-1}(\log n)^2$ can be obtained, which is within $(\log n)^2$ of the best possible [the minimax rate $1/n$ for the noiseless case (A) holds here too since (A) $\subset$ (C)]. This has no actual contradiction to the minimax result $n^{-1/3}$ above for case (B), since (B) can allow "difficult" functions such as $\pi = 0.5 + 0.5x^2\sin(x^{-1})$, which can cross 0.5 infinitely often (lack of "sparsity"), and can become arbitrarily close to 0.5, while such probabililty functions are excluded from (C).
5. Note that the results in this proposition suggest different levels of regularization for different types of problems. In noisy situations, overfitting is prevented by averaging over multiple subsamples. However, case (C) suggests the use of much fewer subsamples than case (B). But if the data are noiseless after all, it is possible that regularization might actually hurt the performance.
6. Even when restricted to consider smooth probability functions in case (B), results (b1, 2) still suggest that one should use more subsamples when there is higher noise σ and fewer when σ is low. Prior knowledge of the noise level or knowledge of σ from a two-stage procedure (using the maximum possible σ-value 0.5 in the first stage) might help.
7. In case (B), the regularization level used in results (b1, 2) actually produces reliable estimation of the probability function $\pi(x)$. The average of AdaBoost predictions $\bar{z}_K^{(t)}(x)$, before the sign transformation, estimates the mean function $E(Z|X = x) = 2\pi(x) - 1$ at a near optimal rate.
8. The implementation of this regularization scheme is simple and modular, since it only involves manipulation of outcomes from the standard AdaBoost algorithm. Subsamples can also be processed in parallel. Although all the results are derived for one-dimensional X, we suspect that the performance of this algorithm, with suitable choice of the number of subsamples and the number of boosting steps, will also be good in higher dimensions. A similar "bag-boosting" algorithm, where the subsamples can overlap, has shown good numerical performance in higher dimensions in both noisy and noiseless situations [14].
9. The key to the proof of these results is to notice that for sufficiently large t, the AdaBoost prediction from each subsample becomes the nearest neighbor rule. Then averaging these nearest neighbor rules, based on independent subsamples, can be easily shown to prevent overfit, even when the subsamples have already been overfitted individually. In practice, using smaller t (and K) may also generate good performance; see [14].

For a particular dataset, using a "dataset-based" procedure (such as cross-validation) to determine the level of regularization can generate better results

than just relying on what is suggested by convergence rate analyses. However, the procedure can then become either more computationally involved or can less efficiently utilize the whole dataset. A purely "dataset-based" approach also provides less understanding of the general behavior of boosting algorithms, compared to a "situation-based" convergence rate analysis. It may also be possible to combine the two approaches in some sense, by using some kind of a two-stage analysis (see Remark 6), or by incorporating the knowledge from a convergence rate analysis to reduce the range of searching for a best regularization parameter in cross-validation.

Clearly, further studies on regularization schemes are needed, with attention paid to all three sides: the *theoretical side* (on consistency and convergence rates in various interesting situations), the *numerical side* (on finite sample performance), as well as the *practical side* (on the ease of implementation and computation).

ADDITIONAL REFERENCES

[1] EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression. *Ann. Statist.* To appear.

[2] BREIMAN, L. (1999). Prediction games and arcing algorithms. *Neural Computation* **11** 1493–1517.

[3] BREIMAN, L. (2000). Some infinity theory for predictor ensembles. Technical Report 577, Dept. Statistics, Univ. California, Berkeley.

[4] BÜHLMANN, P. and YU, B. (2001). Discussion of "Additive logistic regression: A statistical view of boosting," by J. Friedman, T. Hastie and R. Tibshirani. *Ann. Statist.* **28** 377–386.

[5] BÜHLMANN, P. and YU, B. (2003). Boosting with the L_2 loss: Regression and classification. *J. Amer. Statist. Assoc.* **98** 324–339.

[6] DEVROYE, L., GYÖRFI, L. and LUGOSI, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

[7] FREUND, Y. and SCHAPIRE, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* **55** 119–139.

[8] FRIEDMAN, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29** 1189–1232.

[9] FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion). *Ann. Statist.* **28** 337–407.

[10] GROVE, A. J. and SCHUURMANS, D. (1998). Boosting in the limit: Maximizing the margin of learned ensembles. In *Proc. 15th National Conference on Artificial Intelligence* 692–699. AAAI Press, Menlo Park, CA.

[11] JIANG, W. (2000). Does boosting overfit: Views from an exact solution. Technical Report 00-03, Dept. Statistics, Northwestern Univ. Available at neyman.stats.nwu.edu/jiang/boost/boost.onedim.ps.

[12] JIANG, W. (2001). Some theoretical aspects of boosting in the presence of noisy data. In *Proc. 18th International Conference on Machine Learning* 234–241. Morgan Kaufmann, San Francisco. (Also Technical Report 01-01, Dept. Statistics, Northwestern University.) Available at neyman.stats.nwu.edu/jiang/boost/boost.icml.ps.

[13] JIANG, W. (2002). On weak base hypotheses and their implications for boosting regression and classification. *Ann. Statist.* **30** 51–73.

[14] KRIEGER, A., LONG, C. and WYNER, A. (2001). Boosting noisy data. In *Proc. 18th International Conference on Machine Learning* 274–281. Morgan Kaufmann, San Francisco.

[15] MANNOR, S., MEIR, R. and ZHANG, T. (2002). The consistency of greedy algorithms for classification. In *Proc. 15th Annual Conference on Computational Learning Theory. Lecture Notes in Comput. Sci.* **2375** 319–333. Springer, New York.

[16] MASON, L., BAXTER, J., BARTLETT, P. and FREAN, M. (2000). Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers* (A. J. Smola, P. L. Bartlett, B. Schölkopf and A. Schuurmans, eds.) 221–247. MIT Press, Cambridge, MA.

[17] SCHAPIRE, R. E., FREUND, Y., BARTLETT, P. and LEE, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Statist.* **26** 1651–1686.

[18] SCHAPIRE, R. E. and SINGER, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning* **37** 397–336.

[19] YANG, Y. (1999). Minimax nonparametric classification. I. Rates of convergence. *IEEE Trans. Inform. Theory* **45** 2271–2284.

DEPARTMENT OF STATISTICS
NORTHWESTERN UNIVERSITY
EVANSTON, ILLINOIS 60208
USA
E-MAIL: wjiang@northwestern.edu

REJOINDER

BY GÁBOR LUGOSI AND NICOLAS VAYATIS

Pompeu Fabra University, Barcelona and Université Paris 6

We thank the discussants for the interesting comments which shed light on many different aspects of boosting and related methods for classification and regression. In this rejoinder we summarize what we have learned about boosting since the writing of the paper, in great part thanks to these discussion pieces.

The new and elegant proof of the consistency theorem of Koltchinskii is not only amusing but also shows how many seemingly different classifiers, including regularized boosting and support vector machines, can be analyzed in a single framework. The main message of Bartlett, Jordan and McAuliffe is similar in that they consider so-called large-margin classification methods which minimize a certain empirical loss function of the margin different from the empirical probability of error and characterize the loss functions which lead to consistent classification. The generality of these conditions is surprising and again, develops a unified treatment that encompasses not only various versions of boosting methods but also support vector machines and related kernel-based methods.

We agree with Freund and Schapire that consistency is just a minimal requirement and does not explain the good practical behavior of boosting. Once consistency is established, attention should be turned to a finer analysis. Koltchinskii points out the importance of establishing rates of convergence.

However, it is not completely obvious what reasonable assumptions are for the distribution in high-dimensional classification problems. We share the view of Friedman, Hastie, Rosset, Tibshirani and Zhu that sparsity should play a key role. We believe that the analysis of consistency provides valuable insight into the behavior of boosting. Indeed, building partly on the techniques of the discussed papers by Zhang and us, and on the recent paper of Bartlett, Jordan and McAuliffe (cited in their discussion), in a recent joint work with Gilles Blanchard [1] we have been able to derive rate-of-convergence results for regularized boosting methods similar to the ones studied in our paper. As it turns out, some regularized boosting methods produce classifiers whose probability of error converges to the Bayes error at a rate independent of the dimension [faster than $O(n^{-1/4})$ and sometimes as fast as $O(n^{-1/2})$] for large classes of distributions. This is an interesting feature not shared by classical nonparametric methods such as the k-nearest neighbor classifier, as also pointed out by Freund and Schapire. The distributions under which such a rate of convergence holds are those for which the function f^* minimizing the cost function $A(f) = \mathbb{E}\phi(-f(X)Y)$ can be approximated arbitrarily (say, in the L_∞ sense) by linear combinations of base functions with coefficients bounded in L_1. The characterization of these distributions is far from being trivial in general, but in some cases it is well understood. As an example, we cite the following special case from [1]:

COROLLARY 1. *Let $X \in \mathbb{R}^d$ with $d \geq 2$. There exist a regularized boosting classifier $\widehat{f}_n$ based on the logit cost function and decision stumps such that if there exist functions $f_1, \ldots, f_d : \mathbb{R} \to \mathbb{R}$ and a positive constant B such that the sum of the total variations of the f_i is bounded by Bd and such that $\log \frac{\eta(x)}{1-\eta(x)} = \sum_{i=1}^d f_i(x^{(i)})$, then for every n, with probability at least $1 - 1/n^2$, the probability of error $L(\widehat{f}_n)$ satisfies*

$$L(\widehat{f}_n) - L^* \leq C\sqrt{d \log d}\, n^{-(1/(2(2-\alpha)))((V_d+2)/(V_d+1))},$$

where C is a universal constant, $V_d \leq 2\log_2(2d)$ and the value of $\alpha \in [0, 1]$ depends on the distribution.

This result quantifies the observation of Friedman, Hastie and Tibshirani [2] who pointed out a close relationship between boosting and additive logistic regression. The example described by Bühlmann and Yu fits exactly in the framework of this corollary and explains the good behavior of LogitBoost in their simulations. Interestingly, the same result is not true when the exponential cost function is used. In that case, even though the rate of convergence in terms of the sample size remains the same, the dimension-dependent constant in front grows exponentially rapidly with d. It is a remarkable fact that the dimensionality only appears in the multiplicative constant of the rate of convergence. We believe that, even though now we are closer to the understanding of boosting and related

methods, there is still a lot to discover and interesting unexplored questions abound.

Freund and Schapire point out that in very high-dimensional problems boosting may not be computationally feasible if the base class is one of the usual classes (e.g., decision trees with $d + 1$ extremal nodes) which guarantee universal consistency. In such cases one may have to resort to smaller base classes such as decision stumps. The corollary above shows that boosting based on stumps has excellent behavior if the distribution happens to follow an additive logistic model. However, one should proceed with care when using such "incomplete" base classes. It is shown in [1] that boosting (and other large-margin methods which minimize an empirical cost functional) may have catastrophic behavior if the function f^* cannot be approximated by linear combinations of base functions in the sense that the resulting classifier may have a probability of error which is much larger not only than the Bayes error but also than the error of the best classifier realizable by linear combinations of base classifiers. Thus, an interesting open problem is to find "simple" base classes which are dense in the sense that all possible classifiers can be approximated by convex combinations of base classifiers. In a recent manuscript [4] we show the existence of such a class of VC dimension 1, independently of the dimension of the space. While the construction given in that paper is probably of little practical value, a better understanding of the tradeoff between computational complexity and approximation ability is an important challenge.

Another important issue that Freund and Schapire raise is that by minimizing an empirical cost function such as the exponential or the logit functions one implicitly estimates the whole conditional probability function $\eta(x)$ (more precisely, a monotone function of it). By doing that, one does more than necessary since in binary classification the only thing that matters is whether $\eta(x)$ is greater or less than $1/2$. The results of Bartlett, Jordan and McAuliffe refine this point of view by showing that under conditions on the behavior of $\eta(x)$ around $1/2$ (introduced by Tsybakov) the rate of convergence of boosting methods speeds up considerably. [The constant α in the corollary above is determined by the behavior of $\eta(x)$ in the vicinity of $1/2$.] There is one convex cost function, the "hinge loss" used by support vector machines, which has the distinguishing property that its minimizer is the Bayes classifier g^* itself; see Lin [3]. Thus, as opposed to boosting, support vector classifiers do just what they are supposed to do and do not "waste energy" in estimating the function $\eta(x)$ in irrelevant ranges. However, this does not necessarily mean that support vector machines perform better, as for the hinge loss it seems to become more difficult to approximate the minimizer f^* by linear combinations of base classifiers. Once again, the relationship of minimizers of different empirical cost functions is complex, very far from being well understood.

The discussion of Bühlmann and Yu tackles algorithmic issues of regularized boosting procedures. In our experiments we used MarginBoost.L_1 as a convergent algorithm giving a nearly optimal output in the λ-blowup of the convex hull of the base class (for a *fixed* value λ of the smoothing parameter). Running this algorithm for various values of λ revealed that this smoothing parameter was effectively acting as a relevant complexity measure even for small sample sizes. The discussion of Friedman, Hastie, Rosset, Tibshirani and Zhu, pointing out the connection of regularized boosting methods with L_1-penalty to Tibshirani's Lasso, provides strong intuition on how the practical problem of finding efficient greedy algorithms can be dealt with.

Bühlmann and Yu also comment on the importance of distinguishing between regularizing by an explicit constraint on the sum (or other norm) of the weights and by early stopping. This is an important and difficult question. The very interesting results of Bickel and Ritov show in a general framework that stopping by cross validation works in a strong sense. While early stopping is alluring from a practical point of view (it reduces to AdaBoost, plus a stopping rule), its theoretical analysis is more problematic. Indeed, in most cases, it turns out that there is an optimal value for the smoothing parameter $\lambda = \lambda^*$ (corresponding to the L_1-norm of the weights of the optimal combination). The successive iterations in AdaBoost can be conceived as drawing a path in the space of the weights crossing the iso-surfaces defined by constant values of the L_1-norm of the weights, and early stopping returns an output on this path which may be close to the optimal vector of weights. Since there is no known guarantee that during the iterations the weight vector passes through a near-optimal value for the best choice λ^*, it seems to be difficult to derive rate-of-convergence results such as the corollary above for AdaBoost or LogitBoost with early stopping. To better understand the relationship between explicitly regularized boosting and early stopped AdaBoost is a challenging problem that requires careful study of the approximation properties of the iterative construction of the boosting estimator based on highly redundant dictionaries of base classifiers. We entirely agree with Friedman, Hastie, Rosset, Tibshirani and Zhu, who point out the importance of sparsity. We believe that these perspectives motivate interesting research at the interface of statistics, optimization and approximation theories.

REFERENCES

[1] BLANCHARD, G., LUGOSI, G. and VAYATIS, N. (2003). On the rate of convergence of regularized boosting classifiers. *J. Mach. Learn. Res.* **4** 861–894.

[2] FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion). *Ann. Statist.* **28** 337–407.

[3] LIN, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Min. Knowl. Discov.* **6** 259–275.

[4] LUGOSI, G., MENDELSON, S. and KOLTCHINSKII, V. (2003). A note on the richness of convex hulls of VC classes. *Electron. Comm. Probab.* **8** 167–169.

DEPARTMENT OF ECONOMICS
POMPEU FABRA UNIVERSITY
RAMON TRIAS FARGAS 25-27
08005 BARCELONA
SPAIN
E-MAIL: gabor.lugosi@econ.upf.es

LABORATOIRE DE PROBABILITÉS
ET MODÉLES ALÉATOIRES
UNIVERSITÉ PARIS 6
4 PLACE JUSSIEU
BOÎTE COURRIER 188
75252 PARIS CEDEX 05
FRANCE
E-MAIL: vayatis@ccr.jussieu.fr

REJOINDER

BY TONG ZHANG

IBM T. J. Watson Research Center

The discussants contributed different views on several aspects of large margin classification methods and outlined some interesting future directions. I would like to thank them for the stimulating comments. In the following I will mainly focus on two issues. One is the conditional probability modeling aspect of large margin classification methods and the other is related to properties of greedy algorithms used in boosting procedures.

1. Consistency and conditional probability model. The basic technical ideas used in all the papers for proving consistency were summarized both in the Bartlett, Jordan and McAuliffe discussion and in Koltchinskii's discussion. Koltchinskii presented a simplified proof of consistency, with a flavor similar to that of [7]. Bartlett, Jordan and McAuliffe divided the consistency proof into three main components and focused on "comparison results" that relate the classification error of a classifier that approximately minimizes a loss function to the Bayes error. In particular, they presented some nice results from their recent paper [1] that are clean extensions to related bounds in my paper.

An interesting issue regarding such comparison results is that the risk of a decision function may converge to the optimal value at a different rate than its corresponding classification error does. This difference of convergence speed is reflected by the parameter s in Theorem 2.1 of my paper. The value of s only depends on local properties of the loss function ϕ around zero. In particular, if ϕ is linear around zero (such as SVM), then we may take $s = 1$, which is the best possible value. The property is also related to the fact that SVM determines the sign of the conditional probability minus 0.5 instead of estimating the conditional probability itself. For loss functions that decrease sublinearly around zero, we will have $s > 1$ (and usually $s = 2$).

This suggests that the SVM loss may have an advantage over some other convex loss functions when the corresponding risks converge to the optimal values at

comparable rates. This may happen in certain cases (though not always). In fact, since SVMs directly model the sign of the conditional probability minus 0.5, using the SVM loss with a function class that includes the Bayes classification rule is similar to direct classification error minimization. As argued by Freund and Schapire in their discussion, this may be a better method for binary classification because we do not have to obtain a good estimation of conditional probability in order to perform classification. This opinion is certainly valid and a similar argument was made in [2], where the authors showed that for some problems, binary classification is easier than conditional probability estimation. It should be interesting to note that for many nonparametric models, classification can be as hard as regression, at least when measured by the minimax rate of convergence [9]. However, this does not mean that it will not be beneficial to use a classification error minimization based method for such problems. In fact, my opinion is that if binary classification is our goal, then in most cases it will be better to directly minimize the classification error with an appropriately chosen function class than to use convex risk minimization. In practice, however, due to computational difficulties, direct classification error minimization is often not possible. It is also unclear whether there is still an advantage of performing some variants of direct classification minimization for multi-class classification problems.

The main advantage of convex risk minimization is the ability to provide conditional probability information. In many, if not all applications, such information will be useful. If an application relies on conditional probability estimation, then it is necessary to understand the implication of using different convex loss functions. We shall provide two real-world pattern recognition examples the author worked on recently to illustrate this point.

2. Some empirical examples on real-world datasets. The first example is taken from [10], where the problem of building recommender systems was considered. A recommender system uses historical data on user preferences and other available data on users (e.g., demographics) and items (e.g., taxonomy) to predict items a new user might like. Such systems have been used widely in electronic commerce. For example, Amazon.com recommends new books to a user based on the user's historic buying pattern. Commonly used systems are often based on variations of the nearest-neighbor method.

In general this problem can be posed as a classification problem: one wants to predict how likely a user is interested in each item, based on historic data. The system can then recommend those items that (it believes) a user is most likely to buy to that particular user. We illustrated in [10] (as well as in some IBM internal experiments performed with product groups) that convex risk minimization leads to better prediction accuracy than many existing alternatives. Without a correct understanding of the role of loss functions, one may tend to think that the SVM method can be a good choice due to its success in other related classification tasks. However, it performs poorly here because it does not estimate conditional

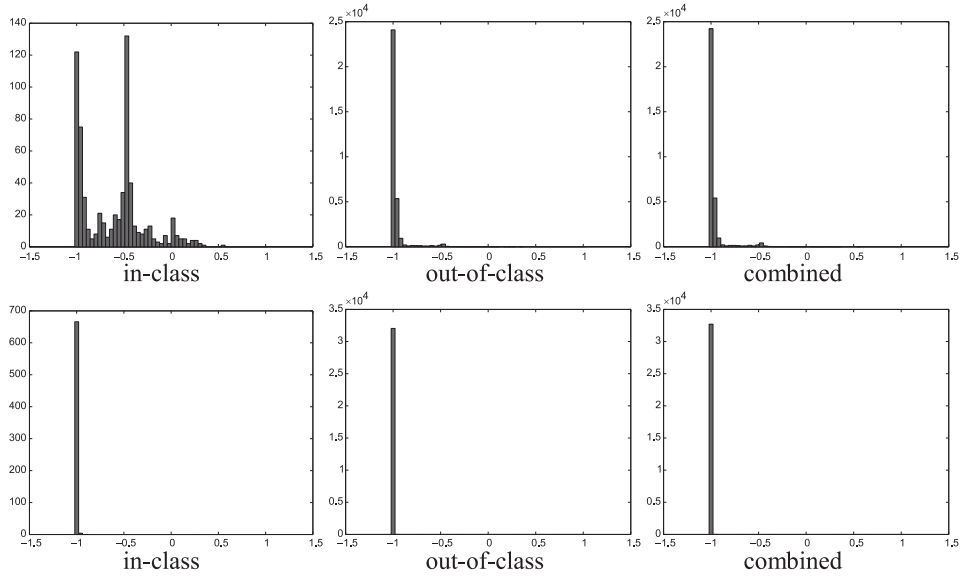

FIG. 1. *Histograms of prediction outputs for recommending an item with* 2% *in-class probability*: *ridge regression* (top row) *versus SVM* (bottom row).

probability, as required for this application. In fact, ridge regression with the least squares loss function performs significantly better due to its ability to provide conditional probability estimation.

Figure 1 shows the histogram of prediction outputs for all users with classifiers trained with the least squares loss and the SVM loss, respectively, on a specific item that has a 2% in-class population. We plot the histograms separately for users that bought the item (in-class), users that did not buy the item (out-class), and the previous two groups combined. We note that the SVM classifier outputs -1 for all users since it only predicts the sign of the conditional probability compared with 0.5. The least squares method, although unable to significantly separate in-class users from the out-of-class users for the purpose of good classification, is able to provide sufficiently useful conditional probability information for the purpose of ranking items to be recommended to the users. It will also be interesting to mention that logistic regression does not do well for this application. Although in theory logistic regression can be used to estimate conditional probability, as mentioned in my paper, it has difficulty modeling a conditional probability $P(Y = 1|X) \approx 0$, which is required for this application.

The above example shows that it is important to understand the behavior of different loss functions for practical applications. Useful intuition can be obtained

by studying the optimal minimizer f_ϕ^*. For example, Lin has also observed that one may take

$$f_\phi^*(\eta) = \text{sign}(2\eta - 1)$$

for SVM in [6], and this observation was later used in [5] to design very interesting multi-category support vector machines that are Bayes consistent.

One can also see that f_ϕ^* does not fully characterize the behavior of a loss function ϕ. For example, exponential loss and logistic regression loss share the same form of f_ϕ^*; SVM loss and L_1-regression share the same f_ϕ^*; and least squares, modified least squares and modified Huber losses share the same f_ϕ^*. What really differentiates these loss functions are the induced distance functions. In particular, using their corresponding distance functions, we can see that the modified Huber's loss leads to a more robust conditional probability model than that of the least squares loss.

Although the difference between the least squares loss and the modified Huber's loss may not always be observable, it can become important in certain applications. For example, the difference is significant in some natural language processing problems that we have worked on. One such problem is the named entity recognition task, which is to find people names, organizations and locations (plus some other possible entities) from electronic text documents. For example, we want to annotate the following sentence as: *Only* [*LOCATION France*] *and* [*LOCATION Britain*] *backed* [*PERSON Fischler*]*'s proposal.*

This problem can be modeled as a multi-class classification problem based on conditional probability estimation. Convex risk minimization can then be used to obtain such conditional probability models. In fact by using modified Huber risk minimization, we achieved the best performance in a recent benchmark contest among seventeen participating systems [4]. The performance of a named entity recognition system is usually measured by the so-called F-measure, which is two times the number of correctly predicted entities divided by the sum of the number of predicted entities and the number of true entities. For the above mentioned benchmark experiment, the F-measure of a certain modified Huber loss minimization based system on the English test set is 85.5, which can be compared with an F-measure of 81.6 using least squares loss minimization (under exactly the same configuration otherwise). The histograms of classifier outputs corresponding to the predicted class with these two different loss functions are presented in Figure 2. From the plots, we observe that the class of most instances (typically words that are not entities) can be predicted very accurately, with estimated probability at about 1. An L_2 minimization based model requires the prediction outputs to be concentrated around 1. A modified Huber loss based estimator can model probability 1 with prediction outputs larger than 1. This is an important advantage that makes a difference for this application.

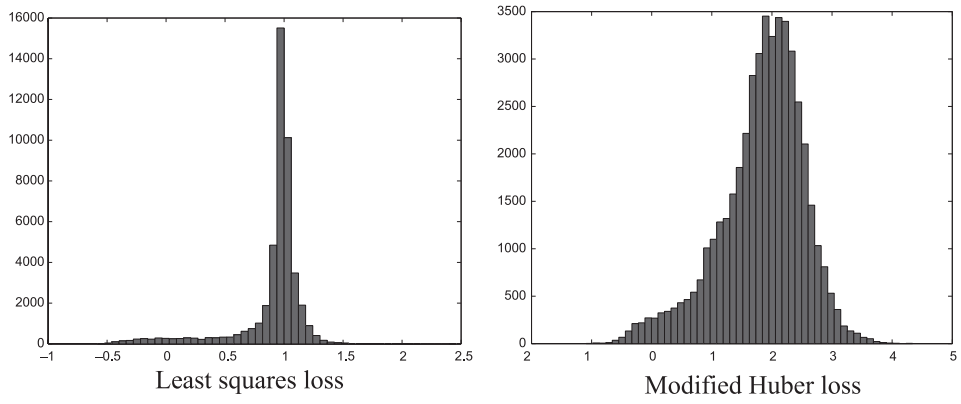

FIG. 2. *Histograms of prediction outputs for named entity recognition systems using the least squares loss and the modified Huber's loss.*

3. Greedy algorithm and sparsity. The greedy search aspect of boosting algorithms has been discussed by Bickel and Ritov, Bühlmann and Yu, and Friedman, Hastie, Rosset and Tibshirani and Zhu. A number of interesting questions and research directions were laid out in Bickel and Ritov's discussion. Friedman, Hastie, Rosset and Tibshirani and Zhu argued that the success of greedy boosting is due to the fact that it approximately solves an L_1 regularized problem (Lasso).

Although there is a close relationship between greedy boosting and L_1 regularization, which has been known in the boosting community, the two approaches are not equivalent. This point has been made in Bühlmann and Yu's discussion. The special case used by Friedman, Hastie, Rosset and Tibshirani and Zhu to illustrate the equivalence of L_1 regularization and greedy boosting is certainly very interesting, but it requires assumptions that can be restrictive for practical problems. In fact, under appropriate conditions, the equivalence will also follow from the general convergence analysis presented in [11]. In more realistic situations, there is still a strong connection between greedy boosting and L_1 regularization [11], but they are not identical.

Another interesting issue is to achieve sparse representation using either greedy boosting or L_1 regularization. I would like to argue that these two methods behave differently for this purpose. It is known that L_1 regularization can lead to sparse solutions, although it is not equivalent to sparse (L_0) regularization. The L_0 regularization of a weight vector $w = [w_i]$ is defined as $\lim_{\rho \to 0} \sum_i |w_i|^\rho$, which is the number of nonzero components in w. If achieving sparsity is our goal, both greedy approximation and L_1 regularization can be regarded as approximations to the true L_0 regularization. One method can work better than the other for achieving sparsity depending on properties of the underlying problem. Greedy boosting is a more direct method of achieving sparsity since it gives a linear combination of k-vectors after k greedy-updating steps.

Although for some special problems, L_1 regularization can actually be equivalent to L_0 regularization (e.g., see [3]), for real applications L_1 regularization often does not lead to a solution with near optimal sparsity. A main practical issue with L_1 regularization is that it does not produce a unique (or numerically stable) solution for basis functions that are identical (or similar) since one can get the same (or similar) system value by varying the corresponding weight components and keeping their sum identical. Therefore L_1 regularization usually does not produce a sparse solution in the case of similar basis functions, and the inherent instability is often undesirable as far as numerical computation is concerned. In practice one often stabilizes the system by introducing a small amount of quadratic regularization. The resulting system is strictly convex, and thus always has a unique solution. However, the stability of the solution also implies that basis functions that are identical (or similar) will have identical (or similar) weight components. This is because if the system is invariant under an interchange of two parameter components, then at the solution the two parameter components have to be equal (otherwise, there will be at least two solutions). The problem is also a consequence of Jensen's inequality, which tends to favor equal weight values when we use convex regularization. Therefore this problem can only be addressed with nonconvex regularization conditions, although they are computationally less desirable. As a comparison, greedy algorithms do not have this particular problem since such procedures are inherently nonconvex.

Another way to understand that L_1 regularization generally fails to produce solutions with sufficient sparsity is that the complexity of an L_1 regularized space is usually quite different from that of an L_0 regularized space. For example, if a function space C has a uniform L_2-covering number that is polynomial in the approximation scale (such as single-layer neural networks or fixed level decision trees), then the function class that is sparsely representable with a fixed number of components (L_0 regularized space) also has a polynomial dependency on the approximation scale. However, the uniform L_2-covering number of the convex hull (L_1 regularized space) is typically exponential in the approximation scale and can approach the upper bound presented in Section 2.6.3 of [8].

Since the minimax-rate of function estimation is usually determined by the dependency of a model family's L_2-covering number on the approximation scale, we know that the L_1 regularized space can be much more complex than the L_0 regularized space. Therefore if the function to be approximated is sparsely representable by the basis functions, then we know that a direct L_1 regularization alone will not always lead to a good estimation method. However in this case, greedy boosting may still succeed, assuming such a procedure can find a good sparse representation in a small number of iterations.

REFERENCES

[1] BARTLETT, P. L., JORDAN, M. I. and MCAULIFFE, J. D. (2003). Convexity, classification, and risk bounds. Technical Report 638, Dept. Statistics, California Univ., Berkeley.

[2] DEVROYE, L., GYÖRFI, L. and LUGOSI, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

[3] DONOHO, D. and HUO, X. (2001). Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inform. Theory* **47** 2845–2862.

[4] FLORIAN, R., ITTYCHERIAH, A., JING, H. and ZHANG, T. (2003). Named entity recognition through classifier combination. In *Proc. CoNLL-2003* 168–171. Morgan Kaufmann, San Francisco.

[5] LEE, Y., LIN, Y. and WAHBA, G. (2004). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *J. Amer. Statist. Assoc.* To appear.

[6] LIN, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Min. Knowl. Discov.* **6** 259–275.

[7] MANNOR, S., MEIR, R. and ZHANG, T. (2002). The consistency of greedy algorithms for classification. In *Proc. 15th Annual Conference on Computational Learning Theory. Lecture Notes in Comput. Sci.* **2375** 319–333. Springer, New York.

[8] VAN DER VAART, A. W. and WELLNER, J. A. (1996). *Weak Convergence and Empirical Processes*. Springer, New York.

[9] YANG, Y. (1999). Minimax nonparametric classification. I. Rates of convergence. *IEEE Trans. Inform. Theory* **45** 2271–2284.

[10] ZHANG, T. and IYENGAR, V. S. (2002). Recommender systems using linear classifiers. *J. Mach. Learn. Res.* **2** 313–334.

[11] ZHANG, T. and YU, B. (2003). Boosting with early stopping: Convergence and consistency. Technical Report 635, Dept. Statistics, Univ. California, Berkeley. Available from www.stat.berkeley.edu/~binyu/publications.html.

IBM T. J. WATSON RESEARCH CENTER
YORKTOWN HEIGHTS, NEW YORK 10598
USA
E-MAIL: tzhang@watson.ibm.com