The Annals of Statistics
2004, Vol. 32, No. 1, 1–11

THE 2002 WALD MEMORIAL LECTURES

POPULATION THEORY FOR BOOSTING ENSEMBLES

BY LEO BREIMAN

University of California, Berkeley

Tree ensembles are looked at in distribution space, that is, the limit case of "infinite" sample size. It is shown that the simplest kind of trees is complete in D-dimensional $L_2(P)$ space if the number of terminal nodes T is greater than D. For such trees we show that the AdaBoost algorithm gives an ensemble converging to the Bayes risk.

1. Introduction. Using ensembles of predictors for classification and regression has proved to give more accurate results than use of a single predictor. The most well-known ensemble method is the AdaBoost algorithm, introduced by Freund and Schapire [12]. Complex in form, it was originally designed to drive the training set error rapidly to zero. But, surprisingly, empirical experiments showed that the generalization error kept decreasing long after the training set error was zero—in fact, slowly decreasing as hundreds of trees were added to the ensemble.

After its introduction, AdaBoost became known as the most accurate general purpose classification algorithm available. (For performance benchmarks, see [1, 8, 9].) Hundreds of papers covering various modifications of AdaBoost or applications of AdaBoost were and are being published in the Machine Learning literature. But the understanding of how and why AdaBoost worked was lacking until a short time ago. During the intervening years I often referred, in prepared talks and conversations, to the mystery of AdaBoost as the most important unsolved problem in Machine Learning.

Schapire, Freund, Bartlett and Lee [17] constructed a bound on the generalization error of ensembles in terms of the training set margin distribution and the VC-dimension of the individual predictors in the ensemble. This led to the conjecture that the key to the low generalization error of the AdaBoost algorithm was that it produced a high margin distribution. But Breiman [5] constructed an algorithm that on a variety of data sets produced uniformly higher margin distributions than AdaBoost using predictors with the same VC-dimension, yet had higher generalization error.

A significant discovery, due to Breiman [3] (and rediscovered several times afterward) was that AdaBoost was a down-the-gradient method for minimizing an exponential function of the error (see Section 3). Then "boosting" became a generic term for greedy down-the-gradient minimization of various loss functions.

Received August 2002; revised May 2003.
AMS 2000 subject classifications. 62H30, 68T10, 68T05.
Key words and phrases. Trees, AdaBoost, Bayes risk.

This context allowed more focused research on how AdaBoost works. An outstanding question has been whether AdaBoost is Bayes-consistent. The author believed for some years that it is. The empirical evidence seemed supportive, but recent work proved otherwise.

The present article first appeared as a technical report [6]. Even though it assumed "infinite sample size" it was suggestive that in this setting the AdaBoost algorithm produced an ensemble with risk converging to the Bayes risk. However, it also indicated that the population case may differ intrinisically from the finite sample case. In the population case, the sum of the squares of the coefficients of each new tree added to the ensemble converges and this result is vital to the proof (see the Appendix). In the finite sample case, the sum of the squares of the coefficients diverges no matter how large the sample size.

Intense recent work in the finite sample case, partly stimulated by the technical report containing these population results, has largely resolved the situation. AdaBoost is not Bayes-consistent. It is Bayes-consistent only if it is regularized. Some of these clarifying and elegant results are contained in work by Jiang [14], who regularizes by early stopping, Lugosi and Vayatis [15], who smoothed the loss function and restricted the sum of the coefficients, Mannor, Meir and Zhang [16], who assume smooth Bayes region boundaries, and Zhang and Yu [20], who study early stopping. There has also been illuminating theoretical work in the case of L_2 loss [7].

We now have a better idea of how AdaBoost works. In its early stages, while performing the gradient descent, it mimics the population version getting close to the Bayes risk. Then, when it cannot get convergence, it "gives up" and goes into a second phase of increasing generalization error. What it is doing in this second phase is so far unknown.

An interesting observation is that although AdaBoost and its many subsequent applications and generalizations were developed in the Machine Learning community, the work on its theory and on the Bayes-consistency has been largely carried out by statisticians. The theoretical analysis and understanding of algorithms that are important in practice is a gratifying development in statistics.

2. Theoretical assumptions and outline of results.

2.1. *Framework.* We work in population space. The outputs and inputs are represented by two random vectors Y, $\mathbf{X}$ whose distributions are known and $\mathbf{X}$ is assumed distributed on a finite closed D-dimensional Euclidean rectangle, $\mathbb{R}^D$. The distribution of $\mathbf{X}$ is given by $P(d\mathbf{x})$ which is assumed to be absolutely continuous w.r. to Lebesgue measure on D dimensions so that $P(d\mathbf{x}) = f(\mathbf{x})\,d\mathbf{x}$. The distribution of Y is given by $P(y|\mathbf{x})$. Both distributions are assumed known.

Only the two-class situation is considered. The members of the ensemble will be trees, all with the same number T of terminal nodes, formed by cuts parallel to the axes (although it will become clear how this generalizes). Also, they will

be ± 1 trees. This means that to each terminal node is assigned the value either $+1$ or -1. Trees with arbitrary values attached to their terminal nodes could be used, but in our context they offer no advantage over ± 1 trees.

2.2. *Nontechnical outline of results.* In Section 3 we define a set of classifiers ζ to be complete on $\mathbb{R}^D$ if every real-valued function on $\mathbb{R}^D$ is equal to a linear combination (perhaps infinite) of classifiers in ζ. We show that the class of ± 1 trees with T terminal nodes is complete if $T > D$. A simple example shows that stumps, with $T = 2$, are not complete if $D = 2$.

The implication is that a linear combination of trees with $T > D$ terminal nodes can achieve the minimal error rate (Bayes risk). The question that rises is how to compute the coefficients of a combination that will give minimum error. The fundamental down-the-gradient algorithm is defined in Section 4. This gives a method for evaluating coefficients that minimize any target function. In Section 5 the minimization algorithm applied to an exponential function (AdaBoost analog) is shown to converge to the minimum possible error (Bayes rate).

3. Completeness. First some familiar Hilbert space definitions and properties are given (see, e.g., [10]).

DEFINITION 1. Let $L_2(P)$ be the space of functions on $\mathbb{R}^D$ that are square-integrable with respect to $P(d\mathbf{x})$. A set of functions F in $L_2(P)$ will be called complete if the L_2 closure of the set of all finite linear combinations of functions in F, denoted by $\bar{c}(F)$, equals $L_2(P)$.

PROPERTY 1. A set of functions F is complete if and only if there is no nonzero function g in $L_2(P)$ such that $(g, f) = 0$ for all $f \in F$.

Note: $(f, g) = \int f(\mathbf{x})g(\mathbf{x})P(d\mathbf{x})$.

PROPOSITION 1. *A set of functions F is complete if $\bar{c}(F)$ includes the indicators of all D-dimensional subrectangles of $\mathbb{R}^D$.*

PROOF. The proof is well known. □

3.1. ± 1 *trees.* The ± 1 trees are that set of trees most commonly used in two-class classification.

DEFINITION 2. $h(\mathbf{x})$ is a ± 1 tree if the T terminal nodes are formed by successive univariate splits of the input variables such that for all $\mathbf{x}$ in a given terminal node, $h(\mathbf{x})$ is always $+1$ or always -1.

PROPOSITION 2. *If $T > D$ the class of ± 1 trees is complete in $L_2(P)$.*

PROOF. We will show that the indicator of any rectangle can be gotten as a finite linear combination of ± 1 trees and apply Proposition 1. To form the indicator function of the rectangle

$$R_0 = \{r_1 < x_1 \le s_1, r_2 < x_2 < s_2, \ldots, r_D < x_D < s_D\},$$

proceed as follows: Trees with T terminal nodes use $T-1$ splits. If $T > D$, the splits can be used to make a tree Tr_1 which contains the rectangle

$$R_1 = \{x_1 < s_1, x_2 < s_2, \ldots, x_D \le s_D\}$$

as one of its terminal nodes with the value $+1$. Let Tr_2 be a tree with exactly the same terminal nodes as the first but with the value of each node reversed except for R_1. Then $0.5 * \mathrm{Tr}_1 + 0.5 * \mathrm{Tr}_2$ is the indicator of R_1. Repeating the above process, construct the indicator function of

$$R_2 = \{x_1 < r_1, x_2 < s_2, \ldots, x_D \le s_D\}.$$

Subtracting this from the indicator of R_1 gives the indicator of

$$R_3 = \{s_1 < x_1 < r_1, x_2 < s_2, \ldots, x_D \le s_D\}$$

and continuing this process leads to the indicator of R_0.

Proposition 2 is probably if and only if. For instance, the class of stumps (two terminal node trees) is not complete in $D = 2$. Take P uniformly distributed on the square $(-1, +1)^2$. Take $g(\mathbf{x})$ to equal $+1$ in the first and third quadrants, and -1 in the second and fourth. Then $(h, g) = 0$ for every two-node ± 1 tree $\mathbf{h}(\mathbf{x})$. $\square$

3.2. *Implications for predicting with ensembles.* The implications of completeness in terms of using ensembles for prediction are encouraging. Suppose that the loss function $L(Y, \phi(\mathbf{X}))$ is a measure of the error in using $\phi(\mathbf{X})$ to predict Y. The expected loss is

$$L^*(\phi) = E_{Y,\mathbf{X}} L^2\big(Y, \phi(\mathbf{X})\big),$$

where the subscripts indicate expectation with respect to $Y, \mathbf{X}$. Assume that functions exist in $L_2(P)$ that minimize $L^*(\phi)$. Then:

THEOREM 1. *In $\mathbb{R}^D$ there exist linear combinations of ± 1 trees with $T > D$ that converge in $L_2(P)$ to any minimizer of $L^*(\phi)$ that is in $L_2(P)$.*

We illustrate with classification. Here

$$L^*(\phi) = P_{Y,\mathbf{X}}\big(Y \ne \phi(\mathbf{X})\big).$$

To put this into a more familiar ensemble context, assume that $\phi(\mathbf{x})$ predicts $y = 1$ if $\phi(\mathbf{x}) > 0$, else predicts -1. Then

$$L^*(\phi) = P_{Y,\mathbf{X}}\big(Y \ne \mathrm{sign}(\phi(\mathbf{X}))\big).$$

Let $P(i|\mathbf{x}) = P(Y = i|\mathbf{x})$. Then

$$L^*(\phi) = \int I\big(\phi(\mathbf{x}) \le 0\big) P(1|\mathbf{x}) P(d\mathbf{x}) + \int I\big(\phi(\mathbf{x}) > 0\big) P(-1|\mathbf{x}) P(d\mathbf{x}),$$

where I is the indicator function. Take finite combinations of ± 1 trees

$$\sum_m c_m h_m(\mathbf{x})$$

that converage to a nonpositive function on the set $P(1|\mathbf{x}) < P(-1|\mathbf{x})$ and a positive function on the complement of the set. Then the loss converges to

$$\int \min\big(P(1|\mathbf{x}), P(-1|\mathbf{x})\big) P(d\mathbf{x}),$$

which is the Bayes risk.

In the following sections we adopt the convention that "all trees" refers to all ± 1 trees with T terminal nodes where $T > D$.

4. A constructive algorithm. The results in Section 3 are nonconstructive. They assert existence, but give no idea as to how to construct linear combinations of trees that converge to a desired function. There is an algorithm, first introduced in [3], further elaborated in [5] (and often rediscovered), that offers a constructive method for producing such sequences. The finite-dimensional numerical optimization version of this algorithm has been around for a while and is called the Gauss–Southwell method (see [11]).

The Gauss–Southwell method for minimizing a differentiable function $f(x_1, \ldots, x_m)$ of m real variables goes this way: at a point $\mathbf{x}$ compute all the partial derivatives $\partial f(x_1, \ldots, x_m)/\partial x_k$. Let the minimum of these be at x_j. Find the step of size α that minimizes $f(x_1, \ldots, x_j + \alpha, \ldots, x_m)$. Let the new $\mathbf{x}$ be $x_1, \ldots, x_j + \alpha, \ldots, x_m$ for the minimizing α value. If f is strictly convex on its domain, this algorithm converges to the global minimum.

The analog "boosting" algorithm produces a sequence of linear combinations of trees that minimize (under appropriate conditions) a given real-valued target function of the sequence. Just as the Gauss–Southwell method does, it finds the largest negative gradient at each $\mathbf{x}$ and then does a line minimization, but there are an infinite number of gradient directions.

4.1. *Minimization algorithm.* To find the coefficients of a sum of trees that minimizes

$$E_{\mathbf{X}} \theta\left(\sum_1^\infty c_m h_m(\mathbf{x}), \mathbf{x}\right),$$

where $\theta(s, \mathbf{x})$ is real-valued, continuous and differentiable in s for each value of $\mathbf{x}$, proceed as follows: After M steps:

(i) Compute the minimum over all trees $h(\mathbf{x})$ of

$$E_{\mathbf{X}}\left[\theta_s\left(\sum_1^M c_m h_m(\mathbf{x}), \mathbf{x}\right)h(\mathbf{x})\right],$$

where θ_s is the partial with respect to s. Denote a minimizing tree by $h_{M+1}(\mathbf{x})$.

(ii) Find the α that minimizes

$$E_{\mathbf{X}}\theta\left(\sum_1^M c_m h_m(\mathbf{x}) + \alpha h_{M+1}(\mathbf{x}), \mathbf{x}\right)$$

and let c_{M+1} equal the minimizing value of α.

(iii) Repeat until convergence.

Regarding (i), it is simple to show that there exist minimizing trees (see the Appendix). Uniqueness is difficult and may not hold.

As pointed out in [3], this algorithm can be implemented with finite data and reweighting of the training set.

4.2. *Heuristic proof of convergence.* A sketch of a convergence proof is given with strong assumptions. When the algorithm is applied in the next section to AdaBoost, the assumptions will be partly lifted.

Assume the following:

(i) The domain of $\mathbf{x}$ is a closed finite rectangle.

(ii) The functional $E_{\mathbf{X}}\theta(f(\mathbf{x}), \mathbf{x})$ is strictly convex on $L_2(P)$.

(iii) The sequence of functions $s_M(\mathbf{x}) = \sum_1^M c_m h_m(\mathbf{x})$ is sequentially compact in $L_2(P)$, that is, every subsequence contains a convergent subsequence.

(iv) The function $\theta(s, \mathbf{x})$ and its first and second derivatives with respect to s are uniformly bounded.

From (ii) there is a unique function $s(\mathbf{x})$ minimizing $E_{\mathbf{X}}\theta(s(\mathbf{x}), \mathbf{x})$.

THEOREM 2. *Under the above assumptions, the sequence $s_M(\mathbf{x})$ converges in $L_2(P)$ norm to $s(\mathbf{x})$.*

PROOF. Suppose that the minimum value over all trees h of

$$E_{\mathbf{X}}[\theta_s(s_M, \mathbf{x})h(\mathbf{x})] \tag{1}$$

is zero. Then the maximum value is also zero since the trees can be reversed. Since linear combinations of trees are dense in $L_2(P)$, then

$$E_{\mathbf{X}}[\theta_s(s_M, \mathbf{x})f(\mathbf{x})] = 0$$

for all functions f in $L_2(P)$. This implies that for small ε and any f

$$E_{\mathbf{X}}\big[\theta\big(s_M + \varepsilon f(x), \mathbf{x}\big)\big] \approx 0.$$

These are the first-order necessary conditions for a minimum. The second-order conditions are automatically satisfied by the convexity. Thus, s_M equals the minimizing function s.

If the iterations do not stop, let h_{M+1} be the minimizing tree at step $M+1$, and $-b_{M+1}$ the minimum value of (1). Now

$$E_{\mathbf{X}}[\theta(s_M + \alpha h_{M+1}, \mathbf{x})] \le \theta_0 - \alpha b_{M+1} + B\alpha^2, \tag{2}$$

where B is the upper bound of the second partial of θ and $\theta_0 = E_{\mathbf{X}}\theta(s_M, \mathbf{x})$. From (2) it follows that the decrease in $E_{\mathbf{X}}\theta(s_M, \mathbf{x})$ given by the $(M+1)$st step is greater than $(b_{M+1})^2/2B$. Since the sum of the decreases must be finite, $b_M \to 0$.

Take a subsequence s_m that converges in 2-norm to a function g [assumption (iii)]. For any tree h,

$$E_{\mathbf{X}}[\theta_s(s_m, \mathbf{x})h(\mathbf{x})] \to 0.$$

At the same time

$$E_{\mathbf{X}}[\theta_s(s_m, \mathbf{x})h(\mathbf{x})] \to E_{\mathbf{X}}[\theta_s(g, \mathbf{x})h(\mathbf{x})].$$

This gives the necessary conditions for the minimum and implies that $g = s$. Therefore, the entire sequence s_M converges in norm to s. (Note that for a convex function, second-order conditions are automatically satisfied.) $\square$

However, the hard thing is showing the sequential compactness of the $\{s_M\}$ sequence—a difficulty we have assumed away. The assumptions for Theorem 2 can be weakened with more technical arguments.

5. AdaBoost converges to the Bayes risk. For a sequence of trees $\{h_m\}$ define a voting function as

$$\phi(\mathbf{x}) = \sum_m c_m h_m(\mathbf{x})$$

and vote for class 1 if $\phi > 0$, else for class -1. As noted in Section 3, the expected error for ϕ is

$$L^*(\phi) = \int I\big(\phi(\mathbf{x}) \le 0\big) P(1|\mathbf{x}) P(d\mathbf{x}) + \int I\big(\phi(\mathbf{x}) > 0\big) P(-1|\mathbf{x}) P(d\mathbf{x}).$$

This is not a pleasant functional to try and minimize. Using the inequalities

$$I(\phi \le 0) \le \exp(-\phi), \qquad I(\phi > 0) \le \exp(\phi) \tag{3}$$

gives $L^*(\phi) \le L(\phi)$, where

$$L(\phi) = \int \exp\big(-\phi(\mathbf{x})\big) P(1|\mathbf{x}) P(d\mathbf{x}) + \int \exp(\phi(\mathbf{x})) P(-1|\mathbf{x}) P(d\mathbf{x}). \tag{4}$$

This functional has a more workable form and is the functional that AdaBoost minimizes using the algorithm given in Section 4.

The following two assumptions are made:

(i) The domain of $\mathbf{x}$ is a closed finite D-dimensional rectangle.
(ii) The function $\log(p(-1|\mathbf{x})/p(1|\mathbf{x}))$ is continuous on this rectangle.

Let

$$r(\mathbf{x}) = \log\bigl(p(-1|\mathbf{x})/p(1|\mathbf{x})\bigr)/2.$$

Then (4) can be rewritten as

$$L(\phi) = \int \cosh\bigl(\phi(\mathbf{x}) + r(\mathbf{x})\bigr)\bigl[\sqrt{p(1|\mathbf{x})p(-1|\mathbf{x})}\,\bigr]P(d\mathbf{x}).$$

Let the probability Q be defined by

$$Q(d\mathbf{x}) = \bigl[\sqrt{p(1|\mathbf{x})p(-1|\mathbf{x})}\,\bigr]P(d\mathbf{x})\Big/\int\bigl[\sqrt{p(1|\mathbf{x})p(-1|\mathbf{x})}\,\bigr]P(d\mathbf{x}).$$

Up to a constant factor, then

$$L(\phi) = \int \cosh\bigl(\phi(\mathbf{x}) + r(\mathbf{x})\bigr)Q(d\mathbf{x}). \tag{5}$$

Note that P and Q differ only by multiplication by a bounded positive function.

Apply the minimization algorithm to (5) building up a sum of trees for ϕ,

$$s_M(\mathbf{x}) = \sum_1^M c_m h_m(\mathbf{x}).$$

THEOREM 3. *$s_M(\mathbf{x})$ converges in $L_2(P)$ norm to $-r(\mathbf{x})$.*

The proof is not difficult but has various details. It is deferred to the Appendix.

PROPOSITION 3. *The error in using $-r(\mathbf{x})$ as a decision function is the Bayes risk.*

PROOF. Look at the loss

$$L(\phi) = \int I\bigl(-r(\mathbf{x}) \le 0\bigr)P(1|\mathbf{x})P(d\mathbf{x}) + \int I\bigl(-r(\mathbf{x}) > 0\bigr)P(-1|\mathbf{x})P(d\mathbf{x}).$$

A simple computation shows that

$$L(\phi) = \int \min\bigl(P(-1|\mathbf{x}),\, P(1|\mathbf{x})\bigr)P(d\mathbf{x}),$$

which is the Bayes risk. □

APPENDIX

Minimizing trees exist. Let $f(\mathbf{x})$ be in $L_2(P)$. Denote by H the set of all ± 1 trees with $T+1$ terminal nodes and let

$$\alpha = \inf_{h \in H} (f, h).$$

We show that there is a tree h^* in H such that $\alpha = (f, h^*)$. Take h_n a sequence of trees in H such that $(f, h_n) \to \alpha$.

The skeleton (h) of a tree is a set of integers $(i_1, i_2, \ldots)$ that describes the tree's architecture and the labeling by $+1$ or -1 of its terminal nodes. For instance, i_1 is the number of the variable used in the first split. Over H, skeleton (h) takes on only a finite number of values. So we can select a subsequence $\{h_{n'}\}$ of the $\{h_n\}$ such that all $h_{n'}$ have the same skeleton.

For this subsequence, let $s^{(n')} = (s_1^{(n')}, s_2^{(n')}, \ldots, s_T^{(n')})$ be the set of T split points for the tree $h_{n'}$. Allowing $\pm\infty$ as limit points, there is a subsequence $\{n''\}$ such that $s^{(n'')} \to s^*$. Let h^* be the tree having the same skeleton as the $h_{n''}$ with split points at s^*. By the Schwarz inequality

$$(f, h_{n''} - h^*) \le \|f\| \cdot \|h_{n''} - h^*\|.$$

But clearly, $\|h_{n''} - h^*\| \to 0$

PROOF OF THEOREM 3. (a) If the algorithm stops at a finite M, then

$$\int \sinh\bigl(r(\mathbf{x}) + S_M(\mathbf{x})\bigr) h(\mathbf{x}) Q(d\mathbf{x}) = 0$$

for all trees h. Hence, for all functions f in $L_2(P)$

$$\int \sinh\bigl(r(\mathbf{x}) + s_M(\mathbf{x})\bigr) f(\mathbf{x}) Q(d\mathbf{x}) = 0,$$

which implies that $s_M(\mathbf{x})$ equals $-r(\mathbf{x})$ a.s.

(b) If the algorithm continues indefinitely, let

$$I_M = \int \cosh\bigl(r(\mathbf{x}) + s_M(\mathbf{x})\bigr) Q(d\mathbf{x}),$$

$$b_{M+1} = -\inf_h \int \sinh\bigl(r(\mathbf{x}) + s_M(\mathbf{x})\bigr) h(\mathbf{x}) Q(d\mathbf{x}) / I_M$$

and h_{M+1} a minimizing tree. Using

$$\cosh(A + B) = \cosh(A)\cosh(B) + \sinh(A)\sinh(B),$$

$$\cosh(\alpha h_{M+1}) = \cosh(\alpha), \qquad \sinh(\alpha h_{M+1}) = h_{M+1}\sinh(\alpha)$$

we get that

$$\int \cosh\bigl(r(\mathbf{x}) + s_M(\mathbf{x}) + \alpha h_{M+1}(\mathbf{x})\bigr) Q(d\mathbf{x})$$

$$= \bigl(\cosh(\alpha) - b_{M+1}\sinh(\alpha)\bigr) I_M.$$

The minimizing value of α is

$$c_{M+1} = \tfrac{1}{2}\log\bigl((1+b_{M+1})/(1-b_{M+1})\bigr) = b_{M+1} + O(b_{M+1}^2) \tag{6}$$

and

$$I_{M+1}^2 = I_M^2(1-b_{M+1}^2).$$

Since I_M is greater than or equal to one, this implies that $\sum_1^\infty b_m^2 < \infty$. We will use the inequality

$$\left|\int \sinh\bigl(r(\mathbf{x})+s_M(\mathbf{x})\bigr)s_M(\mathbf{x})\,Q(d\mathbf{x})\right| \le I_1 b_{M+1}\sum_1^M c_m. \tag{7}$$

(c) There is a subsequence $\{m'\}$ such that $\|r+s_{m'}\|\to 0$. To show this, note

$$\left(b_{M+1}\sum_1^M c_m\right)^2 \le M b_{M+1}^2 \sum_1^M c_m^2. \tag{8}$$

Suppose $\liminf_M (b_{M+1}\sum_1^M c_m) > 0$. Then since $\sum_1^\infty c_m^2 < \infty$, for all M sufficiently large, $b_{M+1}^2 > a/M$, which cannot be. Thus, there is a subsequence such that the left-hand side of (7) goes to zero.

For f any finite sum of trees, $\int \sinh(r(\mathbf{x})+s_M(\mathbf{x}))f(\mathbf{x})\,Q(d\mathbf{x})\to 0$. Since $r(\mathbf{x})$ is continuous and bounded, for any $\varepsilon>0$ there is a finite sum of trees $\tilde r(\mathbf{x})$ such that $\sup_{\mathbf{x}\in S}|r(\mathbf{x})-\tilde r(\mathbf{x})|\le\varepsilon$. Hence

$$\begin{aligned}&\left|\int \sinh\bigl(r(\mathbf{x})+s_M(\mathbf{x})\bigr)r(\mathbf{x})\,Q(d\mathbf{x})\right| \\ &\quad\le \left|\int \sinh\bigl(r(\mathbf{x})+s_M(\mathbf{x})\bigr)\tilde r(\mathbf{x})\,Q(d\mathbf{x})\right| + \varepsilon\int \bigl|\sinh\bigl(r(\mathbf{x})+s_M(\mathbf{x})\bigr)\bigr|\,Q(d\mathbf{x}).\end{aligned} \tag{9}$$

The last term in (9) can be bounded by

$$\varepsilon\int \cosh\bigl(r(\mathbf{x})+s_M(\mathbf{x})\bigr)\,Q(d\mathbf{x}) \le C\varepsilon,$$

leading to the conclusion that $\int \sinh(r(\mathbf{x})+s_M(\mathbf{x}))r(\mathbf{x})\,Q(d\mathbf{x})\to 0$. Thus,

$$\int \sinh\bigl(r(\mathbf{x})+s_{m'}(\mathbf{x})\bigr)\bigl(r(\mathbf{x})+s_{m'}(\mathbf{x})\bigr)\,Q(d\mathbf{x})\to 0.$$

Since $x\sinh(x)\ge x^2$ this proves assertion (c).

(d) On the full sequence $\|r+s_M\|\to 0$. This follows by noting that $x\sinh(x)\ge \cosh x - 1$. Hence, on the subsequence $I_{m'}\to 1$. Since I_M is nonincreasing in M, on the whole sequence $I_M\to 1$. Now use $\cosh x - 1\ge x^2$ to prove (d). $\square$

REFERENCES

[1] BAUER, E. and KOHAVI, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning* **36** 105–139.

[2] BREIMAN, L. (1996). Bagging predictors. *Machine Learning* **24** 123–140.

[3] BREIMAN, L. (1997). Arcing the edge. Technical Report 486, Dept. Statistics, Univ. California, Berkeley. Available at www.stat.berkeley.edu.

[4] BREIMAN, L. (1998). Arcing classifiers (with discussion). *Ann. Statist.* **26** 801–849.

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

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

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

[8] DIETTERICH, T. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning* **40** 139–157.

[9] DRUCKER, H. and CORTES, C. (1996). Boosting decision trees. In *Advances in Neural Information Processing Systems* **8** 479–485. MIT Press, Cambridge, MA.

[10] DUNFORD, N. and SCHWARTZ, J. (1958). *Linear Operators. I.* Interscience Publishers, New York.

[11] FORSYTHE, G. E. and WASOW, W. R. (1960). *Finite-Difference Methods for Partial Differential Equations*. Wiley, New York.

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

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

[14] JIANG, W. (2004). Process consistency for AdaBoost. *Ann. Statist.* **32** 13–29.

[15] LUGOSI, G. and VAYATIS, N. (2004). On the Bayes-risk consistency of regularized boosting methods. *Ann. Statist.* **32** 30–55.

[16] 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 Comp. Sci.* **2375** 319–333. Springer, New York.

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

[18] SCHAPIRE, R. and SINGER, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning* **37** 297–336.

[19] WHEWAY, V. (1999). Variance reduction trends on "boosted" classifiers. Unpublished manuscript.

[20] 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: leo@stat.berkeley.edu