Consistency of random forests and other averaging classifiers *

Gérard Biau
LSTA & LPMA
Université Pierre et Marie Curie – Paris VI
Boîte 158, 175 rue du Chevaleret
75013 Paris, France

Luc Devroye
School of Computer Science, McGill University
Montreal, Canada H3A 2K6

Gábor Lugosi
ICREA and Department of Economics
Pompeu Fabra University
Ramon Trias Fargas 25-27
08005 Barcelona, Spain

May 9, 2008

This paper is dedicated to the memory of Leo Breiman.

Abstract

In the last years of his life, Leo Breiman promoted random forests for use in classification. He suggested using averaging as a means

*The second author's research was sponsored by NSERC Grant A3456 and FQRNT Grant 90-ER-0291. The third author acknowledges support by the Spanish Ministry of Science and Technology grant MTM2006-05650 and by the PASCAL Network of Excellence under EC grant no. 506778.

of obtaining good discrimination rules. The base classifiers used for averaging are simple and randomized, often based on random samples from the data. He left a few questions unanswered regarding the consistency of such rules. In this paper, we give a number of theorems that establish the universal consistency of averaging rules. We also show that some popular classifiers, including one suggested by Breiman, are not universally consistent.

1 Introduction

Ensemble methods, popular in machine learning, are learning algorithms that construct a set of many individual classifiers (called base learners) and combine them to classify new data points by taking a weighted or unweighted vote of their predictions. It is now well-known that ensembles are often much more accurate than the individual classifiers that make them up. The success of ensemble algorithms on many benchmark data sets has raised considerable interest in understanding why such methods succeed and identifying circumstances in which they can be expected to produce good results. These methods differ in the way the base learner is fit and combined. For example, bagging (Breiman [2]) proceeds by generating bootstrap samples from the original data set, constructing a classifier from each bootstrap sample, and voting to combine. In boosting (Freund and Shapire [13]) and arcing algorithms (Breiman [3]) the successive classifiers are constructed by giving increased weight to those points that have been frequently misclassified, and the classifiers are combined using weighted voting. On the other hand, random split selection (Dietterich [11]) grows trees on the original data set. For a fixed number S, at each node, S best splits (in terms of minimizing deviance) are found and the actual split is randomly and uniformly selected from them. For a comprehensive review of ensemble methods, we refer the reader to Dietterich [12] and the references therein.

Breiman [5] provides a general framework for tree ensembles called “random forests”. Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees. Thus, a random forest is a classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. Algorithms for inducing a random forest were first developed by Breiman and Cutler, and “Random Forests” is their trademark. The web page http://www.stat.berkeley.edu/users/breiman/RandomForests

provides a collection of downloadable technical reports, and gives an overview of random forests as well as comments on the features of the method.

Random forests have been shown to give excellent performance on a number of practical problems. They work fast, generally exhibit a substantial performance improvement over single tree classifiers such as CART, and yield generalization error rates that compare favorably to the best statistical and machine learning methods. In fact, random forests are among the most accurate general-purpose classifiers available (see for example Breiman [5]).

Different random forests differ in how randomness is introduced in the tree building process, ranging from extreme random splitting strategies (Breiman [4], Cutler and Zhao [7]) to more involved data-dependent strategies (Amit and Geman [1], Breiman [5], Dietterich [11]). As a matter of fact, the statistical mechanism of random forests is not yet fully understood and is still under active investigation. Unlike single trees, where consistency is proved letting the number of observations in each terminal node become large (Devroye, Györfi, and Lugosi [10, Chapter 20]), random forests are generally built to have a small number of cases in each terminal node. Although the mechanism of random forest algorithms appears simple, it is difficult to analyze and remains largely unknown. Some attempts to investigate the driving force behind consistency of random forests are by Breiman [4, 6] and Lin and Jeon [14], who establish a connection between random forests and adaptive nearest neighbor methods. Meinshausen [15] proved consistency of certain random forests in the context of so-called quantile regression.

In this paper we offer consistency theorems for various versions of random forests and other randomized ensemble classifiers. In Section 2 we introduce a general framework for studying classifiers based on averaging randomized base classifiers. We prove a simple but useful proposition showing that averaged classifiers are consistent whenever the base classifiers are.

In Section 3 we prove consistency of two simple random forest classifiers, the *purely random forest* (suggested by Breiman as a starting point for study) and the *scale-invariant random forest* classifiers.

In Section 4 it is shown that averaging may convert inconsistent rules into consistent ones.

In Section 5 we briefly investigate consistency of bagging rules. We show that, in general, bagging preserves consistency of the base rule and it may even create consistent rules from inconsistent ones. In particular, we show that if the bootstrap samples are sufficiently small, the bagged version of the 1-nearest neighbor classifier is consistent.

Finally, in Section 6 we consider random forest classifiers based on randomized, greedily grown tree classifiers. We argue that some greedy random forest classifiers, including Breiman's random forest classifier, are inconsistent and suggest a consistent greedy random forest classifier.

2 Voting and averaged classifiers

Let $(X, Y), (X_1, Y_1), \ldots, (X_n, Y_n)$ be i.i.d. pairs of random variables such that X (the so-called *feature vector*) takes its values in $\mathbb{R}^d$ while Y (the *label*) is a binary $\{0, 1\}$-valued random variable. The joint distribution of (X, Y) is determined by the marginal distribution μ of X (i.e., $\mathbb{P}\{X \in A\} = \mu(A)$ for all Borel sets $A \subset \mathbb{R}^d$) and the *a posteriori* probability $\eta : \mathbb{R}^d \to [0, 1]$ defined by

$$\eta(x) = \mathbb{P}\{Y = 1|X = x\} \ .$$

The collection $(X_1, Y_1), \ldots, (X_n, Y_n)$ is called the *training data*, and is denoted by D_n. A classifier g_n is a binary-valued function of X and D_n whose probability of error is defined by

$$L(g_n) = \mathbb{P}_{(X,Y)}\{g_n(X, D_n) \neq Y\}$$

where $\mathbb{P}_{(X,Y)}$ denotes probability with respect to the pair (X, Y) (i.e., conditional probability, given D_n). For brevity, we write $g_n(X) = g_n(X, D_n)$. It is well-known (see for example [10]) that the classifier that minimizes the probability of error, the so-called *Bayes classifier* is $g^*(x) = \mathbb{1}_{\{\eta(x) \geq 1/2\}}$. The risk of g^* is called the Bayes risk: $L^* = L(g^*)$.

A sequence $\{g_n\}$ of classifiers is *consistent* for a certain distribution of (X, Y) if $L(g_n) \to L^*$ in probability.

In this paper we investigate classifiers that calculate their decisions by taking a majority vote over *randomized classifiers*. A randomized classifier may use a random variable Z to calculate its decision. More precisely, let $\mathcal{Z}$ be some measurable space and let Z take its values in $\mathcal{Z}$. A randomized classifier is an arbitrary function of the form $g_n(X, Z, D_n)$, which we abbreviate by $g_n(X, Z)$. The probability of error of g_n becomes

$$L(g_n) = \mathbb{P}_{(X,Y),Z}\{g_n(X, Z, D_n) \neq Y\} = \mathbb{P}\{g_n(X, Z, D_n) \neq Y|D_n\} \ .$$

The definition of consistency remains the same by augmenting the probability space appropriately to include the randomization.

Given any randomized classifier, one may calculate the classifier for various draws of the randomizing variable Z. It is then a natural idea to define an averaged classifier by taking a majority vote among the obtained random classifiers. Assume that $Z_1, \ldots, Z_m$ are identically distributed draws of the randomizing variable, having the same distribution as Z. Throughout the paper, we assume that $Z_1, \ldots, Z_m$ are independent, conditionally on X, Y, and D_n. Letting $Z^m = (Z_1, \ldots, Z_m)$, one may define the corresponding *voting classifier* by

$$g_n^{(m)}(x, Z^m, D_n) = \begin{cases} 1 & \text{if } \frac{1}{m}\sum_{j=1}^{m} g_n(x, Z_j, D_n) \geq \frac{1}{2} \ , \\ 0 & \text{otherwise.} \end{cases}$$

By the strong law of large numbers, for any fixed x and D_n for which $\mathbb{P}_Z\{g_n(x, Z, D_n) = 1\} \neq 1/2$, we have almost surely $\lim_{m\to\infty} g_n^{(m)}(x, Z^m, D_n) = \overline{g}_n(x, D_n)$, where $\overline{g}_n(x, D_n) = \overline{g}_n(x) = \mathbb{1}_{\{\mathbb{E}_Z g_n(x,Z) \geq 1/2\}}$ is a (non-randomized) classifier that we call the *averaged classifier*. (Here $\mathbb{P}_Z$ and $\mathbb{E}_Z$ denote probability and expectation with respect to the randomizing variable Z, i.e., conditionally on X, Y, and D_n.)

$\overline{g}_n$ may be interpreted as an idealized version of the classifier $g_n^{(m)}$ that draws many independent copies of the randomizing variable Z and takes a majority vote over the resulting classifiers.

Our first result states that consistency of a randomized classifier is preserved by averaging.

Proposition 2.1 *Assume that the sequence $\{g_n\}$ of randomized classifiers is consistent for a certain distribution of (X, Y). Then the voting classifier $g_n^{(m)}$ (for any value of m) and the averaged classifier $\overline{g}_n$ are also consistent.*

PROOF. Consistency of $\{g_n\}$ is equivalent to saying that $\mathbb{E}L(g_n) = \mathbb{P}\{g_n(X, Z) \neq Y\} \to L^*$. In fact, since $\mathbb{P}\{g_n(X, Z) \neq Y | X = x\} \geq \mathbb{P}\{g^*(X) \neq Y | X = x\}$ for all $x \in \mathbb{R}^d$, consistency of $\{g_n\}$ means that for μ-almost all x,

$$\mathbb{P}\{g_n(X, Z) \neq Y | X = x\} \to \mathbb{P}\{g^*(X) \neq Y | X = x\} = \min(\eta(x), 1 - \eta(x)) \ .$$

Without loss of generality, assume that $\eta(x) > 1/2$. (In the case of $\eta(x) = 1/2$ any classifier has a conditional probability of error $1/2$ and there is nothing to prove.) Then $\mathbb{P}\{g_n(X,Z) \neq Y | X = x\} = (2\eta(x) - 1)\mathbb{P}\{g_n(x,Z) = 0\} + 1 - \eta(x)$, and by consistency we have $\mathbb{P}\{g_n(x,Z) = 0\} \to 0$.

To prove consistency of the voting classifier $g_n^{(m)}$, it suffices to show that $\mathbb{P}\{g_n^{(m)}(x,Z^m) = 0\} \to 0$ for μ-almost all x for which $\eta(x) > 1/2$. However,

$$\begin{aligned}
\mathbb{P}\{g_n^{(m)}(x,Z^m) = 0\} &= \mathbb{P}\left\{(1/m)\sum_{j=1}^{m} \mathbb{1}_{\{g_n(x,Z_j)=0\}} > 1/2\right\} \\
&\leq 2\mathbb{E}\left[(1/m)\sum_{j=1}^{m} \mathbb{1}_{\{g_n(x,Z_j)=0\}}\right] \\
&\qquad \text{(by Markov's inequality)} \\
&= 2\mathbb{P}\{g_n(x,Z) = 0\} \to 0 \ .
\end{aligned}$$

Consistency of the averaged classifier is proved by a similar argument.
□

3 Random forests

Random forests, introduced by Breiman, are averaged classifiers in the sense defined in Section 2.

Formally, a random forest with m trees is a classifier consisting of a collection of randomized base tree classifiers $g_n(x,Z_1),\ldots,g_n(x,Z_m)$ where $Z_1,\ldots,Z_m$ are identically distributed random vectors, independent conditionally on X, Y, and D_n.

The randomizing variable is typically used to determine how the successive cuts are performed when building the tree such as selection of the node and the coordinate to split, as well as the position of the split. The random forest classifier takes a majority vote among the random tree classifiers. If m is large, the random forest classifier is well approximated by the averaged classifier $\overline{g}_n(x) = \mathbb{1}_{\{\mathbb{E}_Z g_n(x,Z) \geq 1/2\}}$. For brevity, we state most results of this paper for the averaged classifier only, though by Proposition 2.1 various results remain true for the voting classifier $g_n^{(m)}$ as well.

In this section we analyze a simple random forest already considered by Breiman [4], which we call the *purely random forest*.

The random tree classifier $g_n(x,Z)$ is constructed as follows. Assume, for simplicity, that μ is supported on $[0,1]^d$. All nodes of the tree are associated with rectangular cells such that at each step of the construction of the tree, the collection of cells associated with the leaves of the tree (i.e., external nodes) forms a partition of $[0,1]^d$. The root of the random tree is $[0,1]^d$ itself. At each step of the construction of the tree, a leaf is chosen uniformly at random. The split variable J is then selected uniformly at random from the d candidates $x^{(1)}, \ldots, x^{(d)}$. Finally, the selected cell is split along the randomly chosen variable at a random location, chosen according to a uniform random variable on the length of the chosen side of the selected cell. The procedure is repeated k times where $k \geq 1$ is a deterministic parameter, fixed beforehand by the user, and possibly depending on n.

The randomized classifier $g_n(x, Z)$ takes a majority vote among all Y_i for which the corresponding feature vector X_i falls in the same cell of the random partition as x. (For concreteness, break ties in favor of the label 1.)

The purely random forest classifier is a radically simplified version of random forest classifiers used in practice. The main simplification lies in the fact that recursive cell splits do not depend on the labels $Y_1, \ldots, Y_n$. The next theorem mainly serves as an illustration of how the consistency problem of random forest classifiers may be attacked. More involved versions of random forest classifiers are discussed in subsequent sections.

Theorem 3.1 *Assume that the distribution of X is supported on $[0,1]^d$. Then the purely random forest classifier $\overline{g}_n$ is consistent whenever $k \to \infty$ and $k/n \to 0$ as $k \to \infty$.*

PROOF. By Proposition 2.1 it suffices to prove consistency of the randomized base tree classifier g_n. To this end, we recall a general consistency theorem for partitioning classifiers proved in Devroye, Györfi, and Lugosi [10, Theorem 6.1]. According to this theorem, g_n is consistent if both $\operatorname{diam}(A_n(X, Z)) \to 0$ in probability and $N_n(X, Z) \to \infty$ in probability, where $A_n(x, Z)$ is the rectangular cell of the random partition containing x and

$$N_n(x, Z) = \sum_{i=1}^{n} \mathbb{1}_{\{X_i \in A_n(x,Z)\}}$$

is the number of data points falling in the same cell as x.

First we show that $N_n(X, Z) \to \infty$ in probability. Consider the random tree partition defined by Z. Observe that the partition has $k+1$ rectangular cells, say $A_1, \ldots, A_{k+1}$. Let $N_1, \ldots, N_{k+1}$ denote the number of points of

$X, X_1, \ldots, X_n$ falling in these $k+1$ cells. Let $S = \{X, X_1, \ldots, X_n\}$ denote the set of positions of these $n+1$ points. Since these points are independent and identically distributed, fixing the *set* S (but not the order of the points) and Z, the conditional probability that X falls in the i-th cell equals $N_i/(n+1)$. Thus, for every fixed $t > 0$,

$$\begin{aligned} \mathbb{P}\{N_n(X,Z) < t\} &= \mathbb{E}\left[\mathbb{P}\{N_n(X,Z) < t | S, Z\}\right] \\ &= \mathbb{E}\left[\sum_{i: N_i < t} \frac{N_i}{n+1}\right] \leq (t-1)\frac{k+1}{n+1} \end{aligned}$$

which converges to zero by our assumption on k.

It remains to show that $\operatorname{diam}(A_n(X,Z)) \to 0$ in probability. To this aim, let $V_n = V_n(x,Z)$ be the size of the first dimension of the rectangle containing x. Let $T_n = T_n(x,Z)$ be the number of times that the box containing x is split when we construct the random tree partition.

Let K_n be binomial $(T_n, 1/d)$, representing the number of times the box containing x is split along the first coordinate.

Clearly, it suffices to show that $V_n(x,Z) \to 0$ in probability for μ-almost all x, so it is enough to show that for all x, $\mathbb{E}[V_n(x,Z)] \to 0$. Observe that if $U_1, U_2, \ldots$ are independent uniform $[0,1]$, then

$$\begin{aligned} \mathbb{E}[V_n(x,Z)] &\leq \mathbb{E}\left[\mathbb{E}\left[\prod_{i=1}^{K_n} \max(U_i, 1-U_i) \middle| K_n\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\max(U_1, 1-U_1)\right]^{K_n}\right] \\ &= \mathbb{E}\left[(3/4)^{K_n}\right] \\ &= \mathbb{E}\left[\left(1 - \frac{1}{d} + \frac{3}{4d}\right)^{T_n}\right] \\ &= \mathbb{E}\left[\left(1 - \frac{1}{4d}\right)^{T_n}\right] . \end{aligned}$$

Thus, it suffices to show that $T_n \to \infty$ in probability. To this end, note that the partition tree is statistically related to a random binary search tree with $k+1$ external nodes (and thus k internal nodes). Such a tree is obtained as follows. Initially, the root is the sole external node, and there are no internal nodes. Select an external node uniformly at random, make it an internal node and give it two children, both external. Repeat until we have precisely k internal nodes and $k+1$ external nodes. The resulting tree is the random binary search tree on k internal nodes (see Devroye [8] and Mahmoud [16] for more equivalent constructions of random binary search trees). It is known that all levels up to $\ell = \lfloor 0.37 \log k \rfloor$ are full with probability tending to one as $k \to \infty$ (Devroye [9]). The last full level F_n is called the fill-up level. Clearly, the partition tree has this property. Therefore, we know that all final cells have been cut at least ℓ times and therefore $T_n \geq \ell$ with probability converging to 1. This concludes the proof of Theorem 3.1. □

Remark. We observe that the largest first dimension among external nodes does not tend to zero in probability except for $d = 1$. For $d \geq 2$, it tends to a limit random variable that is not atomic at zero (this can be shown using the theory of branching processes). Thus the proof above could not have used the uniform smallness of all cells. Despite the fact that the random partition contains some cells of huge diameter of non-shrinking size, the rule based on it is consistent. □

Next we consider a scale-invariant version of the purely random forest classifier. In this variant the root cell is the entire feature space and the random tree is grown up to k cuts. The leaf cell to cut and the direction J in which the cell is cut are chosen uniformly at random, exactly as in the purely random forest classifier. The only difference is that the position of the cut is now chosen in a data-based manner: if the cell to be cut contains N of the data points $X, X_1, \ldots, X_n$, then a random index I is chosen uniformly from the set $\{0, 1, \ldots, N\}$ and the cell is cut so that, when ordered by their J-th components, the points with the I smallest values fall in one of the subcells and the rest in the other. To avoid ties, we assume that the distribution of X has non-atomic marginals. In this case the random tree is well-defined with probability one. Just like before, the associated classifier takes a majority vote over the labels of the data points falling in the same cell as X. The *scale-invariant random forest* classifier is defined as the corresponding averaged classifier.

Theorem 3.2 *Assume that the distribution of X has non-atomic marginals in $\mathbb{R}^d$. Then the scale-invariant random forest classifier $\overline{g}_n$ is consistent whenever $k \to \infty$ and $k/n \to 0$ as $k \to \infty$.*

PROOF. Once again, we may use Proposition 2.1 and [10, Theorem 6.1] to prove consistency of the randomized base tree classifier g_n. The proof of the fact that $N_n(X, Z) \to \infty$ in probability is the same as in Theorem 3.1.

To show that $\mathrm{diam}(A_n(X, Z)) \to 0$ in probability, we begin by noting that, just as in the case of the purely random forest classifier, the partition tree is equivalent to a binary search tree, and therefore with probability converging to one, all final cells have been cut at least $\ell = \lfloor 0.37 \log k \rfloor$ times.

Since the classification rule is scale-invariant, we may assume, without loss of generality, that the distribution of X is concentrated on the unit cube $[0, 1]^d$.

Let n_i denote the cardinality of the i-th cell in the partition, $1 \leq i \leq k+1$, where the cardinality of a cell C is $|C \cap \{X, X_1, \ldots, X_n\}|$. Thus, $\sum_{i=1}^{k+1} n_i = n + 1$. Let V_i be the first dimension of the i-th cell. Let $V(X)$ be the first dimension of the cell that contains X. Clearly, given the n_i's, $V(X) = V_i$ with probability $n_i/(n + 1)$. We need to show that $\mathbb{E}[V(X)] \to 0$. But we have

$$\mathbb{E}[V(X)] = \mathbb{E}\left[\frac{\sum_{i=1}^{k+1} n_i V_i}{n+1}\right].$$

So, it suffices to show that $\mathbb{E}[\sum_i n_i V_i] = o(n)$.

It is worthy of mention that the random split of a box can be imagined as follows. Given that we split along the s-th coordinate axis, and that a box has m points, then we select one of the $m + 1$ spacings defined by these m points uniformly at random, still for that s-th coordinate. We cut that spacing properly but are free to do so anywhere. We can cut in proportions $\lambda, 1 - \lambda$ with $\lambda \in (0, 1)$, and the value of λ may vary from cut to cut and even be data-dependent. In fact, then, each internal and external node of our partition tree has associated with it two important quantities, a cardinality, and its first dimension. If we keep using i to index cells, then we can use n_i and V_i for the i-th cell, even if it is an internal cell.

Let A be the collection of external nodes in the subtree of the i-th cell. Then trivially,

$$\sum_{j \in A} n_j V_j \leq n_i V_i \leq n.$$

Thus, if E is the collection of all external nodes of a partition tree, ℓ is at most the minimum path distance from any cell in E to the root, and L is the collection of all nodes at distance ℓ from the root, then, by the last inequality,

$$\sum_{i\in E} n_i V_i \le \sum_{i\in L} n_i V_i.$$

Thus, using the notion of fill-up level F_n of the binary search tree, and setting $\ell = \lfloor 0.37 \log k \rfloor$, we have

$$\mathbb{E}\left[\sum_{i\in E} n_i V_i\right] \le n\mathbb{P}\{F_n > \ell\} + \mathbb{E}\left[\sum_{i\in L} n_i V_i\right].$$

We have seen that the first term is $o(n)$. We argue that the second term is not more than $n(1-1/(8d))^\ell$, which is $o(n)$ since $k \to \infty$. That will conclude the proof.

It suffices now to argue recursively and fix one cell of cardinality n and first dimension V. Let C be the collection of its children. We will show that

$$\mathbb{E}\left[\sum_{i\in C} n_i V_i\right] \le \left(1 - \frac{1}{8d}\right) nV.$$

Repeating this recursively ℓ times shows that

$$\mathbb{E}\left[\sum_{i\in L} n_i V_i\right] \le n\left(1 - \frac{1}{8d}\right)^\ell$$

because $V = 1$ at the root.

Fix that cell of cardinality n, and assume without loss of generality that $V = 1$. Let the spacings along the first coordinate be $a_1, \ldots, a_{n+1}$, their sum being one. With probability $1 - 1/d$, there the first axis is not cut, and thus, $\sum_{i\in C} n_i V_i = n$. With probability $1/d$, the first axis is cut in two parts. We will show that conditional on the event that the first direction is cut,

$$\mathbb{E}\left[\sum_i n_i V_i\right] \le \frac{7n}{8}.$$

Unconditionally, we have

$$\mathbb{E}\left[\sum_i n_i V_i\right] \le \left(1 - \frac{1}{d}\right)n + \frac{1}{d}\cdot\frac{7n}{8} = \left(1 - \frac{1}{8d}\right)n\ ,$$

as required. So, let us prove the conditional result.

Using δ_j to denote numbers drawn from $(0,1)$, possibly random, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_i n_i V_i\right] \\
&= \frac{1}{n+1}\mathbb{E}\Big[\sum_{j=1}^{n+1}[(j-1)(a_1+\cdots+a_{j-1}+a_j\delta_j) \\
&\qquad +(n+1-j)(a_j(1-\delta_j)+a_{j+1}+\cdots+a_{n+1})]\Big] \\
&= \frac{1}{n+1}\mathbb{E}\left[\sum_{k=1}^{n+1} a_k\left(\sum_{k<j\le n+1}(j-1)\right.\right. \\
&\qquad \left.\left.+\sum_{1\le j<k}(n+1-j)+\delta_k(k-1)+(1-\delta_k)(n+1-k)\right)\right] \\
&\le \frac{1}{n+1}\left(\sum_{k=1}^{n+1} a_k\left(n(n+1)-\frac{k(k-1)}{2}\right.\right. \\
&\qquad \left.\left.-\frac{(n-k+1)(n-k+2)}{2}+\max(k-1,n+1-k)\right)\right) \\
&= \frac{1}{n+1}\left(\sum_{k=1}^{n+1} a_k\left(\frac{n(n+1)}{2}+(k-1)(n+1-k)+\max(k-1,n+1-k)\right)\right) \\
&\le \frac{1}{n+1}\left(\left(\frac{n(n+1)}{2}+\left(\frac{n}{2}\right)^2+n\right)\sum_{k=1}^{n+1} a_k\right) \\
&= n\left(\frac{3n/4+(3/2)}{n+1}\right) \\
&\le \frac{7n}{8} \quad \text{if } n>4.
\end{aligned}$$

□

Our definition of the scale-invariant random forest classifier permits cells to be cut such that one of the created cells becomes empty. One may easily prevent this by artificially forcing a minimum number of points in each cell. This may be done by restricting the random position of each cut so that both created subcells contain at least, say, m points. By a minor modification of the proof above it is easy to see that as long as m is bounded by a constant, the resulting random forest classifier remains consistent under the same conditions as in Theorem 3.2.

4 Creating consistent rules by randomization

Proposition 2.1 shows that if a randomized classifier is consistent, then the corresponding averaged classifier remains consistent. The converse is not true. There exist inconsistent randomized classifiers such that their averaged version becomes consistent. Indeed, Breiman's [5] original random forest classifier builds tree classifiers by successive randomized cuts until the cell of the point X to be classified contains only one data point, and classifies X as the label of this data point. Breiman's random forest classifier is just the averaged version of such randomized tree classifiers. The randomized base classifier $g_n(x, Z)$ is obviously not consistent for all distributions.

This does not imply that the averaged random forest classifier is not consistent. In fact, in this section we will see that averaging may "boost" inconsistent base classifiers into consistent ones. We point out in Section 6 that there are distributions of (X, Y) for which Breiman's random forest classifier is not consistent. The counterexample shown in Proposition 6.1 is such that the distribution of X doesn't have a density. It is possible, however, that Breiman's random forest classifier is consistent whenever the distribution of X has a density. Breiman's rule is difficult to analyze as each cut of the random tree is determined by a complicated function of the entire data set D_n (i.e., both feature vectors and labels). However, in Section 6 below we provide arguments suggesting that Breiman's random forest is not consistent when a density exists. Instead of Breiman's rule, next we analyze a stylized version by showing that inconsistent randomized rules that take the label of only one neighbor into account can be made consistent by averaging.

For simplicity, we consider the case $d = 1$, though the whole argument extends, in a straightforward way, to the multivariate case. To avoid complications introduced by ties, assume that X has a non-atomic distribution. Define a randomized nearest neighbor rule as follows: for a fixed $x \in \mathbb{R}$, let $X_{(1)}(x), X_{(2)}(x), \ldots, X_{(n)}(x)$ be the ordering of the data points $X_1, \ldots, X_n$ according to increasing distances from x. Let $U_1, \ldots, U_n$ be i.i.d. random variables, uniformly distributed over $[0, 1]$. The vector of these random variables constitutes the randomization Z of the classifier. We define $g_n(x, Z)$ to be equal to the label $Y_{(i)}(x)$ of the data point $X_{(i)}(x)$ for which

$$\max(i, mU_i) \le \max(j, mU_j) \quad \text{for all } j = 1, \ldots, n$$

where $m \le n$ is a parameter of the rule. We call $X_{(i)}(x)$ the perturbed nearest neighbor of x. Note that $X_{(1)}(x)$ is the (unperturbed) nearest neighbor of x. To obtain the perturbed version, we artificially add a random uniform coordinate and select a data point with the randomized rule defined above. Since ties occur with probability zero, the perturbed nearest neighbor classifier is well defined almost surely. It is clearly not, in general, a consistent classifier.

Call the corresponding averaged classifier $\overline{g}_n(x) = \mathbb{1}_{\{\mathbb{E}_Z g_n(x,Z) \ge 1/2\}}$ the *averaged perturbed nearest neighbor classifier*.

In the proof of the consistency result below, we use Stone's [17] general consistency theorem for locally weighted average classifiers, see also [10, Theorem 6.3]. Stone's theorem concerns classifiers that take the form

$$g_n(x) = \mathbb{1}_{\{\sum_{i=1}^n Y_i W_{ni}(x) \ge \sum_{i=1}^n (1-Y_i) W_{ni}(x)\}}$$

where the weights $W_{ni}(x) = W_{ni}(x, X_1, \ldots, X_n)$ are non-negative and sum to one. According to Stone's theorem, consistency holds if the following three conditions are satisfied:

(i)

$$\lim_{n\to\infty} \mathbb{E}\left[\max_{1\le i\le n} W_{ni}(X)\right] = 0.$$

(ii) For all $a > 0$,

$$\lim_{n\to\infty} \mathbb{E}\left[\sum_{i=1}^n W_{ni}(X) \mathbb{1}_{\{\|X_i - X\| > a\}}\right] = 0.$$

(iii) There is a constant c such that, for every non-negative measurable function f satisfying $\mathbb{E}f(X) < \infty$,

$$\mathbb{E}\left[\sum_{i=1}^n W_{ni}(X) f(X_i)\right] \le c\mathbb{E}f(X).$$

Theorem 4.1 *The averaged perturbed nearest neighbor classifier $\overline{g}_n$ is consistent whenever the parameter m is such that $m \to \infty$ and $m/n \to 0$.*

PROOF. If we define

$$W_{ni}(x) = \mathbb{P}_Z\{X_i \text{ is the perturbed nearest neighbor of } x\}$$

then it is clear that the averaged perturbed nearest neighbor classifier is a locally weighted average classifier and Stone's theorem may be applied. It is convenient to introduce the notation

$$p_{ni}(x) = \mathbb{P}_Z\{X_{(i)}(x) \text{ is the perturbed nearest neighbor of } x\}$$

and write $W_{ni}(x) = \sum_{j=1}^{n} \mathbb{1}_{\{X_i = X_{(j)}(x)\}} p_{nj}(x)$.

To check the conditions of Stone's theorem, first note that

$$\begin{aligned} p_{ni}(x) &= \mathbb{P}\{mU_i \le i \le \min_{j<i} mU_j\} + \mathbb{P}\{i < mU_i \le \min_{j\le n} \max(j, mU_j)\} \\ &= \mathbb{1}_{\{i\le m\}} \frac{i}{m}\left(1-\frac{i}{m}\right)^{i-1} + \mathbb{P}\{i < mU_i \le \min_{j\le n} \max(j, mU_j)\} \ . \end{aligned}$$

Now we are prepared to check the conditions of Stone's theorem. To prove that (i) holds, note that by monotonicity of $p_{ni}(x)$ in i, it suffices to show that $p_{n1}(x) \to 0$.

But clearly, for $m \ge 2$,

$$\begin{aligned} p_{n1}(x) &\le \frac{1}{m} + \mathbb{P}\left\{U_1 \le \min_{j\le m} \max\left(\frac{j}{m}, U_j\right)\right\} \\ &= \frac{1}{m} + \mathbb{E}\left[\prod_{j=2}^{m} \mathbb{P}\left\{U_1 \le \max\left(\frac{j}{m}, U_j\right) |U_1\right\}\right] \\ &= \frac{1}{m} + \mathbb{E}\left[\prod_{j=2}^{m} \left[1 - \mathbb{1}_{\{U_1 > j/m\}} U_1\right]\right] \\ &\le \frac{1}{m} + \mathbb{E}\left[(1-U_1)^{mU_1 - 2} \mathbb{1}_{\{\lfloor mU_1 \rfloor \ge 3\}}\right] + \mathbb{P}\left\{\lfloor mU_1 \rfloor < 3\right\} \end{aligned}$$

which converges to zero by monotone convergence as $m \to \infty$.

(ii) follows by the condition $m/n \to 0$ since $\sum_{i=1}^{n} W_{ni}(X) \mathbb{1}_{\{\|X_i - X\| > a\}} = 0$ whenever the distance of m-th nearest neighbor of X to X is at most a. But this happens eventually, almost surely, see [10, Lemma 5.1].

Finally, to check (iii), we use again the monotonicity of $p_{ni}(x)$ in i. We may write $p_{ni}(x) = a_i + a_{i+1} + \cdots + a_n$ for some non-negative numbers

$a_j, 1 \leq j \leq n$, depending upon m and n but not x. Observe that $\sum_{j=1}^n j a_j = \sum_{i=1}^n p_{ni}(x) = 1$. But then

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^n W_{ni}(X) f(X_i)\right] \\
&= \mathbb{E}\left[\sum_{i=1}^n p_{ni}(X) f(X_{(i)})\right] \\
&= \mathbb{E}\left[\sum_{i=1}^n \sum_{j=i}^n a_j f(X_{(i)})\right] \\
&= \mathbb{E}\left[\sum_{j=1}^n a_j \sum_{i=1}^j f(X_{(i)})\right] \\
&= \sum_{j=1}^n a_j \mathbb{E}\left[\sum_{i=1}^j f(X_{(i)})\right] \\
&\leq c \sum_{j=1}^n a_j j \mathbb{E} f(X) \\
&\quad \text{(by Stone's [17] lemma, see [10, Lemma 5.3], where } c \text{ is a constant)} \\
&= c \mathbb{E} f(X) \sum_{j=1}^n a_j j = c \mathbb{E} f(X)
\end{aligned}$$

as desired. □

5 Bagging

One of the first and simplest ways of randomizing and averaging classifiers in order to improve their performance is *bagging*, suggested by Breiman [2]. In bagging, randomization is achieved by generating many bootstrap samples from the original data set. Breiman suggests selecting n training pairs (X_i, Y_i) at random, *with replacement* from the *bag* of all training pairs $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$. Denoting the random selection process by Z, this way one obtains new training data $D_n(Z)$ with possible repetitions and given a classifier $g_n(X, D_n)$, one can calculate the randomized classifier $g_n(X, Z, D_n) = g_n(X, D_n(Z))$. Breiman suggests repeating this procedure for many independent draws of the bootstrap sample, say m of them, and calculating the voting classifier $g_n^{(m)}(X, Z^m, D_n)$ as defined in Section 2.

In this section we consider a generalized version of bagging predictors in which the size of the bootstrap samples is not necessary the same as that the original sample. Also, to avoid complications and ambiguities due to replicated data points, we exclude repetitions in the bootstrapped data. This is assumed for convenience but sampling with replacement can be treated by minor modifications of the arguments below.

To describe the model we consider, introduce a parameter $q_n \in [0, 1]$. In the bootstrap sample $D_n(Z)$ each data pair (X_i, Y_i) is present with probability q_n, independently of each other. Thus, the size of the bootstrapped data is a binomial random variable N with parameters n and q_n. Given a sequence of (non-randomized) classifiers $\{g_n\}$, we may thus define the randomized classifier

$$g_n(X, Z, D_n) = g_N(X, D_n(Z)) ,$$

that is, the classifier is defined based on the randomly re-sampled data. By drawing m independent bootstrap samples $D_n(Z_1), \ldots, D_n(Z_m)$ (with sizes $N_1, \ldots, N_m$), we may define the *bagging classifier* $g_n^{(m)}(X, Z^m, D_n)$ as the voting classifier based on $g_{N_1}(X, D_n(Z_1)), \ldots, g_{N_m}(X, D_n(Z_m))$ as in Section 2. For the theoretical analysis it is more convenient to consider the averaged classifier $\overline{g}_n(x, D_n) = \mathbb{1}_{\{\mathbb{E}_Z g_N(x, D_n(Z)) \geq 1/2\}}$ which is the limiting classifier one obtains as the number m of the bootstrap replicates grows to infinity.

The following result establishes consistency of bagging classifiers under the assumption that the original classifier is consistent. It suffices that the expected size of the bootstrap sample goes to infinity. The result is an immediate consequence of Proposition 2.1. Note that the choice of m does not matter in Theorem 5.1. It can be one, constant, or a function of n.

Theorem 5.1 *Let $\{g_n\}$ be a sequence of classifiers that is consistent for the distribution of (X, Y). Consider the bagging classifiers $g_n^{(m)}(x, Z^m, D_n)$ and $\overline{g}_n(x, D_n)$ defined above, using parameter q_n. If $nq_n \to \infty$ as $n \to \infty$ then both classifiers are consistent.*

If a classifier is insensitive to duplicates in the data, Breiman's original suggestion is roughly equivalent to taking $q_n \approx 1 - 1/e$.

However, it may be advantageous to choose much smaller values of q_n. In fact, small values of q_n may turn inconsistent classifiers into consistent ones via the bagging procedure. We illustrate this phenomenon on the simple example of the 1-nearest neighbor rule.

Recall that the 1-nearest neighbor rule sets $g_n(x, D_n) = Y_{(1)}(x)$ where $Y_{(1)}(x)$ is the label of the feature vector $X_{(1)}(x)$ whose Euclidean distance to x is minimal among all $X_1, \ldots, X_n$. Ties are broken in favor of smallest indices. It is well-known that g_n is consistent only if either $L^* = 0$ or $L^* = 1/2$, otherwise its asymptotic probability of error is strictly greater than L^*. However, by bagging one may turn the 1-nearest neighbor classifier into a consistent one, provided that the size of the bootstrap sample is sufficiently small. The next result characterizes consistency of the bagging version of the 1-nearest neighbor classifier in terms of the parameter q_n.

Theorem 5.2 *The bagging averaged* 1*-nearest neighbor classifier* $\overline{g}_n(x, D_n)$ *is consistent for all distributions of* (X, Y) *if and only if* $q_n \to 0$ *and* $nq_n \to \infty$.

PROOF. It is obvious that both $q_n \to 0$ and $nq_n \to \infty$ are necessary for consistency for all distributions.

Assume now that $q_n \to 0$ and $nq_n \to \infty$. The key observation is that $\overline{g}_n(x, D_n)$ is a locally weighted average classifier for which Stone's consistency theorem, recalled in Section 4, applies.

Recall that for a fixed $x \in \mathbb{R}$, $X_{(1)}(x), X_{(2)}(x), \ldots, X_{(n)}(x)$ denotes the ordering of the data points $X_1, \ldots, X_n$ according to increasing distances from x. (Points with equal distances to x are ordered according to their indices.) Observe that $\overline{g}_n$ may be written as

$$\overline{g}_n(x, D_n) = \mathbb{1}_{\{\sum_{i=1}^n Y_i W_{ni}(x) \geq \sum_{i=1}^n (1-Y_i) W_{ni}(x)\}}$$

where $W_{ni}(x) = \sum_{j=1}^n \mathbb{1}_{\{X_i = X_{(j)}(x)\}} p_{nj}(x)$ and $p_{ni}(x) = (1-q_n)^{i-1} q_n$ is defined as the probability (with respect to the random selection Z of the bootstrap sample) that $X_{(i)}(x)$ is the nearest neighbor of x in the sample $D_n(Z)$. It suffices to prove that the weights $W_{ni}(X)$ satisfy the three conditions of Stone's theorem.

Condition (i) obviously holds because $\max_{1 \leq i \leq n} W_{ni}(X) = p_{n1}(X) = q_n \to 0$.

To check condition (ii), define $k_n = \left\lceil \sqrt{n/q_n} \right\rceil$. Since $nq_n \to \infty$ implies that $k_n/n \to 0$, it follows from [10, Lemma 5.1] that eventually, almost surely,

$\|X - X_{(k_n)}(X)\| \le a$ and therefore

$$\begin{aligned}
\sum_{i=1}^{n} W_{ni}(X)\mathbb{1}_{\{\|X_i - X\| > a\}} &\le \sum_{i=k_n+1}^{n} p_{ni}(X) \\
&= \sum_{i=k_n+1}^{n} q_n(1-q_n)^{i-1} \\
&\le (1-q_n)^{k_n} \\
&\le (1-q_n)^{\sqrt{n/q_n}} \\
&\le e^{-\sqrt{nq_n}}
\end{aligned}$$

where we used $1 - q_n \le e^{-q_n}$. Therefore, $\sum_{i=1}^{n} W_{ni}(X)\mathbb{1}_{\{\|X_i - X\| > a\}} \to 0$ almost surely and Stone's second condition is satisfied by dominated convergence.

Finally, condition (iii) follows from the fact that $p_{ni}(x)$ is monotone decreasing in i, after using an argument as in the proof of Theorem 4.1. □

6 Random forests based on greedily grown trees

In this section we study random forest classifiers that are based on randomized tree classifiers that are constructed in a greedy manner, by recursively splitting cells to minimize an empirical error criterion. Such greedy forests were introduced by Breiman [5, 6] and have shown excellent performance in many applications. One of his most popular classifiers is an averaging classifier, $\overline{g}_n$, based on a randomized tree classifier $g_n(x, Z)$ defined as follows. The algorithm has a parameter $1 \le v < d$ which is a positive integer. The feature space $\mathbb{R}^d$ is partitioned recursively to form a tree partition. The root of the random tree is $\mathbb{R}^d$. At each step of the construction of the tree, a leaf is chosen uniformly at random. v variables are selected uniformly at random from the d candidates $x^{(1)}, \ldots, x^{(d)}$. A split is selected along one of these v variables to minimize the number of misclassified training points if a majority vote is used in each cell. The procedure is repeated until every cell contains exactly one training point X_i. (This is always possible if the distribution of X has non-atomic marginals.)

In some versions of Breiman's algorithm, a bootstrap subsample of the training data is selected before the construction of each tree to increase the effect of randomization.

As observed by Lin and Jeon [14], Breiman's classifier is a weighted *layered nearest neighbor* classifier, that is, a classifier that takes a (weighted) majority vote among the layered nearest neighbors of the observation x. X_i is called a layered nearest neighbor of x if the rectangle defined by x and X_i as their opposing vertices does not contain any other data point X_j $(j \neq i)$. This property of Breiman's random forest classifier is a simple consequence of the fact that each tree is grown until every cell contains just one data point. Unfortunately, this simple property prevents the random tree classifier from being consistent for all distributions:

Proposition 6.1 *There exists a distribution of* (X, Y) *such that* X *has non-atomic marginals for which Breiman's random forest classifier is not consistent.*

PROOF. The proof works for any weighted layered nearest neighbor classifier. Let the distribution of X be uniform on the segment $\{x = (x^{(1)}, \ldots, x^{(d)}) : x^{(1)} = \cdots = x^{(d)}, x^{(1)} \in [0,1]\}$ and let the distribution of Y be such that $L^* \neq \{0, 1/2\}$. Then with probability one, X has only two layered nearest neighbors and the classification rule is not consistent. (Note that Problem 11.6 in [10] erroneously asks the reader to prove consistency of the (unweighted) layered nearest neighbor rule for any distribution with non-atomic marginals. As the example in this proof shows, the statement of the exercise is incorrect. Consistency of the layered nearest neighbor rule is true however, if the distribution of X has a density.) $\square$

One may also wonder whether Breiman's random forest classifier is consistent if instead of growing the tree down to cells with a single data point, one uses a different stopping rule, for example if one fixes the total number of cuts at k and let k grow slowly as in the examples of Section 3. The next two-dimensional example provides an indication that this is not necessarily the case. Consider the joint distribution of (X, Y) sketched in Figure 1. X has a uniform distribution on $[0,1] \times [0,1] \cup [1,2] \times [1,2] \cup [2,3] \times [2,3]$. Y is a function of X, that is $\eta(x) \in \{0,1\}$ and $L^* = 0$. The lower left square $[0,1] \times [0,1]$ is divided into countably infinitely many vertical stripes in which the stripes with $\eta(x) = 0$ and $\eta(x) = 1$ alternate. The upper right square

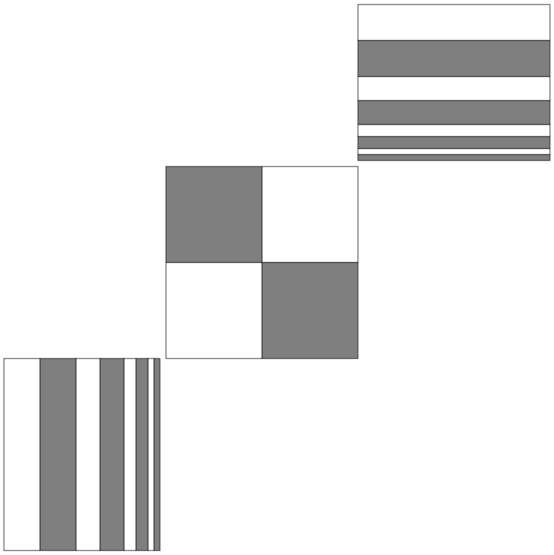

Figure 1: An example of a distribution for which greedy random forests are inconsistent. The distribution of X is uniform on the union of the three large squares. White areas represent the set where $\eta(x) = 0$ and on the grey regions $\eta(x) = 1$.

$[2, 3] \times [2, 3]$ is divided similarly into horizontal stripes. The middle rectangle $[1, 2] \times [1, 2]$ is a 2×2 checkerboard. Consider Breiman's random forest classifier with $v = 1$ (the only possible choice when $d = 2$).

For simplicity, consider the case when, instead of minimizing the empirical error, each tree is grown by minimizing the true probability of error at each split in each random tree. Then it is easy to see that no matter what the sequence of random selection of split directions is and no matter for how long each tree is grown, no tree will ever cut the middle rectangle and therefore the probability of error of the corresponding random forest classifier is at least 1/6.

It is not so clear what happens in this example if the successive cuts are made by minimizing the empirical error. Whether the middle square is ever cut will depend on the precise form of the stopping rule and the exact parameters of the distribution. The example is here to illustrate that consistency of greedily grown random forests is a delicate issue. Note however that if Breiman's original algorithm is used in this example (i.e., when all cells with more than one data point in it are split) then one obtains a consistent classification rule. If, on the other hand, horizontal or vertical cuts are selected to minimize the probability of error, and $k \to \infty$ in such a way that $k = O(n^{1/2-\varepsilon})$ for some $\varepsilon > 0$, then, as errors on the middle square are never more than about $O(1/\sqrt{n})$ (by the limit law for the Kolmogorov-Smirnov statistic), we see that thin strips of probability mass more than $1/\sqrt{n}$ are preferentially cut. By choosing the probability weights of the strips, one can easily see that we can construct more than $2k$ such strips. Thus, when $k = O(n^{1/2-\varepsilon})$, no consistency is possible on that example.

We note here that many versions of random forest classifiers build on random tree classifiers based on bootstrap subsampling. This is the case of Breiman's principal random forest classifier. Breiman suggests to take a random sample of size n drawn with replacement from the original data. While this may result in an improved behavior in some practical instances, it is easy to see that such a subsampling procedure does not vary the consistency property of any of the classifiers studied in this paper. For example, non-consistency of Breiman's random forest classifier with bootstrap resampling for the distribution considered in the proof of Proposition 6.1 follows from the fact that the two layered nearest neighbors on both sides are included in the bootstrap sample with a probability bounded away from zero and therefore the weight of these two points is too large, making consistency impossible.

In order to remedy the inconsistency of greedily grown tree classifiers, Devroye, Györfi, and Lugosi [10, Section 20.14] introduce a greedy tree classifier which, instead of cutting every cell along just one direction, cuts out a whole hyper-rectangle from a cell in a way to optimize the empirical error. The disadvantage of this method is that in each step, d parameters need to be optimized jointly and this may be computationally prohibitive if d is not very small. (The computational complexity of the method is $O(n^d)$.) However, we may use the methodology of random forests to define a computationally feasible consistent greedily grown random forest classifier.

In order to define the consistent greedy random forest, we first recall the tree classifier of Devroye, Györfi, and Lugosi [10, Section 20.14].

The space is partitioned into rectangles as shown in Figure 2.

A hyper-rectangle defines a split in a natural way. A partition is denoted by $\mathcal{P}$, and a decision on a set $A \in \mathcal{P}$ is by majority vote. We write $g_{\mathcal{P}}$ for such a rule:

$$g_{\mathcal{P}}(x) = \mathbb{1}_{\{\sum_{i:X_i \in A(x)} Y_i > \sum_{i:X_i \in A(x)} (1-Y_i)\}}$$

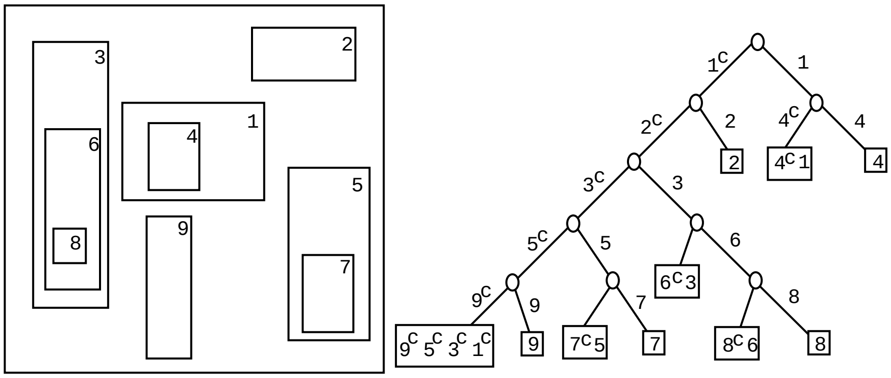

Figure 2: A tree based on partitioning the plane into rectangles. The right subtree of each internal node belongs to the inside of a rectangle, and the left subtree belongs to the complement of the same rectangle (i^c denotes the complement of i). Rectangles are not allowed to overlap.

where $A(x)$ denotes the cell of the partition containing x. Given a partition $\mathcal{P}$, a legal hyper-rectangle T is one for which $T \cap A = \emptyset$ or $T \subseteq A$ for all sets $A \in \mathcal{P}$. If we refine $\mathcal{P}$ by adding a legal rectangle T somewhere, then we obtain the partition $\mathcal{T}$. The decision $g_{\mathcal{T}}$ agrees with $g_{\mathcal{P}}$ except on the set $A \in \mathcal{P}$ that contains T.

Introduce the convenient notation

$$\begin{aligned} \nu_j(A) &= \mathbb{P}\{X \in A, Y = j\},\ j \in \{0,1\}, \\ \nu_{j,n}(A) &= \frac{1}{n}\sum_{i=1}^{n} I_{\{X_i \in A, Y_i = j\}},\ j \in \{0,1\}. \end{aligned}$$

The empirical error of $g_{\mathcal{P}}$ is

$$\widehat{L}_n(\mathcal{P}) \stackrel{\text{def}}{=} \sum_{R \in \mathcal{P}} \widehat{L}_n(R),$$

where

$$\widehat{L}_n(R) = \frac{1}{n}\sum_{i=1}^{n} I_{\{X_i \in R, g_{\mathcal{P}}(X_i) \neq Y_i\}} = \min(\nu_{0,n}(R), \nu_{1,n}(R)).$$

We may similarly define $\widehat{L}_n(\mathcal{T})$. Given a partition $\mathcal{P}$, the greedy classifier selects that legal rectangle T for which $\widehat{L}_n(\mathcal{T})$ is minimal (with any appropriate policy for breaking ties). Let R be the set of $\mathcal{P}$ containing T. Then the greedy classifier picks that T for which

$$\widehat{L}_n(T) + \widehat{L}_n(R - T) - \widehat{L}_n(R)$$

is minimal. Starting with the trivial partition $\mathcal{P}_0 = \{\mathbb{R}^d\}$, we repeat the previous step k times, leading thus to $k+1$ regions. The sequence of partitions is denoted by $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_k$.

Devroye, Györfi, and Lugosi [10, Theorem 20.9] establish consistency of this classifier. More precisely, it is shown that if X has non-atomic marginals, then the greedy classifier with $k \to \infty$ and $k = o\left(\sqrt{n/\log n}\right)$ is consistent.

Based on the greedy tree classifier, we may define a random forest classifier by considering its bagging version. More precisely, let $q_n \in [0,1]$ be a parameter and let $Z = Z(D_n)$ denote a random subsample of size binomial (n, q_n) of the training data (i.e., each pair (X_i, Y_i) is selected at random, without replacement, from D_n, with probability q_n) and let $g_n(x, Z)$ be the greedy tree classifier (as defined above) based on the training data $Z(D_n)$. Define the corresponding averaged classifier $\overline{g}_n$. We call $\overline{g}_n$ the *greedy random forest* classifier. Note that $\overline{g}_n$ is just the bagging version of the greedy tree classifier and therefore Theorem 5.1 applies:

Theorem 6.1 *The greedy random forest classifier is consistent whenever X has non-atomic marginals in $\mathbb{R}^d$, $nq_n \to \infty$, $k \to \infty$ and $k = o\left(\sqrt{nq_n/\log(nq_n)}\right)$ as $n \to \infty$.*

PROOF. This follows from Theorem 5.1 and the fact that the greedy tree classifier is consistent (see Theorem 20.9 of [10]). □

Observe that the computational complexity of building the randomized tree classifier $g_n(x, Z)$ is $O((nq_n)^d)$. Thus, the complexity of computing the voting classifier $g_n^{(m)}$ is $m(nq_n)^d$. If $q_n \ll 1$, this may be a significant speed-up compared to the complexity $O(n^d)$ of computing a single tree classifier using the full sample. Repeated subsampling and averaging may make up for the effect of decreased sample size.

Acknowledgments. We thank James Malley for stimulating discussions. We also thank three referees for valuable comments and insightful suggestions.

References

[1] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[3] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 24:801–849, 1998.

[4] L. Breiman. Some infinite theory for predictor ensembles. *Technical Report 577*, Statistics Department, UC Berkeley, 2000. http://www.stat.berkeley.edu/˜breiman .

[5] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[6] L. Breiman. Consistency for a simple model of random forests. *Technical Report 670*, Statistics Department, UC Berkeley, 2004. http://www.stat.berkeley.edu/˜breiman .

[7] A. Cutler and G. Zhao. Pert – Perfect random tree ensembles, *Computing Science and Statistics*, 33:490–497, 2001.

[8] L. Devroye. Applications of the theory of records in the study of random trees. *Acta Informatica*, 26:123–130, 1988.

[9] L. Devroye. A note on the height of binary search trees. *Journal of the ACM*, 33:489–498, 1986.

[10] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, New York, 1996.

[11] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40:139–157, 2000.

[12] T.G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli (Eds.), *First International Workshop on Multiple Classifier Systems*, Lecture Notes in Computer Science, pp. 1–15, Springer-Verlag, New York, 2000.

[13] Y. Freund and R. Shapire. Experiments with a new boosting algorithm. In L. Saitta (Ed.), *Machine Learning: Proceedings of the 13th International Conference*, pp. 148–156, Morgan Kaufmann, San Francisco, 1996.

[14] Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101:578–590, 2006.

[15] N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999, 2006.

[16] H.M. Mahmoud. *Evolution of Random Search Trees*. John Wiley, New York, 1992.

[17] C. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5:595–645, 1977.