Boosting Random Forests to Reduce Bias; One-Step Boosted Forest and Its Variance Estimate

Indrayudh Ghosal and Giles Hooker

Department of Statistics and Data Science, Cornell University, Ithaca, NY

ABSTRACT

In this article, we propose using the principle of boosting to reduce the bias of a random forest prediction in the regression setting. From the original random forest fit, we extract the residuals and then fit another random forest to these residuals. We call the sum of these two random forests a one-step boosted forest. We show with simulated and real data that the one-step boosted forest has a reduced bias compared to the original random forest. The article also provides a variance estimate of the one-step boosted forest by an extension of the infinitesimal Jackknife estimator. Using this variance estimate, we can construct prediction intervals for the boosted forest and we show that they have good coverage probabilities. Combining the bias reduction and the variance estimate, we show that the one-step boosted forest has a significant reduction in predictive mean squared error and thus an improvement in predictive performance. When applied on datasets from the UCI database, one-step boosted forest performs better than random forest and gradient boosting machine algorithms. Theoretically, we can also extend such a boosting process to more than one step and the same principles outlined in this article can be used to find variance estimates for such predictors. Such boosting will reduce bias even further but it risks over-fitting and also increases the computational burden. Supplementary materials for this article are available online.

1. Introduction

Ensemble methods have become one of the most successful and widely used methods in machine learning. Ensembles of trees, in particular, have the advantage of being computationally fast and of having few tuning parameters and requiring minimal human intervention (Breiman 2001; Friedman, Hastie, and Tibshirani 2001). These methods can be classified into two categories: “bagging-type” methods which reduce variance by combining trees that are obtained using identical randomized processes, and “boosting-type” methods which grow trees sequentially, one tree depending on the output of the previous. Recent work in Mentch and Hooker (2016) and Wager and Athey (2018) has demonstrated a central limit theorem for random forests—a bagging-type method—allowing for uncertainty quantification about its predictions. In this article, we leverage this to take a step toward boosting methods. We revisit a bias correction method for regression originally proposed in Breiman (2001) and further studied in Zhang and Lu (2012) and Xu (2013): we build two random forests, the second obtained from the residuals of the first, and then add them together. This represents one step of gradient boosting, as examined in Friedman (2001) for squared-error regression and we name the resulting algorithm one-step boosted forests. While the method is not novel, it has not been widely recognized within statistical learning despite near universal improvement in test set accuracy. In particular, it does better than either random forests or gradient boosting in experiments on data from the UCI repository (Lichman 2013). In this article, we build on recent work in Mentch and Hooker (2016) and Wager and Athey (2018) to develop variance estimates, show asymptotic normality and hence confidence intervals for the resulting predictions when ensemble methods are built using subsamples of the data.

Random forests (Breiman 2001) and other ensemble methods have proven to be one of the most popular machine learning methods (Fernández-Delgado et al. 2014). They proceed by generating trees from bootstrap or subsamples of the available data, potentially incorporating additional randomization within the tree building process. By averaging many trees built in this fashion, random forests achieve a considerable reduction in variance relative to any individual tree. More recently, this structure has been reinterpreted in the framework of $U$-statistics (van der Vaart 2000) allowing (Mentch and Hooker 2016; Wager and Athey 2018) to develop central limit theorems for the resulting predictions. Crucially, the variance of these predictions, and hence confidence intervals for them, can be calculated at no additional computational effort.

Despite their successes, random forests can suffer from bias when targeting complex signals. Since each tree is generated using an identical random process, they cannot be used to compensate each other. In particular, in a complex signal, each tree will target the same part of the signal, potentially leaving a bias that could have been effectively modeled by a random forest, if it were the only target of estimation. This is a result of the nonlinear use of data in the tree-building algorithm, as well as a tree partitioning scheme which quickly reduces the amount of data available to model local features. As an alternative, boosting...
methods build trees sequentially, allowing the current tree to correct for the biases of those that were constructed before it. Boosting was originally developed for classification in Freund and Schapire (1995) and Schapire and Freund (2012). In the regression setting, Friedman (2001) developed gradient boosting; in the context of least-squares regression, each tree is built to predict the residuals from the current model. To reduce over-fitting, gradient boosting introduces a shrinkage parameter and sets the number of trees (boosting steps) as a tuning parameter for predicting the residuals from the current model. To reduce over-fitting, gradient boosting introduces a shrinkage parameter and sets the number of trees (boosting steps) as a tuning parameter. By using the already-stable sets the number of trees (boosting steps) as a tuning parameter and and the boosting step) are the same or not. Ours empirical results and bootstrap bias-correction (Hooker and Mentch 2018). For the proofs of the results in this article (and a lot of the derivations as well), we will refer to specific sections in the appendix. These appendix sections can be found in the supplementary materials for this article.

2. Defining One-Step Boosted Forests

We first set some notation used throughout this article. Let \( Z_{[n]}^{(0)} = (Z_1^{(0)}, Z_2^{(0)}, \ldots, Z_n^{(0)}) \) denote the dataset, where \( Z_i^{(0)} = (Y_i, X_i) \) and \( [n] \) is the set \( \{1, \ldots, n\} \). Further note that random forests were originally described with bootstrap samples, but implementations also allow subsampling and using subsamples of size \( k < n \) has been important in making theoretical progress. In this article, we will refer to these types of random forests as usual ones, unless specified otherwise.

We now formally define the method of creating the one-step boosted forest. We build two forests, the first being the usual random forest—if there are \( B \) trees in the forest then for each of them we select \( k < n \) datapoints at random without replacement and denote by \( T(x; Z_i^{(0)}) \) the estimate from that tree for any test point \( x \), where \( I \) is the set of \( k \) indices selected (i.e., \( |I| = k \)). Let \( I_1^{(0)}, \ldots, I_B^{(0)} \) be the indices selected for each of the trees in the forest, with each of the \( I_i^{(0)} \) having \( k \) elements. Then the estimate after the first stage will be

\[
\hat{F}^{(0)}(x) = \frac{1}{B} \sum_{b=1}^{B} T(x; Z_i^{(0)}),
\]

(2.1)

Since selection of the subsets \( I \) are random, we can assign random weights \( w_i^{(0)} \) to each of the \( \binom{n}{k} \) possible subsets. Each \( w_i^{(0)} \) will be a binary random variable taking the value \( \binom{n}{k}/B \) with probability \( B/\binom{n}{k} \) and the value 0 with probability \( 1 - B/\binom{n}{k} \). The weights \( w_i^{(0)} \) are then iid random variables with \( \mathbb{E}(w_i^{(0)}) = 1 \) and \( c := \mathbb{E}(w_i^{(0)}) = \binom{n}{k}/B - 1 \). Thus, the formula for the random forest in (2.1) can be rewritten as

\[
\hat{F}^{(0)}(x) = \frac{1}{\binom{n}{k}} \sum_{I \subseteq [n]: |I| = k} w_i^{(0)} T(x; Z_i^{(0)}).
\]

(2.2)

Note that this approach is not exactly the same as taking \( B \) subsets at random. Since \( w_i^{(0)} \) are iid the total number of trees has an expected value of \( B \) and is not always exactly equal to \( B \). The difference between these two selection procedures has been shown to be ignorable in Appendix C.

Once we have obtained \( \hat{F}^{(0)} \) we can derive the residuals \( e_i = Y_i - \hat{F}^{(0)}(X_i) \) and construct a new dataset \( Z_{[n]}^{(1)} = (Z_1^{(1)}, \ldots, Z_n^{(1)}) \), where \( Z_i^{(1)} = (e_i, X_i) \). We can repeat the same method as above on this dataset with weights \( w_i^{(1)} \) and get a new estimate for the second stage

\[
\hat{F}^{(1)}(x) = \frac{1}{\binom{n}{k}} \sum_{I \subseteq [n]: |I| = k} w_i^{(1)} T(x; Z_i^{(1)}).
\]

(2.3)

This is the second forest and the first boosted step in our algorithm. Our final estimate is the sum of the first and second stage estimates, (2.2) and (2.3), given by

\[
\hat{F}(x) = \hat{F}^{(0)}(x) + \hat{F}^{(1)}(x).
\]

(2.4)
For this construction, we can take \(w_j^{(1)}\) to be the same as or independent from \(w_j^{(0)}\), that is, choosing the same subsets in the second stage as the first stage or independent ones. Based on that choice our estimate will also change. We thus have to consider two variants in this article:

- If \(w_j^{(1)} = w_j^{(0)}\) then our estimate (2.4) is the one-step boosted forest with same subsamples.
- If \(w_j^{(1)} \perp w_j^{(0)}\) then our estimate (2.4) is the one-step boosted forest with independent subsamples.

Algorithms 1 and 2 give details of these two methods along with their variance estimates discussed in Section 3.4. We will compare the performance of these variants in Sections 5.1 and 5.2. In the next sections, we shall try to quantify the variability and provide theoretical guarantees for our estimate \(\hat{F}\) in (2.4).

### 3. Analyzing One-Step Boosted Forests

In this section, we will show that the one-step boosted forest can be expressed as a weighted \(U\)-statistic (with random weights). Using that fact we calculate the theoretical variance and also provide a central limit theorem for the one-step boosted forest. Then, we also provide a variance estimate for the one-step boosted forest and finally present the formal algorithm combining our findings. Before that we need a crucial assumption.

#### 3.1. \(U\)-Statistics

We first familiarize ourselves with \(U\)-statistics. If \(h(z_1, \ldots, z_k)\) is a symmetric function then the \(U\)-statistic \(U\) with kernel \(h\) is defined by

\[
U(Z_1, \ldots, Z_n) = \frac{1}{\binom{n}{k}} \sum_{|I|=k} h(Z_I)
\]

\[
= \frac{1}{\binom{n}{k}} \sum_{|I|=k} h(Z_{i_1}, \ldots, Z_{i_k}).
\]

Hoeffding (1948) showed further that the \(U\)-statistic is asymptotically normal with variance \(\frac{k^2}{n}\xi_{\ref{k,\ref}}\), where

\[
\xi_{\ref{k,\ref}} = \text{cov}(h(Z_1, \ldots, Z_c, Z_{c+1}, \ldots, Z_k), h(Z_1, \ldots, Z_c, Z_{c+1}, \ldots, Z_k))
\]

\[
= \text{var}(E(h(Z_1, \ldots, Z_k)|Z_1 = z_1, \ldots, Z_c = z_c)).
\]

#### 3.2. A Pivotal Assumption

It is easily seen that the first stage forest in \(\hat{F}^{(0)}(x)\) (2.2) can be thought of as a weighted complete \(U\)-statistic. But then note that \(Z_i^{(1)}\) defined in Section 2 actually depends on the whole of the previous dataset \(Z_{[n]}^{(0)}\) and so does \(T(x; Z_i^{(1)})\) regardless of the subset \(I\). Hence, \(T(x; Z_i^{(1)})\) is not a valid kernel for a \(U\)-statistic which makes \(\hat{F}^{(1)}(x)\) not a valid \(U\)-statistic. However, if we replace \(\hat{F}^{(1)}(x)\) in (2.3) with \(\hat{F}^{(1)}(x)\) trained in the same manner but based on data with "noise-free" residuals:

\[
\hat{Z}_i^{(1)}(x) = (Y_i - \hat{E} [\hat{F}^{(0)}(X_i)], X_i)
\]

then it is easily seen that \(\hat{F}^{(1)}(x)\) is a \(U\)-statistic. This will help in further analysis of boosted forest defined in (2.4) since it can be expressed as the sum of two (weighted) \(U\)-statistics. Note that \(\hat{F}^{(1)}(x)\) does not inherit variability from \(\hat{F}^{(0)}(x)\), although the two will still be correlated. This approximation leads to significantly simplified analysis. Throughout this section and the corresponding proofs in the appendix, we will assume that the following condition holds true.

**Condition 1.** Let

\[
\hat{F}^{(1)}(x) = \frac{1}{\binom{n}{k}} \sum_{|I|=k} w_j^{(1)} T(x; Z_I^{(1)})
\]

then

\[
\frac{\hat{F}^{(1)}(x) - \hat{F}^{(1)}(x)}{\sqrt{\text{var}(\hat{F}^{(1)}(x))}} \overset{p}{\to} 0.
\]

That is, the effect of the variance in \(\hat{F}^{(0)}(x)\) on \(e_i\) is negligible in \(\hat{F}^{(1)}(x)\). Throughout, our theoretical analysis will apply to \(\hat{F}(x) = \hat{F}^{(0)}(x) + \hat{F}^{(1)}(x)\). From Condition 1, it is seen that asymptotic variance for \(\hat{F}\) and \(\hat{F}\) will be the same, as will their asymptotic distributions.

This condition is crucial since it allows us to apply the theory of \(U\)-statistics (especially asymptotic normality) to \(\hat{F}(x)\) and be sure that it also works for \(\hat{F}(x)\). Whether this condition applies in practice depends on the details of the tree building procedure. The true response function \(F(x)\) will influence the tree structure in the first stage which in turn influences the residuals and then the tree structure of the second stage. All of these influences are difficult to quantify and we do not attempt a full analysis here. In practice Condition 1 may not hold for all possible tree/forest building procedures but in Appendix A.1 we consider an analogy with kernel methods as explored in Scornet (2016). There we show that this property holds for Nadaraya–Watson estimators if the bandwidth in the second stage is smaller than in the first: approximately corresponding to using deeper trees with smaller leaf sizes for \(\hat{F}^{(1)}(x)\) compared to \(\hat{F}^{(0)}(x)\). The property also holds if the two stages use different sets of covariates without any restriction on bandwidth relationships. As a heuristic our condition should hold when the trees that comprise \(\hat{F}^{(1)}(x)\) (estimating the bias) tend to have a different set of splits than those in \(\hat{F}^{(0)}(x)\) (the original estimate of the signal). This will, of course, depend on the specific algorithm employed to create the trees, as well as the properties of the underlying response function; and an analysis of such specific cases is beyond the scope of this article. An empirical evaluation of the regularity condition is looked into briefly in Appendix A.2.

As a further check on the validity of this assumption, we provide a detailed examination of the sample distribution of the predictions of the procedure in Section 5.1 where we see empirical confirmation of our results and good coverage of prediction intervals.

For the rest of this article, we shall try to restrict usage of the check (\(\hat{}\)) accent to reduce notational complexity. For theoretical calculations in Section 3.3, we shall use the notations from Section 2, for example, \(\hat{F}\) to denote \(\hat{F}\), \(Z\) to denote \(\hat{Z}\), etc. For
empirical procedures described in Sections 3.4 and 5 onward, we do not have access to $\hat{Z}$, etc. so $\hat{F}$ will denote the usual boosted forest defined in Section 2.

### 3.3. Asymptotic Normality

In this section, we will prove that the one-step boosted forest predictions are asymptotically normal. Later in Section 3.4, we will first condition over the weights to get a complete $U$-statistic. Then, we follow the result about complete $U$-statistics at the beginning of Section 3. Also note that $\xi_{i,k} = \var(h(Z_1, \ldots, Z_k))$ follows from the above definition of $\xi_{i,k}$. To make calculations simpler, we shall assume, without loss of generality, that the individual trees and thus the random forest on both stages have zero mean, that is, $E[T(x; Z_1)] = 0$ for all $I \subseteq [n]: |I| = k$. As discussed near the end of Section 2, there are two variants whose asymptotic variances can be calculated to be

$$V_{\text{same}}(x) := \frac{k^2}{n} \xi_{i,k} + \frac{c}{(l)} \zeta_{i,k},$$

$$V_{\text{ind}}(x) := \frac{k^2}{n} (\xi_{i,k}^{(0)} + \xi_{i,k}^{(1)} + 2 \xi_{i,k}^{(0,1)}) + \frac{c}{(l)} (\zeta_{i,k}^{(0)} + \zeta_{i,k}^{(1)}).$$

Here, the $\xi$ values for $V_{\text{same}}$ are based on the kernel being $T(x; Z_i^{(0)}) + T(x; Z_i^{(1)})$, that is, sum of the trees in the two stages rather than the individual trees for separate stages. The $\zeta^{(0)}$ and $\zeta^{(1)}$ values for $V_{\text{ind}}$ are based on the kernels being $T(x; Z_i^{(0)})$ and $T(x; Z_i^{(1)})$, respectively. Also here $\xi_{i,k}^{(0,1)}$ is the covariance term between the trees in the two stages, $T(x; Z_i^{(0)})$ and $T(x; Z_i^{(1)})$. The details of this calculation can be found in Appendix B.1.

Now it is shown in van der Vaart (2000) that $U$-statistics are asymptotically normal if $k$, in this case the subsample size for random forests, is constant as $n \to \infty$. If we assume that here then asymptotic normality of the boosted forest follows. But in practice as $n$ increases we want $k$ to increase as well. To allow for this, we need to assume the following Lindeberg–Feller type condition (initially presented as Condition 1 in Mentch and Hooker (2016)).

**Condition 2.** Assume that the dataset $Z_1, Z_2, \ldots \sim D_Z$ and let $T_1, T_2, \ldots, T_k$ be the tree kernel based on a subsample of size $k_n$. Define $T_{1,k_n} (x) = ET (x, Z_1, \ldots, Z_k)$. Then we assume that for all $\delta > 0$

$$\lim_{n \to \infty} \frac{1}{k_n} \mathbb{E} \left[ T_{1,k_n} (Z_1) \mathbb{1} \{ |T_{1,k_n} (Z_1)| > \delta k_n \} \right] = 0.$$  

A discussion of the situation in which this condition holds can be found in Mentch and Hooker (2016), in particular they argue that a sub-Gaussian response, combined with a limit to the influence individual data points on the values of a tree are sufficient for this to hold. Using the regularity Conditions 1 and 2, we can prove (in Appendix B.2) the following result.

**Theorem 1.** Assume that the dataset $Z_1^{(0)}, Z_2^{(0)}, \ldots \sim D_Z$ and that $\text{ET} (x; Z_1, \ldots, Z_k) \leq C < \infty$ for all $x, n$ and some constant $C$. Let $B_n$ be the number of trees in each step of the one-step boosted forest. Then as long as $k_n, B_n \to \infty$ such that $\frac{k_n}{n} \to 0$ and $\frac{n}{B_n} \to 0$ as $n \to \infty$ as well as $\lim_{n \to \infty} \frac{k_n}{\text{var} \left( \xi_{i,k} \right)} \neq 0$ we have

$$\frac{\hat{F}(x)}{\sigma_n(x)} \to \mathcal{N}(0, 1)$$

for some sequence $\sigma_n(x)$ given by $\sigma_n^2 (x) = V_{\text{same}}(x)$ from (3.1) for Variant I of the one-step boosted forest or $\sigma_n^2 (x) = V_{\text{ind}}(x)$ from (3.2) for Variant II of the one-step boosted forest.

### 3.4. Variance Estimation of the One-Step Boosted Forest

Now that we have the formulas for the theoretical variance of both variants we can go about finding estimates for them. We will find estimates for each term in (3.1) and (3.2) separately. In this section for simplicity, we define $T_{b,1}^{(b)}(x) = T(x; Z_{b,1}^{(0)})$ for $j = 0, 1$.

Note that the $\xi_{i,k}$ values can be estimated by just the variability of the individual trees in the forests, by adding them up for Variant I and separately for Variant II. As an example

$$\hat{\xi}_{i,k} = \text{var}_s (\tau_b^{(0)}(x)), \quad b = 1, \ldots, B.$$  

Here, $\text{var}_s$ is used to denote empirical variance by varying $b = 1, \ldots, B$. We shall use the same notation (the subscript $(s)$) for the rest of this article.

Now note that $\frac{k^2}{n} \hat{\xi}_{i,k}$ is the variance for a random forest when we consider all possible subsets of the dataset of size $k$, that is, a complete $U$-statistic. As an example if we define

$$\hat{F}(x) = \frac{1}{(l)} \sum_{I \subseteq [n]: |I| = k} T(x; Z_I^{(0)})$$

then $\frac{k^2}{n} \hat{\xi}_{i,k} = \text{var}(\hat{F}(x))$. From Theorem 1 (or Theorem 9) of Wager and Athey (2018), we know that an asymptotically consistent estimate is given by the infinitesimal Jackknife estimator, the formula for which is

$$\hat{\text{var}}_{ij}(\hat{F}(x)) = \frac{n}{\text{cov}_s \left( N_{i,b}, T_{b,1}^{(b)}(x) \right)}.$$  

as given by Theorem 1 of Efron (2014) where $\text{cov}_s$ indicates the empirical covariance over $b$. Here, $N_{i,b} = \mathbb{1} (i \in T_b^{(b)})$ is the indicator of whether the $i$th datapoint is included in the calculations for the $b$th tree. So we can estimate the variance for Variant I in (3.1) by using equivalent expressions to (3.5) for the first term and (3.3) for the second term:

$$\hat{V}_{\text{same}}(x) = \hat{V}_{lj} + (1/B) \ast \hat{\xi}_{i,k}$$

$$= \sum_{i=1}^{n} \text{cov}_s \left( N_{i,b}, T_{b,1}^{(b)}(x) + T_{b,1}^{(b)}(x) \right)^2$$

$$+ \frac{1}{B} \cdot \text{var}_s \left( T_{b,1}^{(b)}(x) + T_{b,1}^{(b)}(x) \right).$$  

(3.6)
In this formula, we used the approximation \( \frac{c}{(1)^2} \approx \frac{1}{(2)^2} \).

For the variance estimate of Variant II of the one-step boosted forest, we first need to find an estimate for \( \frac{\hat{\xi}_{1,k}}{n} \), the covariance between the first and second stages of our estimate in case of Variant II. It is reasonable to expect that we can have a two-sample analog of (3.5), that is, an infinitesimal jackknife estimate for the covariance given by

\[
\hat{\text{cov}}_{II}(\hat{F}^{(0)}(x), \hat{F}^{(1)}(x)) = \sum_{i=1}^{n} \text{cov}_{\text{ind}} \left[ N_{i,b}^{(0)}, T_{b}^{(0)}(x) \right] \cdot \text{cov}_{\text{ind}} \left[ N_{i,b}^{(1)}, T_{b}^{(1)}(x) \right],
\]

\( b = 1, \ldots, B. \) (3.7)

The consistency of this estimate is proved in Appendix B.4. We can now estimate the variance for Variant II in (3.2) by using equivalent expressions to (3.5) and (3.7) for the first term and (3.3) for the second term. We get

\[
\hat{V}_{\text{ind}}(x) = \hat{V}_{II} + (1/B) * \hat{\xi}_{1,k}
\]

\[
= \sum_{i=1}^{n} \left( \text{cov}_{\text{ind}} \left[ N_{i,b}^{(0)}, T_{b}^{(0)}(x) \right] + \text{cov}_{\text{ind}} \left[ N_{i,b}^{(1)}, T_{b}^{(1)}(x) \right] \right)^2
\]

\[
+ \frac{1}{B} \left( \text{var}_{\text{ind}} \left[ T_{b}^{(0)}(x) \right] + \text{var}_{\text{ind}} \left[ T_{b}^{(1)}(x) \right] \right)
\]

\[
= \sum_{i=1}^{n} \sum_{j=0}^{1} \text{cov}_{\text{ind}} \left[ N_{i,b}^{(j)}, T_{b}^{(j)}(x) \right]^2
\]

\[
+ \frac{1}{B} \sum_{j=0}^{1} \text{var}_{\text{ind}} \left[ T_{b}^{(j)}(x) \right],
\]

\( (3.8) \)

Note that we still use \( \frac{c}{(1)} \approx \frac{1}{(2)} \). Thus, we have found variance estimates for the one-step boosted forest (2.4) formalized in Section 2. We have the following result regarding these estimates.

**Theorem 2.** The variance estimates discussed above are consistent:

- \( V_{\text{same}}(x) \(3.1) \) is consistently estimated by \( \hat{V}_{\text{same}}(x) \) in (3.6).
- \( V_{\text{ind}}(x) \) in (3.2) is consistently estimated by \( \hat{V}_{\text{ind}}(x) \) in (3.8).

The proof follows directly from Lemma 3 in Appendix B.4 and the fact that the sample variance is a consistent estimator of the population variance.

### 3.5. The Complete One-Step Boosted Forest Algorithm

We present below complete algorithms for construction and variance estimation for Variant I (Algorithm 1) and Variant II (Algorithm 2) of the one-step boosted forest. The performance of both variants are compared among themselves and to other standard algorithms in Sections 5.1 and 5.2.

#### Algorithm 1: One-step boosted forest—Variant I (same subsets in both stages)

**Input:** The data \( (Z_{i}^{(0)} = (Y_{i}, X_{i})^{n}_{i=1}) \), the tree function \( T \), the number of trees in the forest \( B \), the subsample size for each tree \( k \), and the test point \( x \).

**for** \( b = 1 \) **to** \( B \) **do**

- Choose \( I_{b}^{(0)} \subseteq \{1, \ldots, B\} \) randomly such that \( |I_{b}^{(0)}| = k \).
- Calculate: \( T_{b}^{(0)}(x) = T \left( x; Z_{i}^{(0)} \right) \) and \( N_{b}^{(0)} = \{i \in I_{b}^{(0)} \} \).

**end**

**Obtain:** The first stage estimate \( \hat{F}^{(0)}(x) = \hat{F}_{b}^{(0)}(x) \).

**Calculate:** \( T_{b}^{(1)}(x) = T \left( x; Z_{i}^{(1)} \right) \).

**end**

**Obtain:** The second stage estimate \( \hat{F}^{(1)}(x) = \hat{F}_{b}^{(1)}(x) \).

**Calculate:** The first term of the variance estimate \( \hat{V}_{II} = \sum_{i=1}^{n} \text{cov}_{\text{ind}} \left[ N_{i,b}^{(0)}, T_{b}^{(0)}(x) + T_{b}^{(1)}(x) \right]^2 \).

**Calculate:** The (unscaled) second term of the variance estimate \( \hat{\xi}_{b,k} = \text{var}_{\text{ind}} \left[ T_{b}^{(0)}(x) + T_{b}^{(1)}(x) \right] \).

**Output:** The one-step boosted forest estimate at the test point \( x \) given by \( \hat{F}(x) = \hat{F}^{(0)}(x) + \hat{F}^{(1)}(x) \) and the variance estimate given by \( \hat{V}_{\text{same}}(x) = \hat{V}_{II} + (1/B) * \hat{\xi}_{1,k} \).

#### 4. Further Discussions

#### 4.1. Comparison With Prior Results in the Literature

Theorem 1 is an extension and combination of previous work. Lemma 2 (or Theorem 1(i)) of Mentch and Hooker (2016) and Theorem 1 of Wager and Athey (2018) show that \( \hat{F}^{(0)}(x) \) has an asymptotically (possibly biased in the former article) normal distribution; the former listing a variance of \( \frac{\hat{\xi}_{1,k}}{n} \), while the latter employs the infinitesimal jackknife estimator as a consistent variance estimate. However, the two papers use different assumptions to demonstrate normality. We have used the conditions for the former result, but note that inspection of their proof of Theorem 1 (see Mentch and Hooker 2016, p. 29) allows a replacement of their conditions—\( k_{n} = o(\sqrt{n}) \) and \( \lim_{n \to \infty} k_{n} \neq 0 \)—with those we give above; see Peng, Coleman, and Mentch (2019). Wager and Athey (2018) required \( k_{n}(\log n)^{d} \to 0 \) along with some conditions on the tree building process, but demonstrated that the bias in the resulting estimators is asymptotically ignorable. Either set of conditions could be employed within our result.
Since we had assumed that the tree function $T$ has zero mean, our central limit theorem is actually centered on $E[\hat{F}(x)]$, but we could add the honesty assumption from Wager and Athey (2018) (detailed in Lemma 2 and Theorem 3 of that paper) to change the center to be the target function $F(x)$. Note in that case the second boosting stage $\hat{F}^{(1)}(x)$ is asymptotically estimating $0$. Now boosting is supposed to reduce the bias $E[\hat{F}(x)] - F(x)$, and the high empirical values of performance improvement (due to low values of MSE) in Section 5.1 suggests that in this case the honesty assumption might not be necessary in practice.

We can also get a similar result about the joint distribution of each stage of Variant II of the boosted forest, under the extra condition that $\lim_{n \to \infty} \mathbb{E}[(Z_{i,k}^{(1)}/\xi_{i,k})] \notin [0, \infty]$. This will be a more general result compared to the main theorem above, and we can use any linear combination of the boosting steps to arrive at the final estimate rather than just adding them. This result and its proof is in Appendix B.3.

Our variance estimates discussed in Section 3.4 borrow from the infinitesimal jackknife estimate used in Wager and Athey (2018) where there is an assumption that the number of trees (i.e., the number of times we subsample) $B$ be so large as to negate Monte Carlo effects, that is, large $B$ leads to $\text{Var}_T$ being close to $\text{Var}_T = \text{Var}_T$. However, our theoretical variance formulas in (3.1) and (3.2) accounts for this with an additional term. We thus use the infinitesimal jackknife approach to only estimate $\text{Var}_T$ (the first term in our formulas for the variance) and add an estimate for the second term. We also remove the finite sample correction factor $n/(n+1)$ discussed in Wager and Athey (2018) and the additive correction term in Wager, Hastie, and Efron (2014). Our simulation results below demonstrate an upward bias of the infinitesimal jackknife estimator, particularly for small $B$; we have found that standard correction terms in Wager, Hastie, and Efron (2014) often result in negative variance estimates, see Zhou, Menth, and Hooker (2019) for a discussion.

Our boosting method corresponds to the method BC3 in Zhang and Lu (2012); other bias correction methods in that paper also incorporate the response within a correction term. When the correction is given by a random forest (BC1 and BC3 in Zhang and Lu (2012)) our central limit theorem continues to hold. When correcting for response bias via smoothing splines (method BC2), the same conditions would require an analysis of the variance due to both random forests and splines.

The boosted forest algorithm is unlike the bootstrap bias correction method in Hooker and Menth (2018), where bias was directly estimated via the bootstrap, but which did not include a variance estimate for the bias corrected random forest. The algorithm in Hooker and Menth (2018) is akin to a two-sample $U$-statistic but the dependency within the data and the residuals (on which the bootstrap is done) makes it harder to obtain a variance estimate via the infinitesimal jackknife. However, we speculate that the algorithms in Menth and Hooker (2016) can be used to find an estimate of the variance of the bias correction algorithm.

4.2. Extensions: More Than One Boosting Step

We could continue with the boosting process and reduce the bias even further. For example, if we boosted once more we would define $Z_{i,k}^{(2)} = Y_i - \hat{F}^{(0)}(X_i) - \hat{F}^{(1)}(X_i), X_i$ to be the dataset for the third stage output $\hat{F}^{(2)}(x)$. Our final output would be the 2-step boosted forest given by

$$\hat{F}(x) = \hat{F}^{(0)}(x) + \hat{F}^{(1)}(x) + \hat{F}^{(2)}(x).$$

Its variance would depend of which variant of the original algorithm we use. If we used the same subsets to generate all three random forests then the variance would be consistently estimated by

$$\hat{V}_\text{same}(x) = \sum_{i=1}^{n} \text{cov}_x \left[ N_{i,b}^{(0)}, \sum_{j=0}^{2} \hat{T}_b^{(j)}(x) \right]^2 + \frac{1}{B} \cdot \text{var}_x \left[ \sum_{j=0}^{2} \hat{T}_b^{(j)}(x) \right].$$

---

**Algorithm 2: One-step boosted forest—Variant II (independent subsets in 2 stages).**

**Input:** The data $(Z_i^{(0)} = (Y_i, X_i))_{i=1}^{n}$, the tree function $T$, the number of trees in the forest $B$, the subsample size for each tree $k$, and the test point $x$.

**for** $b = 1$ **to** $B$ **do**

Choose $i_b^{(0)} \subseteq [n]$ randomly such that $|i_b^{(0)}| = k$.

**Calculate:** $T_b^{(0)}(x) = T(x; Z_i^{(0)})$ and $N_{i,b}^{(0)} = 1\{i \in I_b^{(0)}\}$.

**end**

**Obtain:** The first stage estimate $\hat{F}^{(0)}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b^{(0)}(x)$.

Calculate residuals $e_i = Y_i - \hat{F}^{(0)}(X_i)$ and new dataset $(Z_i^{(1)} = (e_i, X_i))_{i=1}^{n}$.

**for** $b = 1$ **to** $B$ **do**

Choose $i_b^{(1)} \subseteq [n]$ randomly such that $|i_b^{(1)}| = k$, that is, an independent copy of the first stage subset.

**Calculate:** $T_b^{(1)}(x) = T(x; Z_i^{(1)})$ and $N_{i,b}^{(1)} = 1\{i \in I_b^{(1)}\}$.

**end**

**Obtain:** The second stage estimate $\hat{F}^{(1)}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b^{(1)}(x)$.

**Calculate:** The first term of the variance estimate $\hat{V}_{I} = \sum_{i=1}^{n} \left( \text{cov}_x \left[ N_{i,b}^{(0)}, T_b^{(0)}(x) \right] \right)^2$ + $\text{cov}_x \left[ N_{i,b}^{(1)}, T_b^{(1)}(x) \right]^2$.

**Calculate:** The (unscaled) second term of the variance estimate $\hat{\xi}_{i,k} = \text{var}_x \left[ T_b^{(0)}(x) \right] + \text{var}_x \left[ T_b^{(1)}(x) \right]$.

**Output:** The one-step boosted forest estimate at the test point $x$ given by $\hat{F}(x) = \hat{F}^{(0)}(x) + \hat{F}^{(1)}(x)$ and the variance estimate given by $\hat{V}_{\text{ind}}(x) = \hat{V}_{I} + (1/B) \cdot \hat{\xi}_{i,k}$.
We could also use subsets independently generated for all three stages and then the variance estimate would be given by
\[
\hat{V}_{\text{ind}}(x) = \sum_{i=1}^{n} \left( \sum_{j=0}^{2} \text{cov}_a \left[ N_{i,k,b}, T_b^{(j)}(x) \right] \right)^2 + \frac{1}{B} \sum_{j=0}^{2} \left( \text{var}_a \left[ T_b^{(j)}(x) \right] \right).
\]

We could also tweak the process and take independent subsets in the first two stages and then the same in the last stages, that is, in terms of notation in Section 2 the weights could be \(w_1^{(0)}, w_2^{(1)}, w_2^{(2)}\), respectively, for the 3 stages. We could actually have two more combinations, namely \(w_1^{(0)}, w_1^{(0)}, w_2^{(2)}\) and \(w_1^{(0)}, w_1^{(1)}, w_2^{(1)}\). Thus, there are 5 variants of the 2-step boosted forest based on these combinations and for each combination we can easily find out the variance estimates using the principles outlined in Section 3.4.

For an \(M\)-step boosted forest we can easily see that the number of variants is given by \(a_{M+1}\), where
\[
a_n = \sum_{k=1}^{n} a_{n,k} \quad \text{with} \quad a_{n,k} = ka_{n-1,k} + a_{n-1,k-1} \quad \forall \ n > k \quad \text{and} \quad a_{k,k} = a_{n,1} = 1 \quad \forall \ n, k.
\]

For each of these variants, the final estimate will simply be the sum of all the boosting steps and the variance can be found by following similar steps as outlined in Section 3.4.

5. Empirical Studies for One-Step Boosted Forest

We shall focus on performances of our algorithm in this section. Our implementation differs slightly from the theory above in the following ways.

- In Sections 5.1 and 5.2, we construct random forests with \(B\) trees in them but in the calculations above we assumed that the trees were all randomly weighted such that the random weights add up to an expected value of \(B\), not always exactly equal to \(B\). At the beginning of Appendix C, we have shown in detail that the difference between these two approaches are asymptotically negligible.

- We will also consider the out-of-bag predictions (and hence residuals) in our implementation for calculating \(\hat{F}(0)(x)\) instead of the simple average (inbag prediction) of all the trees in the forest. This is also a form of assigning a weight to the trees in the forest (the weights are not completely random but fixed given the dataset and the randomly selected subsets) but should also asymptotically give us the same results.

Using out-of-bag residuals could be thought of as akin to the honesty condition in Wager and Athey (2018) for the second stage of the boosted forest since instead of using all the data for the residuals we use the data that was not used in construction of that particular tree. In fact, because of this we expect the out-of-bag approach to have more variability and hence the ratios \(\frac{\hat{V}_i}{\hat{V}(0)}\) in Section 5.1 should be higher than the expected value of 1. We shall also get slightly more conservative prediction intervals in Section 5.2 which will lead to higher coverage than the expected value of 95%.

In Appendix C.4, we compare our use of out-of-bag residuals with other boosting formulations where we find this version provides better predictive performance.

5.1. Performance on Simulated Datasets

Here, we compare the performance of the Algorithms 1 and 2 with different simulated datasets. The base learner which we will compare it against is just the simple random forest, that is, without any boosting. We will also test the accuracy of our variance estimate by comparing it with the actual variance of the estimates.

Our model is \(Y = \sum_{i=1}^{5} X_i + \epsilon\), where \(X \sim U([-1, 1])\) and \(\epsilon \sim N(0, 1)\). We fix the points in \([-1, 1]^{15}\) where we will make our predictions, given by
\[
p_1 = 0_{15}, \ p_2 = \left( \frac{1}{3} \ 0^\top_{14} \right)^\top, \ p_3 = \frac{1}{3\sqrt{15}} \ast 1_{15}, \ p_4 = 2p_3, \ p_5 = 3p_3.
\]

We chose \(p_3, p_4, \) and \(p_5\) to have an idea of how distance of a test point from the the “center” of the dataset affects the performance of our algorithm.

Out simulation runs for a 1000 iterations—in each of them we generate a dataset of size \(n = 500\) and train a random forest and one-step boosted forests (both variants) with it with subsample size \(k = 100\) and the number of trees \(B\) in \((5000, 10,000, 15,000)\). For each of these settings we can find a prediction estimate at each of the \(p_i\)’s given by \(\hat{F}_{ij} = \hat{Y}_{ij}\) and also corresponding variance estimates given by \(\hat{V}_{ij}\), for \(i = 1, \ldots, 5, j = 1, \ldots, 1000\).

We test the performance of our algorithm by the following metrics. The corresponding figures are in Table 1.

- The average bias is given by \(\text{Bias} = \frac{1}{1000} \sum_{i=1}^{1000} \hat{F}_{ij} - F(p_i)\), where \(F(x_1, \ldots, x_{15}) = \sum_{i=1}^{5} x_i\). We see that the bias is already fairly low at the origin and the boosted forest does not change that substantially. But as the target points moves away from the origin the improvement in bias becomes very obvious.

- The variance estimate for each algorithm is given by \(\hat{V}_{ij} = \frac{1}{1000} \sum_{i=1}^{1000} \hat{V}_{ij}\). For each \(p_i\), the typical order for the variance estimate is BFv1 > BFv2 > RF but the value also decreases with \(B\) as expected.

- The ratio \(\frac{\hat{V}_i}{\hat{V}(0)}\) shows the consistency of the infinitesimal jackknife estimate. A value of 1 is ideal and we see that the empirical results are not far away from 1. In fact, the ratio decreases as \(B\) gets larger as should be expected, see Zhou, Mentch, and Hooper (2019).

- K.S. gives us the Kolmogorov–Smirnov statistics testing the hypothesis that the predictions should be normal with the mean given by the sample mean and the variability given by the variance estimate. Since smaller values of this statistic are better, thus we can see marked improvement for both variants of the boosted forest as compared to the base random forest. Improvements also get better as the target points move away
from the origin and also as the number of trees increases—the second fact being consistent with the assumption $\frac{1}{n} \to 0$ mentioned in Theorem 1. Overall these numbers are fairly low and show consistency with Theorem 1 although the values are expected to be a bit high because we use variance estimate (consistent by 2) instead of the (unknown) actual variance.

- Constructing 95% confidence intervals $\hat{F}_{ij} \pm \Phi^{-1}(0.975)\sqrt{\hat{V}_{ij}}$ we can check if $F(p_j)$ falls inside that interval for $j = 1, \ldots, 1000$. C.C. denotes this coverage probability which we should expect to be close to 95% for random forests and boosted forests. But we see that due to high bias values for random forests become worse as we move away from the origin. But boosted forests correct for the bias and thus the coverage is always at least 95%. We also get more precision in our variance estimate as the number of trees increases and thus the coverage values also become less overinflated.

- We also test performance improvement (P.I.) which is defined as follows: Fixing $i \in [5]$ we obtain the estimated prediction MSE given by $\text{MSE}_{ij} = (\hat{F}_{ij} - F(p_j))^2$ for $j = 1, \ldots, 1000$. Define

$$\text{Performance improvement (P.I.)} = 1 - \frac{\sum_{j=1}^{1000} \text{MSE}_{ij} \text{for BF}}{\sum_{j=1}^{1000} \text{MSE}_{ij} \text{for RF}}.$$  

Since we are comparing against random forests their own P.I. is 0. As for boosted forests P.I. is actually worse for the points near the origin as boosting does not affect the bias too much but increases the variance quite a lot in comparison. But as we move further away from the origin our algorithm becomes effective at reducing bias compared to the increase in variance and thus we obtain significant improvements.

This also gets better with an increase in number of trees since the variance estimates become more precise. Finally, note that the two variants perform almost the same but later on in the paper we see Variant II performing better than Variant I.

We conclude that the boosted forest algorithms give better predictions than the usual random forest algorithm on simulated datasets and that Variant II is more powerful than Variant I.

In Appendix C.2, we have also done further simulations for datasets with different levels noise than the one in this section and observed that the one-step boosted forest algorithm is resilient to a moderate amount of noise but as expected, increasing noise reduces its performance. Furthermore we have also looked into the performance of the method for a nonlinear signal in Appendix C.3. And finally note that we used out-of-bag residuals to construct the boosted forests in this section. We have compared it to boosted forests constructed with inbag residuals and also using bootstrapped subsamples in Appendix C.4 and we have seen that the out-of-bag method is preferable.

### 5.2. Performance on Real Datasets

We applied the boosted forest algorithms (both variants) to 11 datasets in the UCI database (Lichman 2013) which have a regression setting and compared its performance to the gradient boosting machine algorithm (Friedman 2001 and the R package GBM) and the bias correction algorithm in Hooker and Mentch (2018). The results are reported in Table 2.

For each dataset, we use 10-fold cross-validation to calculate the prediction mistake and then record the improvement (in percentages) compared to the basic random forest algorithm.
Table 2. Comparison of the two variants of the boosted forest with the GBM algorithm (Friedman 2001 and the GBM R package) and the bias correction algorithm in Hooker and Mentch (2018).

| Dataset                  | Basics | Improvement | RF | BFv1 | BFv2 |
|--------------------------|--------|-------------|----|------|------|
|                          | n      | d           | k | var | GBM  | BC  | PI length | PI coverage |
| Yacht-hydrodynamics      | 300    | 6           | 6 | 229.55 | 92.56 | 68.77 | 14.25 | 15.43 | 15.22 |
|                         | 60     |             |   |       | 81.64 | 82.04 | 91.33 | 99.33 | 99.00 |
| BikeSharing-hour         | 17,370 | 14          | 2000 | 2.21 | 20.37 | 64.61 | 1.48 | 1.51 | 1.50 |
|                         | 2000   |             |   |       | 73.64 | 73.99 | 100.00 | 100.00 | 100.00 |
| Concrete                 | 1030   | 8           | 200 | 279.08 | 29.98 | 42.95 | 25.29 | 27.83 | 27.33 |
|                         | 200    |             |   |       | 51.61 | 52.20 | 96.02 | 99.03 | 98.83 |
| Airfoil                  | 1500   | 5           | 300 | 46.95 | -35.40 | 36.72 | 14.21 | 15.54 | 15.25 |
|                         | 300    |             |   |       | 43.92 | 43.65 | 94.33 | 99.27 | 99.07 |
| Boston-housing           | 500    | 13.00       | 150 | 0.17 | -34.27 | 18.25 | 0.61 | 0.65 | 0.64 |
|                         | 150    |             |   |       | 26.12 | 26.22 | 95.80 | 97.40 | 97.00 |
| Auto-mpg                 | 390    | 7.00        | 50 | 61.03 | 7.00 | 16.13 | 11.40 | 11.94 | 11.86 |
|                         | 50     |             |   |       | 21.21 | 20.79 | 93.85 | 96.41 | 95.90 |
| Wine-quality-white       | 4890   | 11.00       | 1000 | 0.79 | -22.82 | 8.77 | 3.34 | 4.06 | 3.96 |
|                         | 1000   |             |   |       | 11.57 | 11.42 | 98.45 | 99.51 | 99.45 |
| Parkinsons               | 5870   | 16.00       | 1000 | 66.14 | -25.06 | 7.58 | 35.63 | 43.23 | 42.35 |
|                         | 1000   |             |   |       | 8.25 | 8.09 | 99.71 | 99.97 | 99.95 |
| Wine-quality-red         | 1590   | 11.00       | 300 | 0.65 | -9.12 | 5.42 | 2.59 | 2.90 | 2.85 |
|                         | 300    |             |   |       | 7.33 | 7.45 | 95.91 | 97.80 | 97.48 |
| SkillCraft               | 3330   | 18.00       | 600 | 2.10 | 2.30 | 2.06 | 4.37 | 5.05 | 5.00 |
|                         | 600    |             |   |       | 4.23 | 4.30 | 98.74 | 99.61 | 99.52 |
| Communities              | 1990   | 96.00       | 400 | 0.05 | -2.40 | 1.68 | 0.62 | 0.69 | 0.68 |
|                         | 400    |             |   |       | 2.93 | 3.05 | 96.88 | 98.19 | 98.09 |

NOTE: We use shorthand for $n$ = number of datapoints, $d$ = number of features, $k$ = size of subset (not applicable for GBM), PI = prediction interval (5.2), RF = random forests, GBM = gradient boosting machine, BC = bootstrap bias correction, BFv1 = boosted forest variant 1 (Algorithm 1) and similarly for BFv2 (Algorithm 2). The improvement and PI coverage figures are in percentages.

Improvement is simply

$$1 - \frac{\text{prediction MSE for improved algorithm}}{\text{prediction MSE for random forest}}$$

For the GBM package in R, we used a 10% validation set to select the optimal tuning parameter (number of trees/boosting steps) out of a maximum of 1000. We did not use subsets but rather the full dataset to construct each tree in that ensemble. For random forests (randomForest package in R) and the two variants of our boosted forest algorithm we also used 1000 trees in the forests and randomly selected subsamples for each tree the size of which is given by the number $k$ in Table 2.

We can see that the GBM algorithm does not always have improvements over random forests and hence is not reliable as a good reference. Further the boosted forest algorithm has consistently registered greater improvement compared to the bias correction algorithm (Hooker and Mentch 2018). Variant II of our algorithm slightly outperforms Variant I in most cases.

We further validate our variance estimate by constructing test set confidence intervals. A 95% prediction interval for the datapoint $Z_i = (Y_i, X_i)$ is given by

$$\left( \hat{Y}_i - \Phi^{-1}(0.975)\sqrt{\hat{V}_i + \hat{V}_e}, \hat{Y}_i + \Phi^{-1}(0.975)\sqrt{\hat{V}_i + \hat{V}_e} \right),$$

where $\hat{Y}_i$ is the estimate, $\hat{V}_i$ is the variance estimate and $\hat{V}_e = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2$ is the residual MSE.

We see that when comparing boosted forests (for both variants) with the basic random forest algorithm the length of the prediction interval increases slightly but the prediction coverage (in percentages) increases significantly. The increment in the length of the prediction interval can be attributed to the increase in variability due to boosting. The same can also partially explain the increase in prediction coverage but the main reason for this increase is the reduction in bias due to boosting which leads to better “centering” of the prediction interval.

For these datasets as well we used the out-of-bag residuals to construct the random forest in the boosting step. A comparison with other approaches is in Appendix C.4.

6. Conclusion

Our algorithm, the one-step boosted forest fits a random forest on a given dataset and then fits another one on the residuals of the former. The sum of these two random forests gives us our estimate. This is a boosting method for random forests which, even if applied only once, provides performance improvements compared to base algorithm. Since it is a boosting method on a previously bagged estimate, the result should be a very stable algorithm.

The boosted forest also provides an estimate of its own variance which can be obtained with nothing more than the computation needed to calculate the boosted forest estimate itself. We have shown that our method leads to substantial reductions in bias (compared to a small increment in variance) in the regression setting and thus the predictive mean squared
error. More such boosting steps can be chained to get more improvements and we devised a fairly simple criteria for when to stop such further boosting. We have only tested our method against the random forest and gradient boosting algorithms but we expect similar results for other ensemble methods.

Following the discussion in Section 4.2, we could suggest some potential stopping rules for the boosting iteration. As in the original boosting framework in Friedman (2001), we expect that while boosting reduces the bias associated with random forests, it will incur greater variance as boosting progresses and a stopping rule can be based on test set error. Here, we can also make use of theoretical results for random forests where we observe that in the $M$-step boosted forest $\left( \hat{F}^{(m)}(x), \hat{F}^{(0)}(x), \ldots, \hat{F}^{(m-1)}(x) \right)$ will have an asymptotic normal distribution. We can thus test whether the expectation of the last step is significantly different from zero—that is, did the last step contribute to bias reduction? Tests of this form can be constructed by using a collection of test points $(x_1, \ldots, x_q)$ for which $\hat{F}^{(m)}(x_1), \ldots, \hat{F}^{(m)}(x_q)$ has a multivariate normal distribution which can be used to for a $\chi^2$ test; similar approaches to testing structure in random forests were described in Mentch and Hooker (2016, 2017) and Zhou, Zhou, and Hooker (2018).

The development of stopping criteria tests is left to future work. We note that commonly employed diagnostic tools for random forests remain applicable to their boosted version. In particular, variable importance measures based on split improvement scores (e.g., Friedman 2001) can be summed across the forests to arrive at combined importance scores. Permutation-based methods described in Breiman (2001) should be applied to the combined model. We note that both types of variable importance are potentially misleading; see Strobl et al. (2007, 2008), Ishwaran (2007), Hooker (2007), and Hooker and Mentch (2019) and therefore do not recommend their use, although Zhou and Hooker (2019) produced a potential fix to split improvement scores.

Supplementary Materials
The supplementary material for this paper contains proofs of the results stated in the main paper and details of some calculations. We also provide further theoretical motivations and empirical explorations of ideas presented in the main paper. All of the appendix sections referred to here also can be found in the supplementary material.

Funding
The authors gratefully acknowledge funding from NSF grants DMS-1712554 and DEB-1933497.

References
Breiman, L. (2001), “Random Forests,” Machine Learning, 45, 5–32. [493,502]
Efron, B. (2014), “Estimation and Accuracy After Model Selection,” Journal of the American Statistical Association, 109, 991–1007. [496]
Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014), “Do We Need Hundreds of Classifiers to Solve Real World Classification Problems,” Journal of Machine Learning Research, 15, 3133–3181. [493]
Friedman, Y., and Schapire, R. E. (1995), “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” in European Conference on Computational Learning Theory, Springer, pp. 23–37. [494]
Friedman, J. H. (2001), “Greedy Function Approximation: A Gradient Boosting Machine,” The Annals of Statistics, 29, 1189–1232. [493,494,500,501,502]
Friedman, J. H., Hastie, T., and Tibshirani, R. (2001), The Elements of Statistical Learning, Springer Series in Statistics (Vol. 1), New York: Springer. [493]
Hoeffding, W. (1948), “A Class of Statistics With Asymptotically Normal Distribution,” The Annals of Mathematical Statistics, 19, 293–325. [495]
Hooker, G. (2007), “Generalized Functional ANOVA Diagnostics for High-Dimensional Functions of Dependent Variables,” Journal of Computational and Graphical Statistics, 16, 709–732. [502]
Hooker, G., and Mentch, L. (2018), “Bootstrap Bias Corrections for Ensemble Methods,” Statistics and Computing, 28, 77–86. [494,498,500,501]
—— (2019), “Please Stop Permuting Features: An Explanation and Alternatives,” arXiv no. 1905.03151. [502]
Ishwaran, H. (2007), “Variable Importance in Binary Regression Trees and Forests,” Electronic Journal of Statistics, 1, 519–537. [502]
Lichman, M. (2013), UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science. [493,494,500]
Mentch, L., and Hooker, G. (2016), “Quantifying Uncertainty in Random Forests via Confidence Intervals and Hypothesis Tests,” The Journal of Machine Learning Research, 17, 841–881. [493,494,496,497,502]
—— (2017), “Formal Hypothesis Tests for Additive Structure in Random Forests,” Journal of Computational and Graphical Statistics, 26, 589–597. [502]
Peng, W., Coleman, T., and Mentch, L. (2019), “Asymptotic Distributions and Rates of Convergence for Random Forests and Other Resampled Ensemble Learners,” arXiv no. 1905.10651. [497]
Schapire, R. E., and Freund, Y. (2012), Boosting: Foundations and Algorithms, Cambridge, MA: MIT Press. [494]
Scornet, E. (2016), “Random Forests and Kernel Methods,” IEEE Transactions on Information Theory, 62, 1485–1500. [495]
Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008), “Conditional Variable Importance for Random Forests,” BMC Bioinformatics, 9, 307. [502]
Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007), “Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution,” BMC Bioinformatics, 8, 25. [502]
van der Vaart, A. W. (2000), Asymptotic Statistics (Vol. 3), Cambridge: Cambridge University Press. [493,496]
Wager, S., and Athey, S. (2018), “Estimation and Inference of Heterogeneous Treatment Effects Using Random Forests,” Journal of the American Statistical Association, 113, 1228–1242. [493,494,496,497,498,499]
Wager, S., Hsieh, T., and Efron, B. (2014), “Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife,” Journal of Machine Learning Research, 15, 1625–1651. [498]
Xu, R. (2013), “Improvements to Random Forest Methodology,” Ph.D. thesis, Iowa State University. [493]
Zhang, G., and Lu, Y. (2012), “Bias-Corrected Random Forests in Regression,” Journal of Applied Statistics, 39, 151–160. [493,494,498]
Zhou, Y., Zhou, Z., and Hooker, G. (2018), “Approximation Trees: Statistical Stability in Model Distillation,” arXiv no. 1808.07573. [502]
Zhou, Z., and Hooker, G. (2019), “Unbiased Measurement of Feature Importance in Tree-Based Methods,” arXiv no. 1903.05179. [502]
Zhou, Z., Mentch, L., and Hooker, G. (2019), “Asymptotic Normality and Variance Estimation for Supervised Ensembles,” arXiv no. 1912.01089. [498,499]