Noname manuscript No.
(will be inserted by the editor)

Bootstrap Bias Corrections for Ensemble Methods

Giles Hooker · Lucas Mentch

Received: date / Accepted: date

Abstract This paper examines the use of a residual bootstrap for bias correction in machine learning regression methods. Accounting for bias is an important obstacle in recent efforts to develop statistical inference for machine learning. We demonstrate empirically that the proposed bootstrap bias correction can lead to substantial improvements in both bias and predictive accuracy. In the context of ensembles of trees, we show that this correction can be approximated at only double the cost of training the original ensemble. Our method is shown to improve test-set accuracy over random forests by up to 70% on example problems from the UCI repository.

Keywords Bagging · Ensemble Methods · Bias Correction · Bootstrap

1 Introduction

This paper proposes a bootstrap-based means of reducing bias in ensemble methods in machine learning. In non-parametric predictive modeling, accuracy is obtained by a trade-off between bias and variance. However, until recently, little attention has been given to quantifying either of these quantities. Very recently Mentch and Hooker (2016b); Wager (2014) have developed tools to quantify the variance in random forests (RF: Breiman, 2001) and other ensemble methods such as bagging (Breiman, 1996). These papers developed central limit theorems for the predictions of ensemble methods with a variance that scales as $n^{-1/2}$. These results follow heuristic means of producing confidence intervals in Sexton and Laake (2009) and Wager et al (2014). Using these results Mentch and Hooker (2016a) examined tests of variable importance and variable interaction. However, such confidence intervals and tests provide inference around the expected value of the estimated model, rather than about the underlying generating function. That is, they neither quantify nor correct for bias which will not necessarily decrease more rapidly than the variance. It is important to note that while variants on RF have been shown to have

Supported by NSF grants DMS 1053252 and DEB 1353039

G. Hooker
Cornell University
Tel.: +1-607-255-1638
Fax: +1-607-255-4698
E-mail: gjh27@cornell.edu

L. Mentch
University of Pittsburgh
E-mail: lkm31@pitt.edu

consistent predictions (Biau et al, 2008; Scornet et al, 2015), these results only show that both bias and variance decrease rather than produce a means of quantifying either.

This paper presents a method to decrease the bias of ensemble methods via a residual bootstrap. Bias correction via the bootstrap has a substantial history (Efron, 1979; Efron and Tibshirani, 1993). While this bias correction does not reduce the order of the bias when applied to kernel smoothing except at the edges of covariate space, it can still yield substantial performance improvements. It also provides an opportunity to improve prediction – while many of the papers cited above *quantify* variance in predictions, none reduce it. By contrast, the methods we present below can yield a substantial improvement in predictive performance for regression problems.

The use of a residual bootstrap was proposed in Freedman et al (1981) and its use in non-parametric regression has been examined in Härdle and Bowman (1988) and Hall and Horowitz (2013), however its direct application to machine learning methods has been hampered by the computational complexity involved in re-fitting a prediction model over B bootstrap replicates. We demonstrate that, in the context of ensemble methods, an approximate residual bootstrap can be computed at the same additional cost as computing only one – rather than B – additional predictive models. In simulation and on example data, this bias correction not only significantly reduces bias, it can also result in dramatic improvements in predictive accuracy for regression problems.

In the remainder of this paper, Section 2 provides an introduction to the residual bootstrap and bias correction; Section 3 develops the application to ensembles of trees; we provide an analysis of its theoretical properties in Section 4. Section 5 assesses the effectiveness of the bias reduction in simulation while Section 6 demonstrates the predictive performance improvement on regression problems from the UCI database (Lichman, 2013).

2 The Bootstrap and Bias Corrections

The bootstrap was introduced in Efron (1979) with the aim of assessing variability in statistics when a theoretical variance is either unknown or not estimable. It also presents a means of correcting for some forms of bias. The idea is simply to simulate from the empirical distribution of the data (i.e. resample with replacement) as a means of constructing an approximation of the sampling distribution of the statistic. For a data set $X_1, \ldots, X_n$ and a statistic of interest $T(X_1, \ldots, X_n)$, the standard bootstrap estimates the sampling distribution of T by recalculating it off bootstrap samples of the data to obtain a set of bootstrapped statistics $T^1, \ldots, T^{B_0}$. This then allows quantities such as variance to be calculated from the T^j. It also allows the bias to be assessed from the difference between the observed statistic and the average of the bootstrapped statistics (see Efron and Tibshirani, 1993, Ch 10):

$$\text{bias} = \left(\frac{1}{B_0} \sum_{b=1}^{B_0} T^b \right) - T(X_1, \ldots, X_n).$$

We can then obtain a bias-corrected statistic:

$$T^c = T(X_1, \ldots, X_n) - \text{bias} = 2T(X_1, \ldots, X_n) - \frac{1}{B_0} \sum_{b=1}^{B_0} T^b.$$

The bootstrap is defined formally in Algorithm 1. An analysis of the asymptotic properties of the bootstrap can be found in Hall (1992a) among many others.

There is an immediate connection between the bootstrap as detailed above and the bagging methods proposed in Breiman (1996) and used also in RF (Breiman, 2001) – the statistic, T being expressed as the map from a training set to the prediction of a single tree. However, since these methods already employ a bootstrap procedure, bootstrapping them again would represent a considerable burden. While

Algorithm 1 Bootstrap

Input: data X_i, size n, statistic $T(X_1, \ldots, X_n)$
for $b = 1$ **to** B_0 **do**
  Obtain $X_{1_b}, \ldots, X_{n_b}$ by resampling X_1,...,X_n with replacement.
  Calculate $T^b = T(X_{1_b}, \ldots, X_{n_b})$
end for
Estimate Var(T) by variance of $T^1, \ldots, T^b$.
Obtain the bias-corrected estimate:

$$T^c = 2T(X_1, \ldots, X_n) - \frac{1}{B_0}\sum_{b=1}^{B_0} T^b.$$

the bootstrap standard deviation is a consistent estimate of the variability of $T(X_1, \ldots, X_n)$, it does not estimate the variance of $\frac{1}{B_0}\sum_{b=1}^{B_0} T^b$. For this reason both Mentch and Hooker (2016b) and Wager (2014) employed subsampling rather than full bootstrap sampling which enables a variance calculation by extending results for U-statistics and the infinitesimal jackknife (Efron, 2014).

The fact that these methods already contain a bootstrap procedure means that a bootstraped RF should have the same mean as the original RF; it is the same as adding further trees. Thus the bias correction above should not be expected to change the bootstrap estimate. Instead, we propose employing a *residual bootstrap*; see Freedman et al (1981). This is a modified bootstrap for regression models of the form:

$$Y_i = F(X_i) + \epsilon_i$$

in which F (specified parametrically or non-parametrically) is the object of interest. For this model, sampling from the residual bootstrap can be expressed, following an estimate of $\hat{F}(X_i)$, as

1. Obtain residuals $\hat{\epsilon}_i = Y_i - \hat{F}(X_i)$
2. Obtain new responses by bootstrapping these residuals

$$Y_i^b = \hat{F}(X_i) + \hat{\epsilon}_{i_b}$$

with the pairs (X_i, Y_i^b) employed to create a new estimate $\hat{F}^b$. This modified resampling scheme can be directly applied to Algorithm 1. In the context of nonparametric regression, Härdle and Bowman (1988) examined bias and variance estimates for kernel smoothing; the coverage of confidence intervals was examined in Hall (1992b). There are numerous variants on this procedure, for example the $\hat{\epsilon}_i$ can be centered ($\hat{\epsilon}_i^c = \hat{\epsilon}_i - \bar{\hat{\epsilon}}$), and inflated to adjust for the optimism in $\hat{F}$. For a linear smoother in which the vector of fitted values can be expressed as $\hat{\boldsymbol{Y}} = H\boldsymbol{Y}$, the inflated residuals can be taken as $\hat{\epsilon}_i^I = \hat{\epsilon}_i/(1 - h_{ii})$ to obtain the original error variance (see Härdle and Bowman, 1988). We will employ out-of-bag residuals as an analogous means of overcoming optimism in Section 3.

While this paper is focussed on regression methodologies, classification can be handled by replacing the bootstrap sample of residuals with a simulation from $P(Y_i = 1|X_i)$ according to the model – the parametric bootstrap (Efron and Tibshirani, 1993).

In the next section, we outline a residual bootstrap that can be applied efficiently to ensemble methods.

3 A Cheap Residual Bootstrap for Ensembles

For RF and other ensemble methods that combine B trees, a naïve implementation of the residual bootstrap methodology in Section **??** requires recomputing the entire ensemble B_0 times – one for each bootstrap – resulting in a total of BB_o trees. In this section, we show that this is unnecessarily computationally intensive if we are only interested in obtaining a bias correction (see Mentch and Hooker, 2016b, for variance estimates).

The central observation is that, rather than learning an entirely new RF for each residual bootstrap, we can simply learn a single new tree for each residual bootstrap iteration. When estimating the bias in the RF, we construct a shortened residual bootstrap given in Algorithm 2 that results in B_o trees rather than the BB_o that a naïve implementation would require. In Section 4, we show that as long as $B_0 = B^{1+\epsilon}$, this procedure provides equivalent bias reduction as the naive implementation.

To make our algorithm formal, we take $T_x((X_1, Y_1), \ldots, (X_n, Y_n), \omega)$ to be the function that builds a tree from the data $(X_1, Y_1), \ldots, (X_n, Y_n)$ using random number seed ω and makes a prediction at the point x. A prediction from a RF can then be expressed as

$$\hat{F}_B(x) = \frac{1}{B}\sum_{b=1}^{B} T_x\left((X_{1_b}, Y_{1_b}), \ldots, (X_{n_b}, Y_{n_b}), \omega_b\right)$$

and an estimate of residuals can be obtained by examining the *out of bag* predictions. That is, we denote by I_b, the set of indices of the observations that occur in bootstrap sample b, then define

$$\hat{\epsilon}_i^o = Y_i - \frac{1}{\sum_{b=1}^{B} 1(i \notin I_b)} \sum_{b=1}^{B} T_x\left((X_{1_b}, Y_{1_b}), \ldots, (X_{n_b}, Y_{n_b}), \omega_b\right) 1(i \notin I_b) \tag{1}$$

to be the residuals calculated from the trees which were not trained using (X_i, Y_i). We can now use these as being the equivalent of inflated residuals (described in Section 2) in a residual bootstrap.

Algorithm 2 Shortened Bootstrap

Input Data (X_i, Y_i) size n.
Input Ensemble of B functions $\hat{F}(x) = \frac{1}{B}\sum_{b=1}^{B} T_x\left((X_{1_b}, Y_{1_b}), \ldots, (X_{n_b}, Y_{n_b}), \omega_b\right)$
Set

$$\hat{\epsilon}_i^o = Y_i - \frac{1}{\sum_{b=1}^{B} 1(i \notin I_b)} \sum_{b=1}^{B} T_{X_i}\left((X_{1_b}, Y_{1_b}), \ldots, (X_{n_b}, Y_{n_b}), \omega_b\right) 1(i \notin I_b)$$

for b = 1 **to** B_o **do**
  Obtain $\hat{\epsilon}_i^{ob}$ by resampling the $\hat{\epsilon}_i^o$ with replacement.
  Set $Y_i^{ob} = \hat{F}(X_i) + \hat{\epsilon}_i^{ob}, i = 1, \ldots, n$.
  Build the tree $T_x((X_{1_b}, Y_{1_b}^{ob}), \ldots, (X_{n_b}, Y_{n_b}^{ob}), \omega_b)$
end for
Set $\hat{F}_{B_o}^0(x) = \frac{1}{B_o}\sum_{b=1}^{B_o} T_x((X_{1_b}, Y_{1_b}^{ob}), \ldots, (X_{n_b}, Y_{n_b}^{ob}))$
Return $\hat{F}_{BB_o}^c(x) = 2\hat{F}_B(x) - \hat{F}_{B_o}^o(x)$.

We label the result of this the bias-corrected Random Forest (RFc). Note that this bootstrap procedure is only valid for bias correction. To estimate variance, we can employ the methods proposed in Mentch and Hooker (2016b).

4 Computational Costs and Theoretical Properties

Our proposed method replaces building an ensemble of trees for each residual bootstrap with building a single tree. At an intuitive level, this allows us to cover the same number of residual bootstrap iterations at lower cost. Further, so long as B_o grows (slightly) faster than B we would expect that the variance reduction in building an ensemble for each residual bootstrap would not yield an asymptotic improvement over the correction in Algorithm 2. In this section we formalize this intuition.

Mentch and Hooker (2016b) demonstrated that under mild regularity conditions, predictions from RFs built using subsamples of size $m = o(\sqrt{n})$ out of n examples have the following central limit theorem

$$\frac{\hat{F}_B(x) - E\hat{F}_B(x)}{\sqrt{\frac{m^2}{n}\zeta_1(x) + \frac{1}{B}\zeta_m}} \xrightarrow{d} N(0,1) \tag{2}$$

in which $\zeta_1(x)$ and $\zeta_m(x)$ have known expressions. For an idealization of RF, Wager et al (2014) relaxed this condition to allow $m = o(n/\log(n)^p)$ in the case that $n/B \to 0$ where p is the dimension of x.

We see here that the variance in this central limit theorem is $O(\min(n, B)^{-1})$. Scornet (2014) identifies the two components of the variance terms in the denominator of (2) as the distinction between infinite RF's (in which $B = \infty$) and their Monte Carlo approximation for finite B. Following this approach, we identify an infinite bootstrap $\hat{F}^c_\infty$ achieved by setting $B_o = B = \infty$. From the law of large numbers, we can equivalently think of $\hat{F}^c_\infty(x)$ as the expectation of $\hat{F}_{BB_o}(x)$ taken over all randomization elements, including the selection of bootstrap samples. In this framework we obtain the following uniform convergence rate:

Theorem 1 *Let $Y_i = F(X_i) + \epsilon_i$, $\epsilon_i \sim N(0, \sigma^2)$ and let $||F||_\infty$, the supremum of F on the support of X, be finite. Then*

$$E\left(\hat{F}^c_{BB_o}(x) - \hat{F}^c_\infty(x)\right)^2 \leq \left(\frac{64}{B} + \frac{80}{B_o}\right)\left[||F||^2_\infty + \sigma^2(1 + 4\log(n))\right].$$

The proof of this result is given formally in Appendix A. It relies on representing each of $\hat{F}_B(x)$ and $\hat{F}^o_{B_o}(x)$ as a weighted average of the Y_i. These weights are themselves averages of weights from individual trees and incorporate any randomization effects, including which observations are used in which bootstrap samples. The expect squared error can them be calculated from the variance of these weights. Since the weights must sum to 1, the bound can be extended to cover the process generating the data.

From Theorem 1, so long as $B_o = O(B^{1+\epsilon})$, the variance associated with employing a reduced number of residual bootstraps can be ignored asymptotically. In practice, we have used $B_o = B$ or $B_o = 2B$ and found our results insensitive to this choice; hence the bias correction may be made at no more than the same cost as obtaining the original ensemble.

We remark here that the $\log(n)$ factor is a consequence of a bound on $E\max_i \epsilon_i^2$ and conjecture that it is not sharp. Theorem 1 can be readily extended to a condition that the ϵ_i have sub-Gaussian tails. Furthermore, when the Y_i are bounded, we can replace $\log(n)$ with a constant. Similar rates can be shown to hold in the case of a parametric bootstrap in which Y_i is simulated according to $\hat{F}_B(X_i)$.

Note here that this calculation does not include the variance associated with the infinite bootstrap bias correction. That is, even with an idealized $B_o = \infty$, $\hat{F}^o_\infty$ will still have some variance and $\hat{F}^c_\infty$ may be more variable than $\hat{F}_\infty$ – we observe about 50% additional variance for our bias-corrected estimates in the examples below. A central limit theorem for $\hat{F}^o_{B_o}$ of the form of (2) can be obtained from an extension of Mentch and Hooker (2016b) using 2-sample U-statistics, and in particular has variance of order $O(\min(n, B_o)^{-1})$, hence maintaining our calculations. However, formal inference for $\hat{F}^c$ also needs to account for the correlation between $\hat{F}_B$ and $\hat{F}^o_{B_o}$ and is beyond the scope of this paper.

5 Numerical Experiments

In this section we present simulation experiments to examine the effect of bootstrap bias corrections. An advantage of employing simulated data is that bias can be evaluated explicitly. We first investigate the

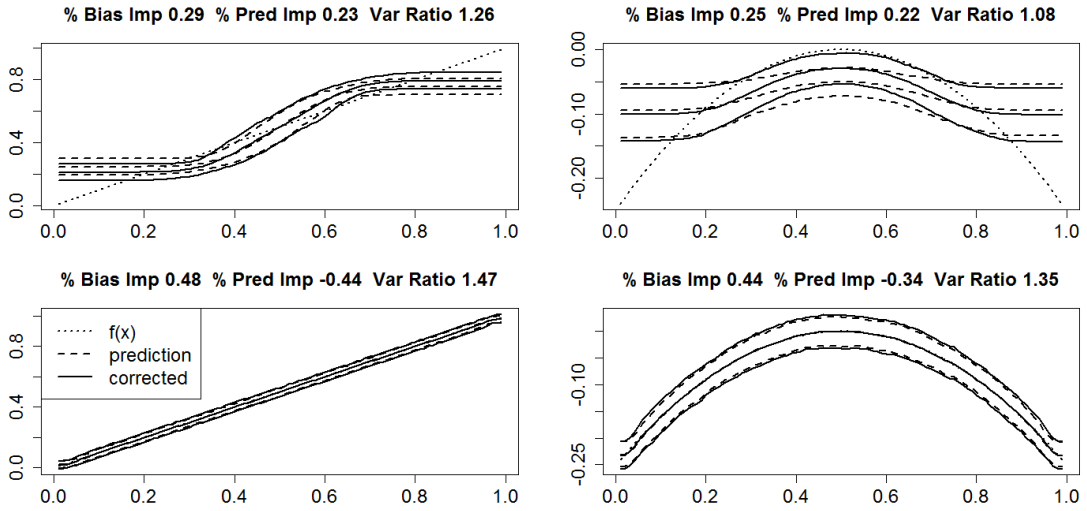

Fig. 1 Effect of bias correction in 1 dimensional bagged trees. Dotted lines provide exact relationship. Dashed: 5%, 95% and mean values of predictions from bagged trees. Solid: 5%, 95% and mean values of predictions from bias-corrected bagged trees. Top panels: based on subsamples of size 20; bottom: subsamples of size 200.

performance of our residual bootstrap in a regression setting. Then we perform a parametric bootstrap for a classification problem. In both cases, we see substantial improvements in the size of the bias, but expect that the bias correction will yield greater improvements for predictive accuracy in regression than in classification.

5.1 Regression

As a first study, Figure 1 presents the results of employing a bagged decision tree using only one covariate generated uniformly on the interval [0, 1]. One dimensional examples were produced in order to visualize the effect of bias at the edges of the data. We examined two response models:

$$y_i = x_i + \epsilon_i,\ y_i = -(x_i - 0.5)^2 + \epsilon_i$$

which have different bias properties. In each case, ϵ_i was generated from a Gaussian distribution with standard deviation 0.1. We used 1000 observations and built 1000 trees – intended to be enough to reduce variance due to subsampling at any subsample size. We present results for providing trees with subsamples of size 20 and of size 200. Above each figure we report

Bias Imp The percentage improvement in squared bias over an uncorrected estimate, averaged over 100 test points with the same distribution as the training data. Bias was calculated by the difference between the prediction *averaged over all simulations* at each point and the true prediction function.

Pred Imp The percentage improvement in squared error between predictions *for each simulation* and the true prediction function. Note that this is a measure for *noiseless* observations at new data points. We would expect this improvement to decrease if noise were added to the test set responses.

Var Ratio The ratio of the variance of bias-corrected predictions to the variance of uncorrected predictions.

Function	Subsample	Type	Bias Imp	Pred Imp	Var Ratio
$(\sum_{i=1}^{10} \lvert x_i \rvert)^{1/2}$	500	BT	0.55	0.51	1.45
	500	RF	0.54	0.48	2.01
	5000	BT	0.35	0.2	1.29
	5000	RF	0.25	0.11	1.46
$-\sum_{i=1}^{10} x_i^2$	500	BT	0.37	0.35	1.43
	500	RF	0.54	0.52	1.73
	5000	BT	0.35	0.36	1.08
	5000	RF	0.24	0.11	1.46

Table 1 Performance of bootstrap bias correction in 10-dimensional regression examples. Models are averages of 1,000 trees based on 5,000 data points using subsamples of size 500 or 5,000 for each tree. Ensemble type is either bagged trees (BT) or Random Forests (RF). Bias Imp = percent improvement in squared bias. Pred Imp = percentage improvement in mean squared error. Var Ratio = ratio of averaged pointwise variances between corrected and uncorrected decision trees.

For one dimensional simulations, the bias correction we propose helps to reduce bias, but this may come at the cost of an increase in variance and hence a reduction in over-all accuracy. This is particularly true for large subsample sizes in which bias is less important.

However, we are rarely interested in one-dimensional prediction. We also expect bias to be larger in higher dimensions and we therefore experimented with a 10 dimensional model. For this simulation, 5000 examples were generated from a 10-dimensional Gaussian model with variance 1.8 for each feature and covariance 0.8 between each feature pair. Here again, two models were considered:

$$y_i = \sqrt{\sum_{j=1}^{10} |x_{ij}|/10} + \epsilon_i \tag{3}$$

$$y_i = -\sum x_{ij}^2/10 + \epsilon_i \tag{4}$$

in which the ϵ_i have standard deviation 0.1. These models provide differing edge-behavior; (4) producing having much steeper behavior which we expect to be difficult for tree-based methods to capture. For each simulated data set we built 1,000 trees using subsamples of sizes 500 and bootstrap samples of size 5000 using both CART and RF trees. In Table 1 we report the statistics described above. For CART trees we see a 35% to 55% reduction in bias as well as a 20% to 50% reduction in prediction error, representing a significant improvement in both; the improvement is similar for RF except for large subsample sizes where we still achieve a 10% reduction. Using larger subsamples improved the performance of the original ensemble and resulted in less (though still significant) bias reduction from our methods. However, using bootstrap samples instead of subsamples compromises the distributional theory on which inferential procedures such as those in Mentch and Hooker (2016b) relies.

5.2 Classification

We now examine the use of our proposed bias correction method in classification problems. We expect the use of this bias correction to have much more limited effect on prediction accuracy (i.e. the squared error between our estimated probability and the truth) in this case. This is because, in our settings, the signal to noise ratio for binary random variables is larger than in our regression simulations. For misclassification loss, this effect is even stronger: we need only determine the classification boundary, making bias correction elsewhere useless. Table 2 reports the results of classification experiments analogous to those above. For each simulation setting, the true probability was a logistic transform of a scaled and shifted version of the response function used in the regression models. They were similarly chosen to contrast the edge behavior of the functions. Table 2 specifies the logit probability of success in each of

Dimension	$Subsample$	logit($P(y=1)$)	Bias Imp	Pred Imp	Var Rat	Miss Imp
1	20	$3(x-1/2)$	0.2	0.07	1.4	-0.001
	200		0.54	-0.45	1.53	-0.008
	20	$-30(x-1/2)^2-2.17$	0.33	0.32	1.26	0.06
	200		0.71	0.04	1.68	-0.011
10	500	$5(\sum_{i=1}^{10}\|x_i\|)^{1/2}-5$	0.52	-0.01	2.17	-0.005
	5000		0.37	-0.64	1.95	-0.02
	500	$-2\sum_{i=1}^{10}x_i^2+2.4$	0.42	0.33	1.94	0.01
	5000		0.42	-0.1	1.87	-0.013

Table 2 Performance of bootstrap bias correction for simulation of classification tasks. Column headings are as in Table 1, in addition, Mis Imp = relative misclassification improvement on test data.

these models. The bias correction was obtained by generating new responses for each tree according to the estimated probability from the original ensemble. We measured both squared-error accuracy in terms of ability to fit the true response and improvement in misclassification risk. Here we see mixed results for improvement in estimating the underlying probability. Unsurprisingly the effect on misclassification rate is negligible. We note that while this correction may not be useful for predictive accuracy, it may still be desirable when the target is scientific inference.

6 Case Studies

In order to assess the impact of the proposed bias correction in real world data, we applied random forests with and without the bias correction to 12 data sets in the UCI repository (Lichman, 2013) for which the task was labelled as regression. A description of the processing for each case study can be found in supplemental materials. In each data set, we applied 10-fold cross-validation to estimate the predictive mean squared error of RF and RFc. For each cross-validation fold, we learned a random forest using 1000 trees as implemented in the `randomForest` package (Liaw and Wiener, 2002) in `R` and employed a bias correction using 2000 residual bootstrap trees. The results are reported in Table 3 and are insensitive to using either 1000 or 5000 residual bootstrap trees. In most cases RFc reduced squared error compared to RF by between 2 and 10 percent. However, some examples (*airfoil*, *BikeSharing*, *Concrete*, and *yacht-hydrodynamics*) saw very substantial MSE reductions (42%, 34% 30% and 74% respectively). The bias correction increased MSE by 1% in two examples. We omitted results for *forestfires* in which RF performed no better than predicting a constant and where RFc increased MSE by 7%.

It is difficult to draw broader patterns concerning when a bias correction will make a significant difference. Informally, it appears to help most in cases with large signal to noise ratios (using RF reduces MSE by a large amount relative to predicting a constant) and moderate dimensions. It produces less improvement for high dimensional problems with large signal to noise ratios. However, there is considerable variation from these patterns in Table 3 which may be due to many other factors including the configuration of covariates and the shape of the underlying relationship.

7 Conclusion

We have proposed a residual bootstrap bias correction to random forests and other ensemble methods in machine learning. This correction can be calculated at no more than the same cost of learning the original ensemble. We have shown that this procedure substantially reduces bias in almost all problems: an important consideration when carrying out statistical inference. In some regression problems, it can also lead to substantial reduction in predictive mean squared error. Our focus has been on the effect of this

	N	p	Var(Y)	RF.Err	RFc.Err	RF.Imp	RFc.Imp
yacht-hydrodynamics	308	6	229.55	13.27	3.45	0.94	0.74
airfoil	1503	5	46.95	12.56	7.29	0.73	0.42
BikeSharing-hour	17379	14	32913.74	55.28	36.6	1	0.34
housing	506	13	0.17	0.02	0.02	0.88	0.09
CCPP	9568	4	291.36	10.8	9.92	0.96	0.08
auto-mpg	392	7	61.03	7.45	7.02	0.88	0.06
winequality-white	4898	11	0.79	0.35	0.33	0.55	0.05
winequality-red	1599	11	0.65	0.33	0.32	0.5	0.03
Concrete	1030	8	279.08	27.24	18.94	0.9	0.3
parkinsons	5875	16	66.14	42.01	40.79	0.36	0.03
communities	1994	96	0.05	0.02	0.02	0.66	-0.01
SkillCraft	3338	18	2.1	0.84	0.84	0.6	-0.01

Table 3 Cross validation performance of random forests and the bias correction in 12 UCI regression tasks. Var(Y) gives the variance of the responses, RF.Err is the cross-validated MSE for random forests, RFc.Err is the cross-validated MSE for bias-corrected random forests, RF.Imp = 1 - RF.Err/Var(Y) is the improvement of random forests relative to predicting a constant, RFc.Imp = 1 - RFc.Err/RF.Err is the relative improvement of adding the bias correction. Results are ordered by improvement due to the bias correction.

bias correction on RF and we have therefore not compared this performance to other methods. Applying this correction to other learning algorithms may not be possible without large computational overhead. However we expect that doing so would demonstrate similar improvements.

Theoretically, we have shown that the Monte Carlo error in this correction can be ignored provided more residual bootstrap samples are used than used to build the original ensemble. However, we have not treated the properties of the bias correction under infinite resampling; ie, the bias properties of $\hat{F}^c_\infty$. Here we do not expect to be able to improve the convergence rate for the bias, except at the edge of covariate space. By way of motivation for this statement; for a one-dimensional Nadaraya-Watson estimator with bandwidth h (see Eubank, 1999), the bias in the interior of the support of X is $O(h^2)$ and the residual bootstrap proposed here will not change this (see calculations in Härdle and Bowman, 1988). However, near the edge of covariate support, it is possible to show that the residual bootstrap will decrease the order of bias from $O(h)$ to $O(h^2)$. A possible explanation for the success of the proposed correction in moderate dimensions is that most covariate values are near the edge of this support. We also believe that a central limit theorem can be obtained for $\hat{F}^c_{BB_o}$, but doing so will need to account for the variance of $\hat{F}_B$ used when building $\hat{F}^o_{B_o}$.

References

Biau G, Devroye L, Lugosi G (2008) Consistency of random forests and other averaging classifiers. Journal of Machine Learning Research 9:20152033

Boucheron S, Lugosi G, Massart P (2013) Concentration inequalities: A nonasymptotic theory of independence. Oxford University Press

Breiman L (1996) Bagging predictors. Machine learning 24(2):123–140

Breiman L (2001) Random forests. Machine Learning 45:5–32

Brooks TF, Pope DS, Marcolini MA (1989) Airfoil self-noise and prediction, vol 1218. National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Division

Cortez P, Morais AdJR (2007) A data mining approach to predict forest fires using meteorological data

Cortez P, Cerdeira A, Almeida F, Matos T, Reis J (2009) Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems 47(4):547–553

Efron B (1979) Bootstrap methods: another look at the jackknife. The annals of Statistics pp 1–26

Efron B (2014) Estimation and accuracy after model selection. Journal of the American Statistical Association 109(507):991–1007

Efron B, Tibshirani RJ (1993) An Introduction to the Bootstrap. CRC Press, New York

Eubank RL (1999) Nonparametric regression and spline smoothing. CRC press

Fanaee-T H, Gama J (2013) Event labeling combining ensemble detectors and background knowledge. Progress in Artificial Intelligence pp 1–15, DOI 10.1007/s13748-013-0040-3, URL [WebLink]

Freedman DA, et al (1981) Bootstrapping regression models. The Annals of Statistics 9(6):1218–1228

Gerritsma J, Onnink R, Versluis A (1981) Geometry, resistance and stability of the delft systematic yacht hull series. Delft University of Technology

Hall P (1992a) The bootstrap and Edgeworth expansion. Springer

Hall P (1992b) On bootstrap confidence intervals in nonparametric regression. The Annals of Statistics pp 695–711

Hall P, Horowitz J (2013) A simple bootstrap method for constructing nonparametric confidence bands for functions. The Annals of Statistics 41(4):1892–1921

Härdle W, Bowman AW (1988) Bootstrapping in nonparametric regression: Local adaptive smoothing and confidence bands. Journal of the American Statistical Association 83(401):102–110

Harrison D, Rubinfeld DL (1978) Hedonic prices and the demand for clean air. Journal of Environmental Economics and Management 5:81–102

Liaw A, Wiener M (2002) Classification and regression by randomforest. R News 2(3):18–22, URL http://CRAN.R-project.org/doc/Rnews/

Lichman M (2013) UCI machine learning repository. URL http://archive.ics.uci.edu/ml

Little MA, McSharry PE, Roberts SJ, Costello DA, Moroz IM, et al (2007) Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. BioMedical Engineering OnLine 6(1):23

Mentch L, Hooker G (2016a) Formal hypothesis tests for additive structure in random forests. Journal of Computational and Graphical Statistics (In Press)

Mentch L, Hooker G (2016b) Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. Journal of Machine Learning Research 17(26):1–41

Quinlan JR (1993) Combining instance-based and model-based learning. In: Proceedings of the Tenth International Conference on Machine Learning, pp 236–243

Redmond M, Baveja A (2002) A data-driven software tool for enabling cooperative information sharing among police departments. European Journal of Operational Research 141(3):660–678

Scornet E (2014) On the asymptotics of random forests. arXiv preprint arXiv:14092090

Scornet E, Biau G, Vert JP (2015) Consistency of random forests. The Annals of Statistics 43(4):1716–1741

Sexton J, Laake P (2009) Standard errors for bagged and random forest estimators. Computational Statistics & Data Analysis 53(3):801–811

Thompson JJ, Blair MR, Chen L, Henrey AJ (2013) Video game telemetry as a critical tool in the study of complex skill learning. PloS one 8(9):e75,129

Tüfekci P (2014) Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. International Journal of Electrical Power & Energy Systems 60:126–140

Wager S (2014) Asymptotic Theory for Random Forests. ArXiv e-prints 1405.0352

Wager S, Hastie T, Efron B (2014) Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. The Journal of Machine Learning Research 15(1):1625–1651

Yeh IC (1998) Modeling of strength of high-performance concrete using artificial neural networks. Cement and Concrete research 28(12):1797–1808

A Proof of Theorem 1

Proof We begin by writing the prediction at x from an individual tree as

$$\begin{aligned} T_b(X, \Omega) &= \sum_{i=1}^{n} \frac{L(x, X_i, \Omega_b)}{N(x, \Omega_b)} Y_i \\ &= \sum_{i=1}^{n} W_i(x, \Omega_b) Y_i \end{aligned}$$

where Ω_b is the realization of a random variable that describes both the selection of bootstrap or subsamples used in learning the tree T_b as well as any additional random variables involved in the learning process (e.g. the selection of candidate split variables in RF). Here $L(x, X_i, \Omega_b)$ is the indicator that x and X_i are in the same leaf of a tree learned with randomization parameters Ω_b and $N(x, \Omega_b)$ is the number of observations in the same leaf as x. We will also write

$$\bar{W}_i^B(x) = \frac{1}{B} \sum_{b=1}^{B} W_i(x, \Omega_b)$$

as the average weight on Y_i across all resamples so that

$$\hat{F}_B(x) = \sum_{i=1}^{n} \bar{W}_i^B(x) Y_i$$

Note that

$$\sum_{i=1}^{n} W_i(x, \Omega_b) = \sum_{i=1}^{n} \bar{W}_i^B(x) = 1.$$

We can similarly write a residual-bootstrap tree as

$$\begin{aligned} T_{b^o}^o &= \sum_{i=1}^{n} \sum_{j=1}^{n} V_{ij}(x, \Omega_{b^o}) Y_i^o \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} V_{ij}(x, \Omega_{b^o}) [\hat{F}(X_i) + (Y_j - \hat{F}(X_j))] \end{aligned}$$

with the corresponding quantities

$$\bar{V}_{ij}^{B_o}(x) = \frac{1}{B_o} \sum_{b^o=1}^{B_o} V_{ij}(x, \Omega_{b^o})$$

where we also have

$$\sum_{i=1}^{n} \sum_{j=1}^{n} V_{ij}(x, \Omega_{b^o}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{V}_{ij}^{B_o}(x) = 1.$$

Using these quantities we can write $\hat{F}_{BB_o}^c(x)$ as

$$\begin{aligned} &2\hat{F}_b(x) - \hat{F}_{B_o}^o(x) \\ &\quad = \sum_{i=1}^{n} 2\bar{W}_i^B(x) Y_i \\ &\qquad - \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{V}_{ij}^{B_o}(x) [\hat{F}_B(X_i) + (Y_j - \hat{F}_B(X_j))] \\ &\quad = \sum_{i=1}^{n} 2\bar{W}_i^B(x) Y_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{V}_{ij}^{B_o}(x) Y_i \\ &\qquad + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \bar{V}_{kj}^{B_o}(x) \left(\bar{W}_i^B(X_k) - \bar{W}_i^B(X_j) \right) Y_i. \end{aligned}$$

Hence letting $\text{Var}_\Omega(W_i(x,\Omega))$ indicate variance with respect to only the randomization parameters Ω, writing $Y_i = F(X_i) + \epsilon_i$ and observing that $0 \leq W_i(x,\Omega) \leq 1$, $0 \leq V_{ij}(x,\Omega) \leq 1$:

$$\begin{aligned}
&E\left(\hat{F}^c - \hat{F}^c_\infty\right)^2 \\
&\leq \frac{8}{B} E_Y \text{Var}_\Omega \left(\sum_{i=1}^n W_i(x,\Omega) Y_i\right) \\
&\quad + \frac{2}{B} E_y \text{Var}_\Omega \left(\sum_{i=1}^n \sum_{j=1}^n V_{ij}(x,\Omega) Y_i\right) \\
&\quad + \frac{2}{B_o} E_Y \text{Var}_{\Omega_{b^o},\Omega_b} \left(\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n H_{ijk}\right) \\
&\leq \frac{8}{B}\left[2 \max_{ij}(F(X_i) - F(X_j))^2 + 2 \max_{ij}(\epsilon_i - \epsilon_j)^2\right] \\
&\quad + \frac{2}{B_0}\left[16 \max_{ij}(F(X_i) - F(X_j))^2 + 10 \max_{ij}(\epsilon_i - \epsilon_j)^2\right] \\
&\leq \frac{64}{B}\left[||F||^2_\infty + \sigma^2(1 + 4\log(n))\right] \\
&\quad + \frac{80}{B_o}\left[||F||^2_\infty + \sigma^2(1 + 4\log(n))\right]
\end{aligned}$$

for

$$H_{ijk} = V^{B_o}_{kj}(x, \Omega_{b^o})\left(W^B_i(X_k, \Omega_b) - W^B_i(X_j, \Omega_b)\right) Y_i.$$

Here we use the fact that for $\epsilon_1, \ldots, \epsilon_n \sim N(0,1)$, $E\left(\max_i \epsilon_i^2\right) \leq 1 + 4\log(n)$ (Boucheron et al, 2013).

B Details of Case Study Data Sets

After processing each data set as described below, we employed 10-fold cross-validation to obtain cross-validated squared error for both $\hat{F}_B$ and $\hat{F}^c_{BB_o}$, removing the final data entries to create equal-sized folds. To maintain comparability, the same folds were used for both estimates. We set $B = 1000$ and $B_o = 2000$, but these results were insensitive to setting $B_o = 1000$ or $B_o = 5000$.

Below we detail each data set and the processing steps taken for it; unless processing is noted, data were taken as is from the UCI repository Lichman (2013).

airfoil 42% improvement over RF. Task is to predict sound pressure in decibels of airfoils at various wind tunnel speeds and angles of attack Brooks et al (1989). 1503 observations, 5 features.

auto-mpg 6% improvement over RF. Task is to predict city-cycle fuel consumption in miles per gallon from physical car and engine characteristics Quinlan (1993). Rows missing horsepower were removed resulting in 392 examples with 8 features, 3 of which are discrete.

BikeSharing-hour 34% improvement over RF. Prediction of number of rental bikes used each hour over in a bike-sharing system Fanaee-T and Gama (2013). Date and Season (columns 2 and 3) were removed from features as duplicating information, leaving 13 covariates related to time, weather and number of users. 17389 examples; prediction task was for log counts.

communities -1% improvement over RF. Prediction of per-capita rate of violent crime in U.S. cities Redmond and Baveja (2002). 1993 examples, 96 features. 30 (out of original 125) feature removed due to high-missingness including state, county and data associated with police statistics. One row (Natchezcity) deleted due to missing values. Cross-validation was done using independently-generated folds.

CCPP 8% improvement over RF. Prediction of net hourly output from Combined Cycle Power Plants Tüfekci (2014). 4 features and 9568 examples.

Concrete 3% improvement over RF. Prediction of concrete compressive strength from constituent components Yeh (1998). 9 features, 1030 examples.

forestfires -8% improvement over RF. Prediction of log(area+1) burned by forrest fires from location, date and weather attributes Cortez and Morais (2007). 517 examples, 13 features. Not reported in main paper because Random Forests predictions had 15% higher squared error than a constant prediction function.

housing 9% improvement over RF. Predict median housing prices from demographic and geographic features for suburbs of Boston Harrison and Rubinfeld (1978). Response was taken to be the log on median house prices. 506 examples, 14 attributes.

parkinsons 3% improvement over RF. Prediction of Motor UPDRS from voice monitoring data in early-state Parkinsons patients Little et al (2007). Removed features for age, sex, test-time and Total UPDRS, resulting in 15 features and 5875 examples.

SkillCraft -1% improvement over RF. Predict league index of gamers playing SkillCraft based on playing statistics Thompson et al (2013). Entries with NA's removed; results in 3338 examples and 18 features.

winequality-white 5% improvement over RF. Predict expert quality score on white wines based on 11 measures of wine composition Cortez et al (2009). 4898 examples.

winequality-red 3% improvement over RF. As in *winequality-white* for red wines Cortez et al (2009). 1599 examples.

yacht-hydrodynamics 70% improvement over RF. Predict residuary resistance per unit weight of displacement of sailing yachts from hull geometry Gerritsma et al (1981). 308 examples, 7 features.