Journal of Applied Statistics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/cjas20

Bias-corrected random forests in regression

Guoyi Zhang [a] & Yan Lu [a]
[a] Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM, 87131-0001, USA

Version of record first published: 19 May 2011.

To cite this article: Guoyi Zhang & Yan Lu (2012): Bias-corrected random forests in regression, Journal of Applied Statistics, 39:1, 151-160

To link to this article: http://dx.doi.org/10.1080/02664763.2011.578621

Journal of Applied Statistics
Vol. 39, No. 1, January 2012, 151–160

Taylor & Francis
Taylor & Francis Group

Bias-corrected random forests in regression

Guoyi Zhang and Yan Lu*

Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131-0001, USA

(*Received 6 December 2010; final version received 28 March 2011*)

It is well known that random forests reduce the variance of the regression predictors compared to a single tree, while leaving the bias unchanged. In many situations, the dominating component in the risk turns out to be the squared bias, which leads to the necessity of bias correction. In this paper, random forests are used to estimate the regression function. Five different methods for estimating bias are proposed and discussed. Simulated and real data are used to study the performance of these methods. Our proposed methods are significantly effective in reducing bias in regression context.

Keywords: bias correction; mean-squared prediction error; random forests; regression; simulation

1. Introduction

Random forests [5], which are a combination of tree predictors using both approaches of bagging and randomization, have received considerable attention in recent years. The accuracy of random forests has been shown to be competitive with many other data-mining techniques [5,11]. Biau and Devroye [1] discussed the links between layered nearest neighbour estimate and the random forest estimates and proved the universal consistency of the bagged nearest neighbour method for regression and classification. Random forest algorithm is able to estimate the important properties including variable importance, mean-squared prediction error (MSPE) and proximity measures. Variable importance measures are a sensible means for variable selection in many applications, but are not reliable in situations where potential predictor variables vary in their scale of measurement or in their number of categories. Strobl *et al.* [12] proposed to employ an alternative implementation of random forests, which provides unbiased variable selection in the individual classification trees. MSPE is commonly used to asses the accuracy of random forests. Brence and Brown [6] proposed a new forest prediction method called booming. In their previous unpublished work, they also studied robust measures in random forest regression. In regression, MSPE is the sum of the noise variance, the predictor bias and predictor variance, where the sum of the latter two terms

*Corresponding author. Email: luyan@math.unm.edu

ISSN 0266-4763 print/ISSN 1360-0532 online

http://dx.doi.org/10.1080/02664763.2011.578621
http://www.tandfonline.com

are called the risk of estimating the regression function. Breiman [2] showed that bagging could effectively reduce the variance of regression predictors, while leaving the bias unchanged. As the variance of the estimator becomes relatively small, it is a common fact that bias dominates the risk. On the other hand, because extreme observations are estimated using averages of response values that are closer to those observations, large values of the regression function are underestimated and small values of the regression function are overestimated. Consequently, bias is not negligible and bias correction is necessary.

Breiman [4] discussed using adaptive bagging to debias regressions. The procedure in his paper works in stages with the first-stage bagging. Based on the outcomes of the first stage, the output values are altered by subtracting the predictors and a second stage of bagging is carried out using the altered output values. This is repeated until a specified noise level is reached. Breiman [4] mentioned in his paper that experiments have showed the distance between the regression function f and the predictor $\hat{f}$ to be small. Consequently, estimating $f - \hat{f}$ at the second stage will bring extra variance. This is particularly true when $f - \hat{f}$ is already a small value, so the bias is a small value too. As more and more stages added, bias of the predictor goes to zero [4], while variance keeps increasing. This leads to the fact that the later stages are usually unnecessary.

In this paper, we develop five different methods to estimate the bias of random forests in regression. Our proposed methods extend the first debias stage from Breiman [4] from tree estimators to random forest estimators and use different models to estimate the bias. Since bias is a measure of the distance between the regression function and the average of the estimators of the regression function, we consider models of bias related to predictor variable $\mathbf{X}$ and response variable Y and models of bias related to both of them. Random forests and univariate smoothing methods are used to estimate the bias. The resulting predictor with subtracting the estimator of bias leads to our title: bias-corrected random forests in regression. Not surprisingly, experiments show that the proposed methods are significantly effective in reducing bias.

This paper is organized as follows. In Section 2, we recall bias–variance decomposition. In Section 3, we discuss why bias correction is necessary and propose our five methods of bias correction. In Section 4, we perform experimental studies. Finally, we give our concluding remarks in Section 5.

2. Bias–variance decomposition

Suppose we have a training set

$$\mathbf{T} = \{(y_i, \mathbf{x}_i), i = 1, 2, \ldots, N\}$$

consisting of N i.i.d. observations with $\mathbf{x}_i$ a vector of p variates. We can write y_i as

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \tag{1}$$

where $f(\mathbf{x}_i) = E(Y|\mathbf{X} = \mathbf{x}_i)$ and each error term ϵ_i follows the same distribution with mean 0 and variance σ^2. The main goal in the regression setting is to estimate the function f based on the fixed training set $\mathbf{T}$, for the purpose of approximating y at future observations of $\mathbf{x}$. We assume that random forests are used to form predictions $\hat{f}(\mathbf{x}, \mathbf{T})$ of y given the input $\mathbf{x}$. Let $\hat{f}(\mathbf{x}; \boldsymbol{\theta}_k)$ represent the predictor from the kth tree model in the forests where $\mathbf{x}$ is an input vector and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k\}_{k=1}^K$ are independently and identically distributed random vectors, which influence the tree construction. The random forest estimator is

$$\hat{f}(\mathbf{x}, \mathbf{T}) = K^{-1} \sum_{k=1}^{K} \hat{f}(\mathbf{x}, \boldsymbol{\theta}_k). \tag{2}$$

A natural measure of the effectiveness of $\hat{f}$ as a predictor of y is defined as MSPE:

$$\text{MSPE}[\hat{f}(\mathbf{X},\mathbf{T})] = E_{\mathbf{T}}[E_{\mathbf{X},Y}[(Y-\hat{f}(\mathbf{X},\mathbf{T}))^2]].$$

The bias–variance decomposition [2,8] is as follows:

$$\begin{aligned}\text{MSPE}[\hat{f}(\mathbf{X},\mathbf{T})] &= E_{\mathbf{T}}[E_{\mathbf{X},Y}[(Y-f(\mathbf{X},\mathbf{T}))^2]] \\ &\quad + E_{\mathbf{X}}[(f(\mathbf{X},\mathbf{T}) - E_{\mathbf{T}}[\hat{f}(\mathbf{X},\mathbf{T})])^2] \\ &\quad + E_{\mathbf{X},\mathbf{T}}[E_{\mathbf{T}}[\hat{f}(\mathbf{X},\mathbf{T})] - \hat{f}(\mathbf{X},\mathbf{T})]^2. \end{aligned} \tag{3}$$

The first term in Equation (3) is the noise variance, which does not depend on the estimator $\hat{f}$. The second term is the predictor bias and the third term is the predictor variance. We call the sum of predictor bias and predictor variance risk of estimating f, which measures the effectiveness of $\hat{f}$.

3. Estimators for bias

In this section, we first discuss why we need bias correction. Then, we propose five different methods to estimate the bias. Bias is a measure of the distance between the regression function and the average of the estimators of the regression function. Both predictor variable $\mathbf{X}$ and response variable Y are important in estimating the bias. We consider models of bias related to $\mathbf{X}$ and Y and models of bias related to both of them. Random forests and univariate smoothing methods are used to estimate the bias.

From now on, we drop the subscript $\mathbf{T}$ and use $\hat{f}$ to denote the random forest estimator of f. Notation $B(\mathbf{X}, Y)$ is used to denote the bias of $\hat{f}$. The subscript $(\mathbf{X}, Y)$ means that bias is related to both $\mathbf{X}$ and Y.

3.1 *Why bias correction*

In practice, we typically have only one data set available. In this situation, we actually want to minimize the risk:

$$R[\hat{f}(x)] = \text{Bias}^2[\hat{f}(\mathbf{x})] + \text{Var}[\hat{f}(\mathbf{x})]. \tag{4}$$

The random forest mechanism usually reduces the variance term. In many situations, the dominating component in the risk (4) turns out to be the squared bias. This is especially true when we calculate the out-of-bag (OOB) predictions.

For each tree k, there is a portion of the training set that is OOB denoted by T_k^{OOB}. Therefore, there are k_i trees that do not use observation i during their construction. Averaging predictions at x_i over these trees, we get the random forest OOB prediction given by

$$\hat{f}^{\text{OOB}}(\mathbf{x}_i) = k_i^{-1}\sum_{k=1}^{K}\hat{f}_k(\mathbf{x}_i)I((y_i,\mathbf{x}_i) \in T_k^{\text{OOB}}), \tag{5}$$

where K is the total number of trees, $\hat{f}_k(\mathbf{x}_i)$ is the prediction of $f(\mathbf{x}_i)$ from the kth tree and $I(\cdot)$ is the indicator function.

For OOB predictions of a finite sample, the risk is dominated by the squared bias term. Lin and Jeon [10] showed that once a random forest is constructed, the prediction at a new point $\mathbf{x}_i$ is a weighted average of the $y_j (i \neq j)$ values in the training data set, although in their paper no bootstrapping was performed. Consequently, OOB predictions of large values of $f(\mathbf{x}_i)$ tend to have negative bias while OOB predictions of small values of $f(\mathbf{x}_i)$ tend to have positive bias.

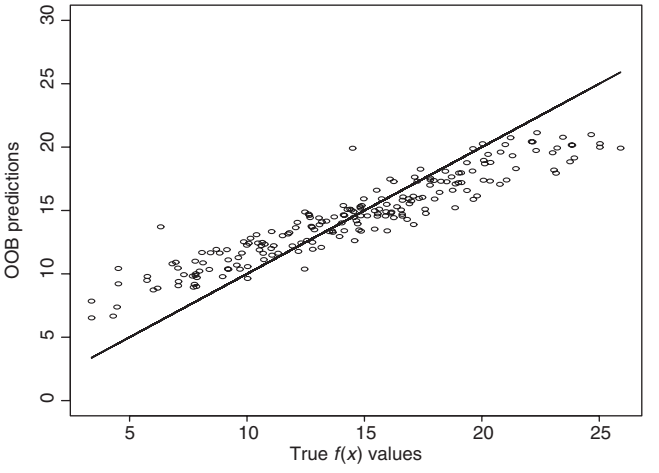

Figure 1. Random forest OOB predictions for the 200 simulated observations. The small circles represent the OOB random forest predictions. Solid line is a 45° line

To illustrate the bias problem, we generated 200 points $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5})^{\mathrm{T}}$, where each x_{ip} is independently distributed uniformly in [0, 1] and ϵ is the standard normal. The response was formulated in Equation (8).

We ran the random forest algorithm under the default settings. For each observation, we calculated the OOB random forest prediction. Figure 1 displays the OOB predictions for every observation versus the true function values. As the function values increase or decrease, the squared bias in the point estimates becomes larger. Large values of f have positive bias while small values of f have negative bias.

3.2 *Methods of estimating bias*

Intuitively, the bias occurs because of the effect of "regression to the mean". Extreme values of a single measurement are, on average, more extreme than their underlying true values. In our research, we want to discover the effects of different methods: bias attributes to both of the predictor and response variables (Method 1), bias attributes to response variables (Method 2), bias attributes to predictor variables (Method 3) and the remaining two methods (Methods 4 and 5), which can be considered as a combination or sequential bias correction from Methods 2 and 3. Because response variables and predictor variables are highly correlated, we conjecture that the five different methods should not be significantly different, but Method 1 would fit better than the other four methods. Our simulation studies confirm this.

Bias is defined as

$$B(\mathbf{X}, Y) = E[\hat{f}(\mathbf{X})] - f(\mathbf{X}). \tag{6}$$

The subscript $(\mathbf{X}, Y)$ is used to indicate that bias is related to both $\mathbf{X}$ and Y.

The general model (1) can be written as the following:

$$Y - E(\hat{f}(\mathbf{X})) = -B(\mathbf{X}, Y) + \boldsymbol{\varepsilon}. \tag{7}$$

In practice, we replace $E(\hat{f}(\mathbf{X}))$ in Equation (7) by $\hat{f}(\mathbf{X})$. The left-hand side of Equation (7) becomes the residual term with $e = Y - \hat{f}(\mathbf{X})$. The following is our proposed Model 1,

$$\text{Model 1:} \quad e = -B(\mathbf{X}, Y) + \epsilon.$$

Two reduced models of Model 1 are considered: (a) bias is in the form of a function related to response Y

$$\text{Model 2:} \quad e = -B(Y) + \epsilon$$

and (b) bias is in the form of a function related to predictor variable $\mathbf{X}$,

$$\text{Model 3:} \quad e = -B(\mathbf{X}) + \epsilon.$$

For Models 1 and 3, we use random forests to estimate the bias. For Model 2, we use univariate smoothing methods to estimate the bias. Models 4 and 5 are a combination of Models 2 and 3. Specifically, for Model 4, we first use Model 2 to estimate $B(Y)$. This results in $e^* = Y - \hat{f} + \hat{B}(Y)$. We then use e^* as the residual term from Model 3 to get $\hat{B}(\mathbf{X})$. The similar method applies to Model 5 by using Model 3 first and then Model 2. The bias-corrected random forest estimators from the five methods are listed below:

BC1 : $\hat{f}_{\text{BC1}} = \hat{f} - \hat{B}(\mathbf{X}, Y)$,
BC2 : $\hat{f}_{\text{BC2}} = \hat{f} - \hat{B}(Y)$,
BC3 : $\hat{f}_{\text{BC3}} = \hat{f} - \hat{B}(\mathbf{X})$,
BC4 : $\hat{f}_{\text{BC4}} = \hat{f} - \hat{B}(Y) - \hat{B}^*(\mathbf{X})$
and
BC5 : $\hat{f}_{\text{BC5}} = \hat{f} - \hat{B}(\mathbf{X}) - \hat{B}^*(Y)$.

4. Experiments

In this section, we use simulated data and real data to study the performance of the proposed bias-corrected methods. All computations were performed using the R software. Random forest estimators were computed using the R package randomForest [9] and nonparametric smoothing estimators were cubic smoothing splines (smooth.spline).

4.1 *Loss and test error comparisons*

To measure the performance of the estimators, we use squared error loss in estimating f by

$$\text{L}(\hat{f}) = \frac{\sum_{i=1}^{n} (f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i))^2}{n}.$$

Another commonly used measurement is mean-squared test set error defined as

$$\text{PE}(\hat{f}) = \frac{\sum_{i=1}^{n} (y_i - \hat{f}(\mathbf{x}_i))^2}{n}.$$

The loss measures the precision of the estimators for training data and the test error is a measure of prediction precision. The loss measure is widely used in nonparametric smoothing and the test error is often used for machine-learning problems. Experimental results will be reported by giving the average and standard deviation of the observed losses and test errors across replications.

We use the following regression function, Friedman #1, which has been used in [4,5,7]

$$f(x) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon, \tag{8}$$

where ε is the standard normal and x_i are uniformly distributed between 0 and 1 for $i = 1, 2, \ldots, 10$.

We generate 100 replicate random samples of size 200 for training and 100 replicate random samples of size 1000 for testing. For each sample, we then applied all six estimators (random

Table 1. Loss comparisons: the resulting average losses are listed with standard deviation in parentheses.

	RF	BC1	BC2	BC3	BC4	BC5
OOB	6.1444 (0.3879)	3.6256 (0.4416)	3.9311 (0.4625)	3.6195 (0.4283)	3.3441 (0.4525)	3.2893 (0.4408)
Not-OOB	1.4007 (0.1245)	0.7804 (0.0831)	0.8755 (0.0930)	0.7586 (0.0818)	0.7966 (0.0846)	0.7790 (0.0824)

Notes: BC1–BC5 means the bias-corrected Methods 1–5; function used is Friedman #1; RF means random forest method.

Table 2. Test error comparisons: the resulting test set errors are listed with standard deviation in parentheses.

	RF	BC1	BC2	BC3	BC4	BC5
OOB	7.0908 (0.6101)	4.0503 (0.4177)	4.9608 (0.4853)	4.0561 (0.4171)	3.9298 (0.3894)	3.8975 (0.3940)
Not-OOB	6.9165 (0.6071)	5.2207 (0.5079)	5.5037 (0.5528)	5.3015 (0.4986)	5.2059 (0.5210)	5.0428 (0.4813)

Notes: BC1–BC5 means the bias-corrected Methods 1–5; function used is Friedman #1; RF means random forest method.

Table 3. Data set summary: Friedman #1, #2 and #3 are synthetic data sets and the other three are real data sets available at the UCI repository.

Data set	Inputs	Train size	Test size
Friedman #1	10	200	2000
Friedman #2	4	200	2000
Friedman #3	4	200	2000
Boston housing	13	456	50
Servo	4	150	17
Abalone	8	3133	1044

forest estimator and BC1–BC5) and calculated the losses. The resulting average losses are listed in Table 1 with standard deviation in parentheses. From Table 1, we see that the bias correction methods work very well compared with random forests by a noticeable decrease in the loss. From Table 1, we also see loss from OOB of Model 1 is 3.6256, while loss from not-OOB of Model 1 is only 0.7804. OOB prediction is intended for estimating the test error and is not good for estimating the true function. The test set errors are summarized in Table 2. Bias correction methods still work very well compared with random forests by a noticeable decrease in the test error. It is interesting to observe that both OOB and not-OOB predictions are good for testing (prediction). If our goal is to minimize the prediction error, both OOB and not-OOB predictions can be used. If we want to find the estimator of the regression function, not-OOB is preferred. Our test errors outperform all results from Breiman [4,5]. It is pretty clear that the random forest estimators have considerable bias to be corrected.

We also investigated the performance of the estimators on more data sets. Six data sets listed in Table 3 from Breiman [4,5] are used in our experiments, among which, three (Friedman #1, #2 and #3) are synthetic data sets and three (Boston Housing, Abalone and Servo) are real data sets. Friedman #1, #2 and #3 originated in [7] and are also described in [3]. All the experimental settings are the same as those in [5]. The test errors are listed in Table 4. Except for Friedman #2, our results outperform those reported in [4,5]. From Table 4, we can see that our proposed bias-corrected methods are statistically significant in reducing test errors for all data sets, except for Abalone. The noise level of Abalone's function is too large, or of the same order as the bias and variance, which prevents the effective prediction of the true function and effective bias correction. For such cases, the underlying reason is that random forests do not work, neither do bias-corrected methods.

4.2 *Further investigation on noise levels and number of redundant variables*

In this section, we investigate the influence of noise levels and the number of redundant variables on prediction error. We set the noise level $\sigma = 0.1$, 1 and 10 and the number of predictor variables $P = 5, 10$ and 30. Regression function (8) is used in our experiments. The first five predictor variables are related to the function and all other variables are not. We use OOB predictions and calculate test set errors. The results are listed in Tables 5–7. From the tables, we see that our bias-corrected methods significantly reduce the test errors, except for the cases with noise level ($\sigma = 10$), where the large noise prevents the effective prediction of the true function and effective bias correction. MSPE is dominated by the noise variance term. The decrease in the predictor bias from bias-corrected methods is relatively small compared with the large noise variance. So we see from Tables 5–7, when $\sigma^2 = 10$, that there is almost no effect of bias correction. Another good property we found from the tables is that our bias-corrected methods are robust to redundant predictor variables.

Table 4. Test error comparisons on more data sets: the resulting test set errors are listed with standard deviation in parentheses.

Data set	RF	BC1	BC2	BC3	BC4	BC5
Friedman #1	6.9311 (0.5636)	3.9119 (0.3429)	4.8253 (0.4994)	3.9211 (0.3385)	3.8027 (0.3847)	3.7913 (0.3341)
Friedman #2	37.4939 (3.5766)	24.5941 (1.7737)	28.2378 (5.7259)	26.0741 (1.9716)	26.8619 (5.8568)	26.0835 (5.3444)
Friedman #3	32.5837 (4.0842)	20.3772 (2.2929)	30.4835 (2.3960)	21.9667 (2.3419)	28.5088 (2.3934)	24.9259 (6.8226)
Boston housing	10.3928 (6.5644)	8.2445 (4.6782)	10.2221 (7.1504)	8.5086 (4.6887)	8.9853 (5.3329)	8.9198 (5.3999)
Servo	50.0805 (12.6808)	21.8276 (10.2899)	31.4697 (13.9336)	26.5004 (10.3183)	23.0648 (10.9213)	24.0422 (11.4242)
Abalone	4.5481 (0.2736)	4.6835 (0.3023)	4.5595 (0.2573)	4.8478 (0.2898)	4.8755 (0.2923)	4.8254 (0.2788)

Note: RF means random forest method.

Table 5. Test error comparisons ($P = 5, \sigma = 0.1, 1, 10$): function used is Friedman #1, number of predictors is 5 and noise level is 0.1, 1 and 10, respectively.

σ	RF	BC1	BC2	BC3	BC4	BC5
0.1	5.4245 (0.5477)	2.0687 (0.2559)	2.7184 (0.3857)	2.2933 (0.2938)	1.9938 (0.2962)	2.0952 (0.3103)
1	6.4017 (0.5056)	3.2716 (0.2861)	3.8445 (0.4253)	3.3979 (0.3151)	3.2085 (0.3052)	3.2635 (0.3075)
10	111.3858 (5.1854)	116.3353 (5.5240)	113.0989 (6.8184)	121.0355 (5.9514)	120.3359 (6.9882)	118.0435 (7.6018)

Note: The resulting test set errors are listed with standard deviation in parentheses; RF means random forest method.

Table 6. Test error comparisons ($P = 10, \sigma = 0.1, 1, 10$): function used is Friedman #1, number of predictors is 10 and noise level is 0.1, 1 and 10, respectively.

σ	RF	BC1	BC2	BC3	BC4	BC5
0.1	6.0700 (0.5316)	2.8290 (0.3109)	3.7831 (0.4090)	2.8771 (0.3322)	2.6269 (0.3085)	2.6759 (0.3020)
1	7.0050 (0.5784)	3.9606 (0.3538)	4.8761 (0.5556)	3.9627 (0.3580)	3.8376 (0.3984)	3.8400 (0.4392)
10	112.5415 (5.8911)	116.3561 (6.0932)	115.1508 (6.9578)	119.7266 (6.3427)	119.7386 (7.2005)	118.6550 (7.0377)

Note: The resulting test set errors are listed with standard deviation in parentheses; RF means random forest method.

Table 7. Test error comparisons ($P = 30, \sigma = 0.1, 1, 10$): function used is Friedman #1, number of predictors is 30 and noise level is 0.1, 1 and 10, respectively.

σ	RF	BC1	BC2	BC3	BC4	BC5
0.1	7.7031 (0.7523)	3.9968 (0.5156)	5.1472 (0.6299)	4.0904 (0.5527)	3.6740 (0.5251)	3.6771 (0.4619)
1	8.6637 (0.7537)	5.0794 (0.5279)	6.0717 (0.6320)	5.1992 (0.5530)	4.7588 (0.5336)	4.7587 (0.4932)
10	114.7805 (5.2813)	118.0054 (5.4281)	116.6157 (6.2161)	119.1494 (5.4889)	119.6621 (6.2200)	118.8698 (6.4095)

Note: The resulting test set errors are listed with standard deviation in parentheses; RF means random forest method.

5. Concluding remarks

Generally, bias of the random forest estimators in regression problem is not negligible. In this paper, we proposed five model-based bias-corrected random forest methods to estimate the regression function. We consider models of bias related to predictor variable $\mathbf{X}$ and response variable Y and models of bias related to both of them. Random forests and univariate smoothing methods are used to estimate the bias. Simulated and real data studies show that the proposed methods are statistically significant in reducing test errors for all the cases where random forests work and they are robust to redundant predictor variables. These five bias-corrected methods performed similarly with Model 1 slightly better and we did not find significant differences between them. The advantage of our methods are that they are simple to use and results are promising. They can be considered as practical ways to correct the bias in regression problem.

Acknowledgements

The authors thank the referees for their helpful comments. The authors also thank the production editor and staffs for careful proofreading.

References

[1] G. Biau and L. Devroye, *On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification*, J. Multivariate Anal. 101 (2010), pp. 2499–2518.
[2] L. Breiman, *Bagging predictors*, Mach. Learn. 24 (1996), pp. 123–140.
[3] L. Breiman, *Arcing classifiers, discussion paper*, Ann. Statist. 26 (1998), pp. 801–824.
[4] L. Breiman, *Using adaptive bagging to debias regressions*, Tech. Rep. 547, University of California at Berkeley, February 1999.
[5] L. Breiman, *Random forests*, Mach. Learn. 45 (2001), pp. 5–32.
[6] M.J.R. Brence and D.E. Brown, *Improving the robust random forest regression algorithm*, Systems and Information Engineering Technical Papers, University of Virginia, 2006.
[7] J. Friedman, *Multivariate adaptive regression splines*, Ann. Statist. 19 (1991), pp. 1–67.
[8] S. Geman, E. Bienenstock, and R. Doursat, *Neural networks and the bias/variance dilemma*, Neural Comput. 4 (1992), pp. 1–58.
[9] A. Liaw and M. Wiener, *Classification and regression by randomforest*, R News 2 (2006), pp. 18–22.
[10] Y. Lin and Y. Jeon, *Random forests and adaptive nearest neighbors*, J. Amer. Statist. Assoc. 101 (2006), pp. 578–590.
[11] M.R. Segal, *Machine learning benchmarks and random forest regression*, 2004. Available at http://repositories.cdlib.org/cbmb/bench rf regn.
[12] C. Strobl, A.L. Boulesteix, A. Zeileis, and T. Hothorn, *Bias in random forest variable importance measures: Illustrations, sources and a solution*, BMC Bioinform. 8 (2007), p. 25.