Statistics and Its Interface Volume 7 (2014) 177–186

Predictor augmentation in random forests*

Ruo Xu†, Dan Nettleton, and Daniel J. Nordman

Random forest (RF) methodology is an increasingly popular nonparametric methodology for prediction in both regression and classification problems. We describe a behavior of random forests (RFs) that may be unknown and surprising to many initial users of the methodology: out-of-sample prediction by RFs can be sometimes improved by augmenting the dataset with a new explanatory variable, independent of all variables in the original dataset. We explain this phenomenon with a simulated example, and show how independent variable augmentation can help RFs to decreases prediction variance and improve prediction performance in some cases. We also give real data examples for illustration, argue that this phenomenon is closely connected with overfitting, and suggest potential research for improving RFs.

AMS 2000 subject classifications: Primary 62-07.
Keywords and phrases: Classification, Machine learning, Prediction, Regression.

1. INTRODUCTION

Random forest (RF) methodology is among many machine learning techniques useful for prediction and classification problems [4]. The popularity of random forests (RFs) is reflected by its extension and incorporation in other methodology, such as multivariate random forests [6, 15], quantile regression forests [13], enriched random forests for microarray analysis [1], random survival forests [10] and the R package "*pathwayRF*" for metabolic pathway analysis, etc. The RF approach has also been found to work well in high dimensional problems [5]. Recent theoretical results of Biau [2] provide insight about the good asymptotic performance of RFs under sparse scenario where the number of relevant predictor variables may be small compared to the total number of predictors.

A RF is a collection of classification or regression trees generated by a bootstrap procedure. Each tree is grown from an independent bootstrap resample until all nodes contain observations no more than a pre-specified maximal node size. No pruning is done, unlike in the case of a single tree. Each tree in the forest then provides a prediction of a response variable of interest, and a single overall prediction is obtained by taking a weighted average over the tree predictions from the forest. In a regression problem, equal weights are assigned to every tree (i.e., sample average); in a classification problem, the class predicted by the most trees is taken as the prediction. This numerical approach of growing trees in a forest through a series of bootstrap resamples and creating predictions as their averages is typically how the RF method is implemented in practice, per Breiman's original intention [4], and the methodology is an application of so-called bagging (bootstrap aggregation) to trees. The RF method has sometimes been referred to as a "black box" because its properties are difficult to study analytically [16]. However, some statistical progress has been made in clarifying RFs. Lin and Jeon showed the connection between RFs and nearest neighbor method [9, 12]. More recently, Biau derived rates of convergence for a model of RFs [2], motivated by Breiman [3], that is more amenable to theoretical study than the actual RF procedure used in practice. Despite these advances, translating and interpreting the general mechanics of RFs for prediction remains challenging.

The purpose of this paper is to investigate a feature of RFs that many new users of the method would find counterintuitive; namely, *out-of-sample* predictions can often be *improved* by augmenting an original dataset with random explanatory (or predictor) variables, created independently of all variables in the original dataset. This is entirely different from the standard linear regression scenario where better in-sample predictions (e.g., higher R^2 values) can be obtained by simply including additional predictor variables. In that case, including random or meaningless explanatory variables usually leads to higher mean squared prediction error by increasing variation without reducing bias. However for RFs, this type of data augmentation can induce smaller mean squared errors (MSEs) in regression, and lower misclassification rates in classification, when predicting outside of the data used to develop the forest.

We introduce this phenomenon with an example involving simple regression. Consider a training sample of 100 iid observational pairs (X_1, Y) with $X_1 \sim U(0,1)$ and $Y|X_1 \sim N(X_1, 0.3^2)$. Suppose that an independent test case is drawn from the joint distribution of (X_1, Y), and that we wish to predict the response Y using knowledge of its X_1 value and a RF built from the 100 training cases. Applying the RF methodology, without using knowledge of

*This work was partially supported by National Science Foundation-Plant Genome Award 0922746 and NSF grant DMS-0906588. The authors are grateful to an Associate Editor and two reviewers for their careful reading and thoughtful suggestions which helped to clarify and improve the manuscript.
†Corresponding author.

the joint distribution of (X_1, Y), gives a prediction MSE of 0.132 (approximated from 1,000 simulations). When repeating this entire process with datasets obtained by augmenting (X_1, Y) with a second predictor variable $X_2 \sim U(0, 1)$, generated independently of (X_1, Y), the RF method using both X_1 and X_2 as predictors interestingly provides approximately a 12% reduction in MSE compared to the RF method without X_2.

The improvement in prediction by augmenting a dataset with an independent predictor is not limited to regression problems. As an example that a similar phenomenon can occur in classification problems, consider a binary response variable $Z = I(Y > 0.5)$ defined in the context of our previous example. When predicting the class of Z (0 or 1) using a RF, the average correct classification rate approximated from 1000 simulations increased from 68% to 73% when the single predictor X_1 was augmented with the independent predictor X_2.

These simple examples have been constructed for illustration purposes and are not meant to represent scenarios where RFs would typically be applied. Nonetheless, the examples do show that augmentation can lead to substantial improvement in prediction performance in some cases. Furthermore, we have found that it is possible to obtain better test case prediction by augmenting real datasets with independent random predictors. Because better predictions can result from including explanatory variables unrelated to the response, not all variables that improve out-of-sample RF predictions should be regarded as important. While no universally accepted criterion of variable importance exists, a commonly held belief is that a predictor variable is important if prediction performance can be improved by its presence and harmed by its absence ([5], pp. 229–30; [14], pp. 1644). Our results indicate that this variable importance criterion may be misleading at times when RFs are used. This is a key point that should be brought to the attention of the increasing number of users of RF methodology. Furthermore, the fact that such augmentation can improve RF predictions also suggests potential for further improvements to RF methodology.

The structure of this paper is as follows. In Section 2, we explain the reasons behind the improvement of RFs by predictor variable augmentation. We tie this feature to the weight selection used by RFs to obtain weighted averages of training responses and, more specifically, to the issues of weight-spreading and overfitting. In Section 3 we provide two real data examples, one for regression and one for classification, to illustrate the effect of predictor augmentation in analyses by RFs. In Section 4, we discuss some implications and generalizations of our findings and connect our work to some other known results about RFs [12]. Section 5 provides some concluding and qualifying remarks about data augmentation in RFs.

It is well known, and will be explained in the following, that introducing certain types of randomness into the RF *procedure* may improve predictions (e.g., defining each tree from a bootstrap resample or defining a split in a tree by a random selection of only a subset of predictor variables [4]. The phenomenon that we discuss here is different. We are not altering the RF procedure itself, but rather examining the impact of augmenting an initial data set with meaningless predictor variables. It is statistically counterintuitive that such augmentation can improve the performance of a serious prediction method, but nonetheless such improvements are possible with RFs. Although we demonstrate the phenomenon of prediction improvement with augmentation, we do not attempt to develop a new machine learning method (i.e., alter the RF procedure), nor do we aim to suggest a practical way to improve the prediction performance by RFs. Instead, by illustrating an effect of data augmentation, we hope to again suggest some caution in interpreting predictor variables which improve RF predictions.

2. IMPROVED PREDICTIONS VIA DATA AUGMENTATION WITH INDEPENDENT EXPLANATORY VARIABLES

2.1 Examination of a simulated example

To explain the improvements to RFs by independent predictor augmentation as alluded to in Section 1, we conducted a more elaborate simulation study with 1,000 simulation runs, where each run used the following data-generating procedure. In each particular simulation, we generated a training sample with 101 cases (X_1, Y), created from a nonrandom explanatory variable, equally partitioning the interval $[0, 1]$ as $X_1 = i/100, i = 0, \ldots, 100$, and a response variable Y generated as $Y|X_1 \sim N(X_1, 0.3^2)$. This **o**riginal dataset will be referred to as *dataset-O*. For the purpose of comparison, we created another dataset by **a**ugmenting dataset-O with an independent variable $X_2 \sim U(0, 1)$ in the same simulation run. We refer to this second dataset as *dataset-A*. Two RFs were grown from the datasets with or without the X_2 variable (denoted as RF-A and RF-O, respectively); each forest had 100 fully grown trees with a maximal node size of 1. Note that the two datasets in each simulation run shared exactly the same X_1 values and Y values, with the only difference between the datasets being whether or not X_2 was present. Additionally, a tree in RF-O was grown using the same bootstrap index as a corresponding tree in RF-A, which helped to reduce variability in the simulation study induced by bootstrap resampling.

We used RF-O and RF-A to predict responses at four values of $X_1 = 0$, 0.25, 0.5, 0.75. To evaluate prediction performance, within the same simulation run, we generated 10 independent response cases at each X_1 value and, separately for each X_1 value, computed the average squared prediction errors and biases between the RF predictions and actual test responses. Averaging these (average) prediction errors

Table 1. Predictions by the RFs grown from the original (RF-O) and augmented (RF-A) datasets

Test case X_1 values	Prediction MSE		Estimated Bias	
	RF-O	RF-A	RF-O	RF-A
$X_1 = 0$	0.1323	0.1189	0.008	0.078
$X_1 = 0.25$	0.1328	0.1112	0.003	0.017
$X_1 = 0.5$	0.1320	0.1073	-0.001	-0.003
$X_1 = 0.75$	0.1287	0.1081	-0.003	-0.012

and biases produced the MSEs and estimated biases listed in Table 1. As Table 1 shows, augmenting with an irrelevant explanatory variable $X_2 \sim U(0,1)$ reduced the prediction MSE at each of the four test sample X_1 values. The improvement was more obvious for the predictions with an X_1 value in the middle of this variable's range $[0,1]$, rather than at the edges.

To begin to understand the results in Table 1, we recall that, as mentioned for the regression case, a prediction by a RF amounts to an average of the predictions produced by the trees in the forest, where each tree is grown from an independent bootstrap resample of the original data. Because each tree prediction corresponds to some average of the responses $Y_1, \ldots, Y_n$ observed in the original training data (i.e., the average of responses in a node from the dataset used to grow the tree), we can view the final prediction of the RF (at some given level of explanatory variables $\boldsymbol{x}_0$) as a convex combination of the training responses

$$\hat{Y}(\boldsymbol{x}_0) = \sum_{i=1}^{n} w_i(\boldsymbol{x}_0) Y_i \tag{1}$$

involving nonnegative weights $w_i \equiv w_i(\boldsymbol{x}_0)$ with $\sum_{i=1}^n w_i = 1$. The weights w_i are functions of the training sample and the regressor value of the test case $\boldsymbol{x}_0$. A single tree in the forest is grown by a series of partitions of the regressor space (i.e., binary splits), which tend to pool data cases with similar regressors in the same nodes. As a result, a RF predicts a new case by selecting training cases over each tree that are close in terms of the explanatory variables, essentially producing a weighting scheme $w_1, \ldots, w_n$ that attempts to put more weight on responses $Y_1, \ldots, Y_n$ in the training dataset with explanatory variables that match those at which a prediction is desired. RFs created with or without augmentation predictor variables (i.e., RF-A or -O) are attempting to produce weights that achieve a good prediction of the response $Y_0 \equiv Y_0(\boldsymbol{x}_0)$ at some given level of the regressors. Letting $\boldsymbol{w} = (w_1, \ldots, w_n)^T$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$, the quality of a predictor $\hat{Y} \equiv \hat{Y}(\boldsymbol{x}_0) = \sum_{i=1}^n w_i Y_i = \boldsymbol{w}^T \boldsymbol{Y}$ of an independent response Y_0 in terms of MSE, given by

$$\begin{aligned} E(Y_0 - \hat{Y})^2 &= Var(Y_0) + Var(\hat{Y}) + [E(\hat{Y}) - E(Y_0)]^2 \\ &= Var(Y_0) + Var(\boldsymbol{w}^T \boldsymbol{Y}) + [E(\boldsymbol{w}^T \boldsymbol{Y}) - E(Y_0)]^2, \end{aligned} \tag{2}$$

depends on the weights $\boldsymbol{w}$ through the variance $Var(\hat{Y})$ and bias $E(\hat{Y}) - E(Y_0)$ of the predictor $\hat{Y} = \boldsymbol{w}^T \boldsymbol{Y}$. As in other regression problems, a trade-off exists in the RF method between prediction bias and variance, which are induced in this case by the selection of weights.

For the same simulation study that produced the prediction MSEs in Table 1, we can closely examine the weights (1) assigned to the training cases in a RF construction. Recall that each of the 101 training observations in this study corresponds to a unique value of a non-random explanatory variable $X_1 = i/100, i = 0, \ldots, 100$, chosen to equally partition the interval [0, 1]. Hence, from 1,000 simulation runs, we determined the average value of a weight w_i assigned by a RF to a response Y_i corresponding to explanatory variable $X_1 = i/100, i = 0, \ldots, 100$, when trying to predict a new response Y generated at each of the X_1 levels 0, 0.25, 0.5, and 0.75 listed in Table 1. The results are displayed in Figure 1 for both RF-O and RF-A. Again, this figure gives an idea of how RFs with or without augmentation variables tend to select and weight training cases that are close in the regressor space to the positions at which predictions are desired. Also included in Figure 1 for comparison are the optimal weights $w_0, \ldots, w_{100} \geq 0$ with $\sum_{i=0}^{100} w_i = 1$ which minimize the prediction MSE (2); these values were computed numerically based on knowledge of the true mean response $E(Y|X_1)$ and variance σ^2, so such weights could not be used in practice. However, optimal weights are useful for comparison against RF weighting schemes.

Figure 1 illustrates the reason for the improvement to RF by independent predictor variable augmentation. When predicting a test case, RF-O tended to concentrate weights only on a few training cases with X_1 values immediately neighboring the X_1 value of the test case; in contrast, RF-A tended to spread nonzero weights on more training cases. This often led to slightly more bias but substantially less variance for RF-A predictions. For predictions of a new response at $X_1 = 0$, RF-A clearly led to more bias (see Table 1) because the weights on training cases could be only spread to training cases with a mean response greater than the mean response at $X_1 = 0$. But, in this study, bias cost much less than the gains made in increased precision, and hence a uniformly smaller MSE was obtained by RF-A. In Figure 1, the inclusion of X_2 dragged the weight assignment in RF-A from the narrow assignment of RF-O towards the optimal one. This weight-spreading effect helped to incorporate more training cases that were appropriately close to a test case (in terms of the meaningful regressor X_1) when forming a weighted average prediction.

We also tried augmenting the dataset with different numbers of independent $U(0,1)$ predictors. When the number of irrelevant predictor variables was not very large (2 to 5), the prediction MSE for RF-A was smaller than for RF-O for test cases with X_1 values away from the 0 or 1 edges of this variable's range, but the degree of improvement was less than that shown in Table 1. This further augmentation had a less substantial effect in reducing the variance of predictions, as balanced against the corresponding losses in predictor accuracy. Furthermore, as the number of irrelevant predictors

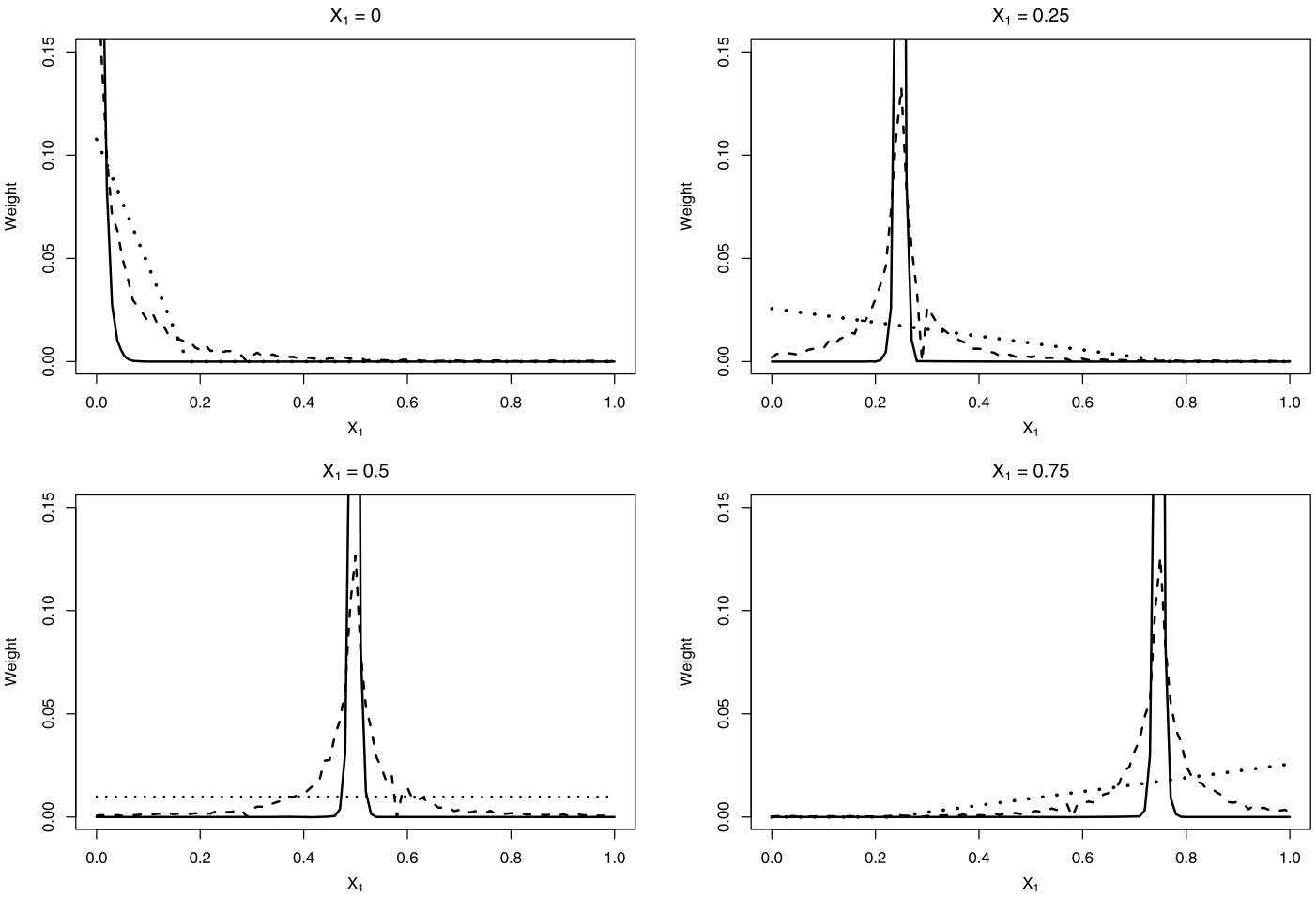

Figure 1. Average weights on responses in the training set (identified by their X_1 values) that contribute to a prediction by either RF-O (–) or RF-A (- -). The dotted line (· · ·) corresponds to the weights used by the best linear predictor of the form (1) that minimizes prediction mean squared error given in (2).

further increased (e.g., larger than five), the weights on responses in the training sample used for prediction became increasingly uniform. In these cases, bias increased and overwhelmed potential reductions in variance, leading to poor prediction performance except for predictions at X_1 values near the center of the [0, 1] interval, where a uniform weighting is optimal (results not shown).

2.2 A simple analytical example

To further describe how augmenting the design matrix affects weight-spreading in RF predictions, it is helpful to recall some details on how trees in the forest are grown. The description provided here is consistent with the default settings of the R package *randomForest* [11]. In the RF procedure, a bootstrap resample of the training sample is used to grow a tree by a series of node splits, where to determine a node split, the algorithm considers a series of binary partitions composed from a set of randomly chosen predictor variables and their values. Consistent with Breiman [4], the number of randomly selected predictors considered for each split is $\max\{1, \lfloor p/3 \rfloor\}$ for regression, where p is the total number of predictors. For each selected predictor, the distinct values observed in the bootstrap sample used for tree construction are ordered, and the midpoints separating consecutive distinct values are considered as candidate split points. For a given predictor and a split point, two nodes are subsequently defined by splitting the cases into two groups depending on whether the value of the predictor is less or greater than the split point.

The mean surface for each node/group is estimated by the average response values over training cases in the node. The tree growing procedure is a greedy one, in that it always chooses the current best partition in each step in order to minimize some loss criterion (i.e., squared error loss in regression), though this immediate best partition may not be the globally best one.

Consider the simulation of the last section, with two predictor variables (X_2 being the augmentation variable). When growing both RF-O and RF-A, only one predictor (either X_1 or X_2) is randomly selected for consideration at

each split. Because the trees in RF-O construction must split only on the right predictor X_1, a *narrow* weighting scheme is induced, and the training cases ultimately weighted to predict a new test case are very close to each other in terms of the distance between their X_1 values. On the other hand, with RF-A construction, both X_1 and X_2 have equal chances to be considered at a split. Splits based on the noninformative X_2 variable may often direct the test case to a node that does not contain the training cases closest to the test case in terms of distance between their X_1 values. Subsequent splits based on X_1 will tend to place the test case with training cases with similar X_1 values, among those training cases that have not already been split onto different branches of the tree. In this way, splits on the noninformative X_2 variable tend to diversify the tree structures and spread weights on more training cases in prediction, which are roughly close to a new test case in terms of X_1 values (even though splits on the variable X_2 are not necessarily meaningful).

As an extremely simple illustration, suppose there are two training cases, $C_1 \equiv C_1(Y_1, X_1 = 0, X_2 = x_{21})$ and $C_2 \equiv C_2(Y_2, X_1 = 1, X_2 = x_{22})$ available for predicting an independent test case $(Y_3, X_1 = 0, X_2 = x_{23})$, where x_{21}, x_{22}, and x_{23} are realizations of iid $U(0,1)$ random variables, and $Y|X_1 \sim N(X_1, \sigma^2)$, similar to our previous simulation example. Consider three different possibilities (a), (b) and (c) below for predicting the test case response.

(a) A tree is grown on the right variable X_1, with neither bagging nor predictor augmentation, which produces a prediction Y_1 with prediction MSE (2) given by $2\sigma^2$; here the prediction bias is zero and the prediction variance is $2\sigma^2$.

(b) The RF-O prediction (i.e., using only X_1) as a bootstrap expectation, based on bagging trees grown from the following resamples $\{C_1, C_1\}$, $\{C_2, C_2\}$ and $\{C_1, C_2\}$ (with probability 1/4, 1/4, and 1/2), is $3Y_1/4 + Y_2/4$ with prediction MSE given by $2/32 + 52\sigma^2/32$, where the first and second terms in the sum correspond to squared prediction bias and prediction variance, respectively. Note also here that the RF prediction, as a bootstrap expectation, was computed directly, without numerical approximation involving resampled trees.

(c) The RF-A prediction as a bootstrap expectation, based on bagging the same resampled trees as RF-O but with X_2 augmentation, is given by $Y_1/4 + Y_2/4 + M/2$, where Y_1, Y_2 and M are the predictions from resamples $\{C_1, C_1\}$, $\{C_2, C_2\}$ and $\{C_1, C_2\}$, respectively, with

$$M = \begin{cases} Y_1 & \text{if } |x_{23} - x_{21}| \le |x_{23} - x_{22}| \\ Y_1/2 + Y_2/2 & \text{if } |x_{23} - x_{21}| > |x_{23} - x_{22}|. \end{cases}$$

Note that M takes the value Y_1 whenever x_{23} is closer to x_{21} than to x_{22} because the test case will end up in the terminal node containing C_1 regardless of which variable (X_1 or X_2) is chosen for splitting. On the other hand, when x_{23} is closer to x_{22} than to x_{21}, splitting on X_1 will result in Y_1 as the predictor of Y_3, while splitting on X_2 will result in Y_2 as the predictor. Hence, in this case that x_{23} is closer to x_{22}, because the variables X_1 and X_2 are selected for splitting with equal probability, the expectation is that 1/2 the trees constructed from resample $\{C_1, C_2\}$ will predict Y_3 by Y_1 and 1/2 will predict Y_3 by Y_2, yielding $M = Y_1/2 + Y_2/2$. Because x_{21}, x_{22}, and x_{23} are iid $U(0,1)$ realizations, the two potential values for M are equally likely. The end result is that the RF-A prediction as a bootstrap expectation is an equal mixture of the linear predictors $3Y_1/4 + Y_2/4$ and $Y_1/2 + Y_2/2$. The MSE for this mixture of linear predictors is $5/32 + 50\sigma^2/32$.

We see that the RF procedure itself (RF-O) assigns more weight on Y_2, when compared to a single tree (no bagging), because bagging leads to unavailability of C_1 in some resamples. Augmenting the design matrix with an independent $U(0,1)$ variable X_2 further spreads the weights from Y_1 to Y_2, because the possibility of splitting on X_2 increases the potential for Y_2 to be used for prediction even when C_1 is in the bootstrap resample. This example intuitively explains how predictor augmentation helps to spread weights on training samples in prediction. As the noise level σ^2 increases, the RF-O method becomes preferred (with respect to MSE) over a single tree, and RF with augmentation becomes preferred over RF-O. More specifically, in this simple example, (a) has lowest MSE for $\sigma^2 < 1/6$, (b) has lowest MSE for $\sigma^2 \in (1/6, 3/2)$, and (c) has lowest MSE for $\sigma^2 > 3/2$. Note that we do compromise prediction accuracy (larger squared prediction bias) by implementing RF and additionally by predictor augmentation, but the pay-off in reduced prediction variances improves the overall MSE for larger σ^2. We will discuss this issue in Subsection 4.2.

From the above argument, we also see that the weighting scheme of training samples used for prediction crucially influences the prediction MSE, and this perspective also largely explains the prediction improvements made by RFs over single trees in the first place. When the maximal node size is one, a tree typically predicts a test case using only one training case, and corresponds to the narrowest possible weighting scheme of the training responses. Bagging then helps to broaden the spread of weights in a convex combination of training responses as a predictor (1) based on training responses as long as the number of *structurally different* trees is not too small. In a sense, this diversification of weights provides a different, though related, perspective for understanding Breiman's original argument that RFs improve prediction by combining trees that are made less "correlated" (in Breiman's words, [4]), or more diverse, by randomly selecting subsets of predictors (instead of using them all) to build trees. From a perspective of weight-spreading, similarly structured trees, even if built by resampling, will tend to narrowly focus weights on a few training responses, while the prediction by more structurally diversified trees (i.e., diversified by randomly choosing regressor variables in resampled trees and, in our case, further diversified by predictor variable augmentation) tend to positively weight a wider range of appropriate training cases and thereby reduce prediction variance. This again intuitively explains how RFs alleviate overfitting compared to an individual tree.

In the standard Monte Carlo (MC)-based implementation of a RF, where we numerically determine the RF prediction from a group of trees grown by a finite set of bootstrap resamples, we also note that there is a separate issue of using a sufficient number of trees (i.e., bootstrap resamples) to obtain a reasonable MC approximation, and the number of tree resamples can also impact the final weights used in a RF prediction in practice. However, simply increasing the number of resamples (or trees in a MC-constructed forest) only improves the MC-approximation to the weights assigned by a given RF procedure, and this does not change the structure of a RF-procedure itself (whether RF-O or RF-A). Our simulation study showed that the advantage of RF-A over RF-O in terms of prediction MSE remained unchanged when the number of resamples (trees per forest) increased from 100 to 10,000 (results not shown). We will further discuss this issue of resample sizes in Sections 3 and 4.

3. IMPROVEMENT BY VARIABLE AUGMENTATION IN REAL DATA EXAMPLES

While the previous section considered simulated data, improving test sample prediction performance of a RF by augmenting the data matrix with independent explanatory variables also occurs in real data analyses. For illustration, we first present some results based on the concrete compressive strength data of Yeh [18]. The dataset has 1,030 observations, with eight quantitative input variables and a response variable, concrete compressive strength. We performed regression analyses (predictions) by RFs based on the original and augmented datasets, and we also examined the effect of maximal node size in the trees. We created 1,000 independent partitions of the original data, where each time we randomly divided the data into a training set (with 1,000 observations) and a test set (the remaining 30 observations) and augmented the original data matrix with an independent $U(0,1)$ predictor variable as in Section 2. RF-A and RF-O were both grown with the maximal node size 1, 5, 10, 20 and 30 using the same 1,000 training cases. The performance was evaluated based on prediction MSE of the test samples, averaging over test samples across the 1000 generated partitions. The results in Table 2 indicate that predictor augmentation reduced the prediction MSE, regardless of node size and number of trees in a RF. In this example, growing each tree to its largest possible form (maximal node size 1) produced better predictions than the default setting (maximal node size 5) in the R package *randomForest*.

RFs have been shown to particularly work well in many prediction problems with a large number of predictors [1, 12, 17]. There can also be benefit if RFs are applied in some problems with a much lower dimension, potentially producing better predictions than other procedures. Using the concrete compressive strength data of Yeh as an example [18], we considered a linear regression model on all eight predictor variables and a linear regression model with LASSO regularization [7] on all first and second order terms involving the eight predictor variables (44 in total, including 8 first order terms, 8 quadratic terms and 28 interaction terms). The LASSO tuning parameter was selected by 10-fold cross validation. With the same data partitioning scheme as for RFs, the prediction MSEs for the linear regression and LASSO model were 111.05 and 68.74, respectively, much larger than any RF-A results shown in Table 2.

Table 2. Prediction MSEs in the concrete compressive strength regression with (RF-A) and without (RF-O) predictor augmentation

RF-O					
Node size	1	5	10	20	30
10 trees/RF	34.99	37.15	41.14	57.07	68.37
100 trees/RF	29.05	31.55	36.21	51.57	62.97
1000 trees/RF	29.00	31.46	36.16	51.56	62.87
RF-A					
Node size	1	5	10	20	30
10 trees/RF	34.44	35.39	38.69	50.49	60.24
100 trees/RF	28.24	29.85	33.39	45.96	55.84
1000 trees/RF	28.25	29.82	33.45	45.91	55.83

Lin and Jeon showed that controlling node size improved prediction performance by RFs in some datasets [12]. Table 2 shows that the relative performance of RF-O could not be improved either by increasing the number of trees per forest (resamples in the implementation of a RF), or increasing node size. The effect of node size and number of trees per RF will be further discussed in Subsection 4.1.

We also tested predictor augmentation in a classification problem with a real data set. Haberman reported a dataset on the survival of patients who had undergone breast cancer surgery at the University of Chicago's Billings Hospital between 1958 and 1970 [8]. The dataset has three explanatory variables: age of patient at time of operation, year of operation, and number of positive axillary nodes detected. The binary response variable is a patient's survival status five years after operation. There are 306 patients in this dataset. Again we augmented the original data matrix with an independent $U(0,1)$ random variable. The dataset was randomly partitioned into a training set with 2/3 of the patients and a test set with the remaining 1/3 of the patients. We grew both RF-O and RF-A with the same training set and predicted the response in the test set with the maximal node size 1, 5, 10, 20 and 30. This process was repeated 1,000 times, and the average correct classification rates are shown in Table 3. Augmenting the data matrix with an independent $U(0,1)$ predictor improved classification except when the maximal node size was 30. As in the previous regression problem with the concrete dataset, increasing the number of trees per forest led to little or no improvement. The default maximal node size of the *randomForest* package for classification problem is 1, which produced slightly worse

Table 3. Classification rates for Haberman's survival data with (RF-A) and without (RF-O) data augmentation

RF-O					
Node size	1	5	10	20	30
10 trees/RF	0.719	0.722	0.726	0.731	0.731
100 trees/RF	0.719	0.722	0.726	0.731	0.733
1000 trees/RF	0.721	0.724	0.728	0.733	0.735
RF-A					
Node size	1	5	10	20	30
10 trees/RF	0.732	0.733	0.733	0.732	0.732
100 trees/RF	0.730	0.729	0.733	0.732	0.733
1000 trees/RF	0.732	0.733	0.734	0.735	0.735

predictions than the larger maximal node size considered for this dataset. The effect of node size will be further discussed in Section 4.1.

4. OTHER CONSIDERATIONS IMPACTING THE EFFECT OF VARIABLE AUGMENTATION

In Subsections 4.1 through 4.3, we briefly connect the evidence of improved random forest (RF) predictions by predictor augmentation to several other aspects influencing the performance of RFs, such as the number and size of individual trees in a forest, signal-to-noise issues, the dimensionality of data, and the functional relationship between mean response and explanatory variables. We also return to the issue of interpreting variable importance in light of variable augmentation in Section 4.4.

4.1 Number and size of trees

As described in Section 2, a RF grows an ensemble of trees with bootstrap samples and thereby improves prediction (over a single, less stable tree [3]) by averaging a series of tree structures, and inducing a broader set of weights on training responses. We have seen that predictor augmentation can further add to tree diversification and weight-spreading in a RF predictor (1) and that, when such augmentation is helpful, it is because this acts to reduce prediction variance and mitigate overfitting. We have also illustrated that augmentation can improve predictions regardless of the number of trees.

Regarding tree size, the default maximal node size values in the R package *randomForest* are 5 for regression and 1 for classification problems. However, it is commonly believed that RFs work best with a maximal node size of 1. Hastie et al. have also suggested that the overfitting of RFs with fully grown trees (i.e., maximal node size of 1) seldom costs much, especially in classification problems [9]. Segal [15] and Lin and Jeon [12] demonstrated minor gains in RF regression problems by controlling the node sizes of individual trees in a forest. In particular, Lin and Jeon [12] related RFs to the adaptive k-nearest neighbor (k-NN) method, and showed that tuning the maximal node size is advantageous for the performance of RFs. There can be an advantage of choosing a maximal node size larger than one for datasets with many observations but relatively small dimension, because this also has the effect of weight-spreading to reduce the variance of RF predictions. However, tuning the maximal node size of individual trees may not be enough to avoid overfitting problems in RFs completely, and predictor augmentation can still be beneficial with maximal node sizes larger than one. Our real data analysis examples in Section 3 (Table 2 and 3) indicate that prediction performance can be further improved by $U(0,1)$ predictor augmentation at different maximal node sizes. This suggests that, when such predictor augmentation is helpful, there may yet be room for improvement in RFs, including the choice of maximal node size.

To better understand how the weight-spreading effect of random predictor augmentation differs from that of increasing maximal node size, it is helpful to consider the k-potential nearest neighbors (k-PNN) concept introduced by Lin and Jeon [12]. Based on Lin and Jeon's Proposition 1, a training case with predictor vector $\boldsymbol{x}_i$ is among the set of k-PNN of a target point $\boldsymbol{x}_0$ if and only if there are fewer than k training cases with predictor vectors in the hyperrectangle defined by $\boldsymbol{x}_i$ and $\boldsymbol{x}_0$. Furthermore, when a tree with maximal terminal node side k is constructed for a particular bootstrap resample, only bootstrap resample training cases in the set of k-PNN of the target point $\boldsymbol{x}_0$ can end up in the terminal node containing $\boldsymbol{x}_0$. Thus, the response for training cases whose predictor vector is relatively far from $\boldsymbol{x}_0$ in all dimensions will receive no weight in the tree prediction of the response at $\boldsymbol{x}_0$ unless k is large.

Figure 2 provides a simple example to show how augmentation can expand the set of k-PNN to include training cases that would be considered far from $\boldsymbol{x}_0$ on the basis of the original predictor variables. In this example, only cases 1 and 2 are 1-PNN of $\boldsymbol{x}_0$ based on the original univariate predictor. However, following augmentation, the set of 1-PNN includes all cases except case 4 (which was a 3-PNN prior to augmentation and becomes a 2-PNN following augmentation). On the basis of the original predictor, case 5 could contribute to the tree prediction of the response at $\boldsymbol{x}_0$ only if the maximal terminal node size were set to 4 and only if cases 2 through 4 were also used for prediction. Following augmentation, case 5 may contribute to the tree prediction regardless of the maximal terminal nodes size and regardless of which other cases receive positive weight.

This example shows that the way trees spread weights to additional training cases when maximal terminal node size is increased is fundamentally different than the way weights are spread following augmentation. Augmentation places fewer restrictions on how weights can be spread than increasing node size because augmentation allows training cases initially considered far from the target point to be considered relatively close. As our results show, this greater flexibility to spread weights to cases has advantages, especially when the signal-to-noise ratio is low.

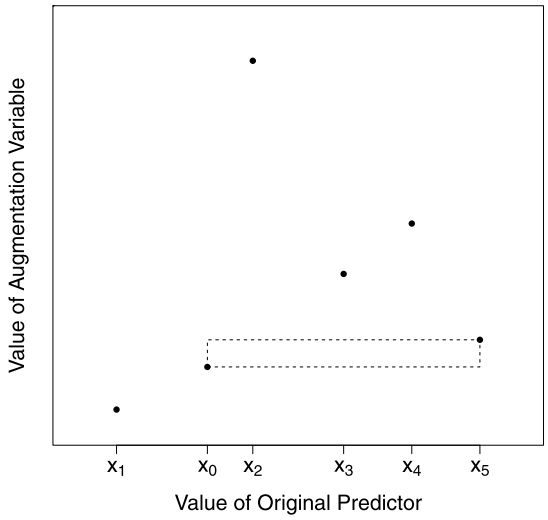

Figure 2. hypothetical example involving five training cases with original univariate predictor values x_1, x_2, x_3, x_4, and x_5. A prediction of the response at target value x_0 is desired. The dashed box shows that the hyperrectangle defined by the target case and case 5 contains no training case other than case 5 so that case 5 becomes a 1-PNN of the target after augmentation.

4.2 Signal-to-noise issues and mean response function

The prediction variance for a given test case is, of course, related to the variance of the training data. More variability in the training data often translates into larger variances for RF predictions relative to their squared biases, and hence a larger benefit by predictor augmentation and widening the weights (1) on the training sample. In the simulation experiment of Table 1, for example, if we generate the response variable $Y|X_1$ from $N(X_1, 0.5^2)$ instead of $N(X_1, 0.3^2)$, the variance/MSE reducing effect is even more obvious by augmenting with an independent X_2 from $U(0, 1)$. In contrast, if $Y|X_1$ is from $N(X_1, 0.1^2)$, there is hardly any improvement. That is to say, data sets with large signal to noise ratios benefit less from predictor augmentation.

Our simulation example in Section 2 was admittedly and intentionally simple in that the mean function of response variable was linear in the only key predictor X_1 equally partitioning the unit interval $[0, 1]$. When the test sample has regressor values that are not too extreme so that there are training cases which approximately neighbor the test case in regressor space, averaging or weighting more training responses, induced by independent predictor augmentation, leads to better prediction precision without sacrificing too much prediction accuracy. However situations can arise where augmentation can hurt the overall prediction MSE because the gains in reduced prediction variance from weight-spreading cannot offset the damage in bias. This can occur, for instance, when the mean function is highly nonlinear over small neighborhoods in the regressor space and the underlying noise is low. Consider another simple simulation study, similar to that in Section 2, where the mean function is $Y \sim \sin[N(X_1, \sigma^2)]$, $X_1 \sim U(0, 2\pi)$, and we examine the effect of augmenting with an independent $U(0, 1)$ predictor for predicting an independently drawn test case (Y_1, X_1). When σ is 0.3, RF-A and RF-O give prediction MSEs of 0.084 and 0.069, respectively. However, augmentation starts to help as the noise of responses rises, and when σ is 0.5, the prediction MSEs become 0.157 (RF-A) vs. 0.164 (RF-O).

4.3 Number of predictor variables

The number of predictor variables is another factor that affects the performance of a RF. In our experience, predictor augmentation is more effective if the predictor dimension is low. When the original dataset has many predictors, the weight-spreading effect by augmenting with independent predictors is weakened for two reasons. First, the chance that a noninformative augmentation predictor will be selected decreases at each split. Second, when the original dataset has some irrelevant or weak predictors already, the weights in (1) can be sufficiently spread by these irrelevant predictors, and an extra augmentation predictor may have little impact. For these reasons, we may not obtain smaller prediction error by augmenting a real dataset that already has many predictors.

However, it is often possible to select a subset of predictor variables and gain prediction improvement by predictor augmentation (in both regression and classification problems). For instance, in the survival classification problem on Haberman's dataset in Section 3, suppose we choose a data matrix with age of patient during operation and year of operation, but excluding the number of positive axillary nodes detected. For this reduced dataset, augmentation by an independent $U(0, 1)$ predictor increases the correct classification rate from 0.673 to 0.733, about an 8% improvement (approximated by simulation), compared to the corresponding 1% improvement in Table 3. This aspect of RF performance has some implications for interpreting variable importance, as described in the next subsection.

Biau [2] argues for consistency of RFs with a convergence rate that depends on the number of relevant predictors (referred to as strong features by Biau [2]) rather than the total number of predictor variables. Consistency here means that the expected squared difference between the RF prediction and the conditional expectation of the response, given the predictor variable values, converges to zero as the sample size n grows. The results of Biau [2] suggest that augmentation is asymptotically irrelevant. Furthermore, Proposition 2 of Biau [2] implies that the variance of RF prediction from fully grown trees is of the order $1/(\log n)^{(S/2d)}$, where S is the number of relevant predictor variables and

d is the total number of predictor variables. This expression suggests that augmentation with independent predictors (increasing d without increasing S) would slightly slow the rate at which the variance decreases to zero. At first glance, this may seem at odds with our examples of reduced variability by augmentation with irrelevant variables. However, Biau's results are for a stylized model of RFs, motivated by Breiman [3], rather than for the standard RF algorithm used in practice, because simplifying assumptions are commonly necessary in mathematical studies of RFs ([16], p. 12). In the model, bootstrap resampling is not used, and all splits for the trees in a forest are selected independently of the training data. Additionally, the mechanism for selecting variables on which to split is assumed to concentrate probability of selection on only the relevant variables as n grows. Given our focus on finite sample cases and the differences between the RF model and the RF method used in practice, the findings of Biau [2] do not rule out the improved prediction performance following augmentation that we have demonstrated in the previous sections of this paper.

4.4 Variable importance

Variable importance may have different meanings in different contexts, with no generally accepted definition. Often, a predictor variable may be regarded as important if the prediction on an independent test sample is more accurate with it, and is less accurate without it (the rejoinder of [5]; [14]). Our examples demonstrate that, in RFs, the presence of predictors independent of the response variable may reduce the prediction error, even though such variables are obviously not important in any scientifically meaningful sense. This illustrates a possible pitfall with this definition of variable importance.

Breiman [5] also proposed a second way to define variable importance as follows: if, within each resampled tree, randomly permuting the values of a certain predictor variable harms the prediction for cases not included in the given tree construction, then this variable is deemed important. This notion is embodied in the "variable importance measure" of the R package *randomForest* [11], which assigns highest values to variables with the greatest discrepancy between original prediction performance and prediction performance after permutation. (In regression problems, such prediction performance is computed by first tree-wise determining squared errors for predicting training cases left out of the resampled tree construction and then averaging all such errors over all trees. The resulting variable importance values have no meaning on an absolute scale, but their relative sizes can be useful for comparing across different predictor variables). By its construction, this second variable importance measure can distinguish independent augmentation predictors from scientifically meaningful predictors because permuting an independent augmentation variable has no impact on the joint distribution of the variables in the dataset. Consequently, permuting an augmentation variable will not substantially change predictions and will typically result in a relatively low measure of variable importance. In the simple illustrative example we employed in Section 1 with $Y|X_1 \sim N(X_1, 0.3^2)$, X_1 and X_2 iid $\sim U(0,1)$, the variable importance measures given by *randomForest* (approximated from 1,000 simulations) for X_1 and X_2 are 20.85 and 0.24, respectively. This correctly indicates that X_1 is far more important than the irrelevant variable X_2. Thus, Breiman's second criterion of variable importance is more meaningful than the first notion of variable importance when using RFs in variable selection and model building problems.

5. CONCLUSIONS AND QUALIFICATIONS

This paper demonstrated and investigated a phenomenon of RF methodology that is not well known to many users of RFs: independent predictor variable augmentation sometimes improves out-of-sample RF predictions. As RF predictions have the form of convex combinations of training responses, augmenting a dataset with an independent predictor variable can often induce a type of weight-spreading which crucially reduces the variance of predictions compared to the additional bias induced, and thereby improve the prediction performance of RFs. While it is possible that other variations of RFs may produce a similar effect, it is nonetheless important to make users of RFs aware of the impact of independent predictor variable augmentation.

As part of this effort, we offer some warning that there is a potential risk in assuming that only useful explanatory variables will contribute to better RF prediction, because augmentation with scientifically meaningless variables demonstrates otherwise. To again qualify this work, our intention here is not to suggest or recommend predictor augmentation in practice for improving RFs. However, the fact that such data augmentation can improve RF predictions at all, despite maximal node size choices or numbers of resampled trees used in numerical construction of forests, indicates that there may exist further research potential to achieve better implementations of RF methodology in practice.

Received 16 June 2013

REFERENCES

[1] Amaratunga, D., Cabrera, J., and Lee, Y.-S. (2008). Enriched random forests. *Bioinformatics* **24**(18), 2010–2014.

[2] Biau, G. (2012). Analysis of a random forests model. *Journal of Machine Learning Research* **13**, 1063–1095. MR2930634

[3] Breiman, L. (2000). Some infinity theory for predictor ensembles. *Technical Report* 577, University of California, Berkeley.

[4] Breiman, L. (2001a). Random forests. *Machine Learning* **45**, 5–32.

[5] Breiman, L. (2001b). Statistical modeling: The two cultures (with discussion). *Statistical Science* **16**(3), 199–231. MR1874152

[6] De'ath, G. mvpart: Multivariate partitioning, 1.3-1. *R Package Version*.

[7] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**(1), 1–22.
[8] Haberman, S. (1976). Generalized residuals for log-linear models. In: *Proc. of the 9th International Biometrics Conference*, Boston, pp. 104–122.
[9] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer. MR2722294
[10] Ishwaran, H., Kogalur, U., Blackstone, E., and Lauer, M. (2008). Random survival forests. *The Annals of Applied Statistics* **2**(3), 841–860. MR2516796
[11] Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News* **2**(3), 18–22.
[12] Lin, Y. and Jeon, Y. (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association* **101**(474), 578–590. MR2256176
[13] Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research* **7**, 983–999. MR2274394
[14] Olshen, R. (2010). Remembering leo breiman. *The Annals of Applied Statistics* **4**(4), 1644–1648. MR2829927
[15] Segal, M. and Xiao, Y. (2011). Multivariate random forests. *WIREs Data Mining and Knowledge Discovery* **1**, 80–87.
[16] Strobl, C., Malley, J., and Tutz, G. (2009). An introduction to recursive partitioning: Rational, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods* **14**(4), 323–348.
[17] Ye, Y., Wu, Q., Huang, J., Ng, M., and Li, X. (2013). Stratified sampling for feature subspace selection in random forests for high dimensional data. *Pattern Recognition* **46**(3), 769–787.
[18] Yeh, I. (1998). Modeling of strength of high performance concrete using artificial neural networks. *Cement and Concrete Research* **28**(12), 1979–1808.

Ruo Xu
Analyst at Google Inc.
Mountain View
CA, 94043
USA
E-mail address: xuruo.isu@gmail.com

Dan Nettleton
Professor at Department of Statistics
Iowa State University
Ames, IA, 50011
USA
E-mail address: dnett@iastate.edu

Daniel J. Nordman
Associate Professor at Department of Statistics
Iowa State University
Ames, IA, 50011
USA
E-mail address: dnordman@iastate.edu