Exploring Random Forest:
An Evaluation of Regression Model Performance through Simulation in R

May Xia and Caiqin Zhou

Abstract:

Random forest has been widely acknowledged for its model stability and its capability of handling high-dimensional datasets. In our simulation study, we compared the performance of random forest, bagging, and single decision tree models in a regression context; we also examined how the number of bootstrap samples and the number of predictors considered at each node would affect the performance of random forests. Overall, the random forest model made predictions with lower variance and mean-squared error (MSE) than the other two models did. Moreover, as the number of bootstrap samples increased, the variance and MSE of random forest predictions initially decreased and then remained constant after a large enough sample size was reached. Random forest models with fewer predictors had smaller variance and MSE. Overall, our simulation results were consistent with findings from the literature, confirming that random forest models generate predictions with lower variance than bagging and single decision tree models.

1. Background and Significance

Single decision trees are simple models that can be used for both regression and classification. They are easy to graph and interpret, yet susceptible to noises in the learning set and may yield predictions with high variance. So how can we reduce the variance and improve the stability of single decision tree models? Captivated by this question, we want to explore other tree-based methods that have lower variance and improved predictive predictive accuracy. Bagging is an advanced tree-based method that involves producing multiple trees and combining them to yield a single prediction (Breiman, 1994). Although the goal is to reduce variance by averaging multiple trees, bagged trees are often highly correlated and thus combining them may not lead to significant reduction in variance. Drawing inspirations from the random split selection method (Dietterich, 1998) and the random subspace method (Ho, 1998), Breiman proposed another tree-based method—random forests. Random forests provide improvement over bagged trees by decorrelating the trees, leading to larger reduction in variance. For our project, we are interested in comparing the predictive performance of random forests, bagged trees and single decision trees in a regression context. We also examine how different parameters (e.g., number of bootstrap samples and number of random predictors to consider at each node in the tree) within the bagging and random forest models affected their performance.

2. Methodology

2.1 Single Decision Trees

To build a single decision tree, we first need to stratifying the predictor space into J distinct and non-overlapping regions, R_1, R_2, …, R_J by minimizing the residual sum of squares (regression) or the classification error rate (classification). Recursive binary splitting is used to reduce computational cost. Then, for every observation i falling in region R_J, the predicted response is the mean or most commonly occurring response for the training observations within this region, depending on whether the outcome is quantitative or categorical. A large drawback of decision trees, however, is the high variance of predicted outcomes.

2.2 Bagging

Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of statistical learning methods. To apply bagging to decision trees, we first generate B repeated samples (i.e., bootstrap samples) from a single training set and then construct a separate regression or classification tree using each sample. The final prediction of the bagged trees is the average of the Bpredicted values (regression) or the most commonly occuring class among the Bpredicted classes (classification), and this averaging process can reduce the variance of predictions. One issue with bagged trees is that they are often highly correlated. Since averaging highly correlated predictions will not significantly reduce their variance, bagging may not lead to a substantial reduction in variance over a single decision tree.

2.3 Random Forest

Random forests improve over the bagging model by decorrelating the trees. The only difference between bagging and random forests is that at each split in the tree, random forest models only consider a fresh random sample of $m$predictors out of the full set of $p$predictors. This method allows us to have decorrelated trees and less variance in the predicted outcomes. Prior research suggests setting$m = \frac{p}{3}$for regression models and $m = \sqrt{p}$for classification models.

3. Simulation Study

3.1 Procedure

Our sample size was 500 and our number of simulations was 200. For each of the 500 units, we created a set of six predictor values (X_1,X_2, … , X_6) by drawing from six normal distributions with different means and standard deviations. We assumed a linear relationship between the predictors and an outcome variable, and calculated the expected outcome for each unit using $E(Y_i)= 1 + 2 \times X_1 + 3 \times X_2 + 4 \times X_3 + 5 \times X_4 + 6 \times X_5 + 7 \times X_6$. We randomly assigned 400 out of the 500 units to be in the training set, and the other 100 units were in the testing set. During each simulation, we first generated a new set of independently and identically distributed error terms from a normal distribution (standard deviation = 40). Then, we calculated the observed outcomes using $Y_i = 1 + 2 \times X_1 + 3 \times X_2 + 4 \times X_3 + 5 \times X_4 + 6 \times X_5 + 7 \times X_6 + \varepsilon_i$. Therefore, we had 200 simulated datasets, each of which included 500 observations. Note that we used the same set of predictor values and different error terms to generate each dataset. Therefore, each unit has the same expected outcomes across different simulated datasets.

We first compared the performance of single decision tree, bagging and random forest models using default parameter values: 500 bootstrap samples and two predictors considered at each node in the random forest model. For each simulated dataset, we fitted the three models using observations in the training set, and recorded predicted outcomes for those in the testing set. Then, we investigated how different parameters in the bagging and random forest models would affect the performance of the models. In particular, we built seven bagging models that used different numbers of bootstrap samples ($B = 5$, 10, 20, 50, 100, 200, or 500); we built 28 random forest models that differed in both the number of bootstrap samples ($B = 5$, 10, 20, 50, 100, 200, or 500) and the number of predictors considered at each node in the tree ($m =$2, 3, 4, or 5). For each simulated dataset, we fit the all the 35 models using observations in the training set, and recorded predicted outcomes for those in the testing set.

3.2 Performance Metrics

When comparing across the three different methods, we considered three performance metrics: bias, variance, and mean squared error of the predicted $\widehat{Y_i}$'s using the testing data set. We calculated the three quantities for each of the 100 $\widehat{Y_i}$'s using the following formulas:

$$bias = mean(\widehat{Y_i}) - E(Y_i)$$

$$var = \frac{\sum(\widehat{Y_i} - E(Y_i))^2}{n-1}, \text{ where } n \text{is the size of the testing set}$$

$$MSE = bias^2 + variance = (\frac{1}{N}\sum_{i=1}^{N}(\widehat{Y_i}) - E(Y_i))^2 + Var(\widehat{Y_i}).$$

3.3 Simulation Results

Across single decision tree, bagging, and random forest models with default parameter values ($B = 500$ and $m = 2$), the random forest model generated predictions with the lowest average

variance and MSE, and the bagging model generated predictions with the lowest average bias. These results matched with our expectation that random forest models would have lower variance and higher stability than bagging and single tree models. Although we did not have any predictions for MSE based on literature, our simulation results suggested that given this particular set of parameter values, the random forest model also had the best bias-variance trade-off.

	Single Decision Tree	**Bagging**	**Random Forest**
Average Bias	4.6826	3.5302	4.6963
Average Variance	482.99	162.40	108.48
Average MSE	523.87	187.81	152.40

Table 1:Comparing Performance Metrics for Candidate Models

For both bagging and random forest models, as B increased, the average variance and MSE of the predicted outcomes initially dropped at a decreasing rate; beyond a large enough Bvalue, variance and MSE no longer decreased and remained constant as Bcontinued to increase (Figure 1B & 1C). Moreover, across different random forest models, those with fewer number of predictors had lower variance and higher bias, reflecting the trade-off between bias and variance (Figure 1A).

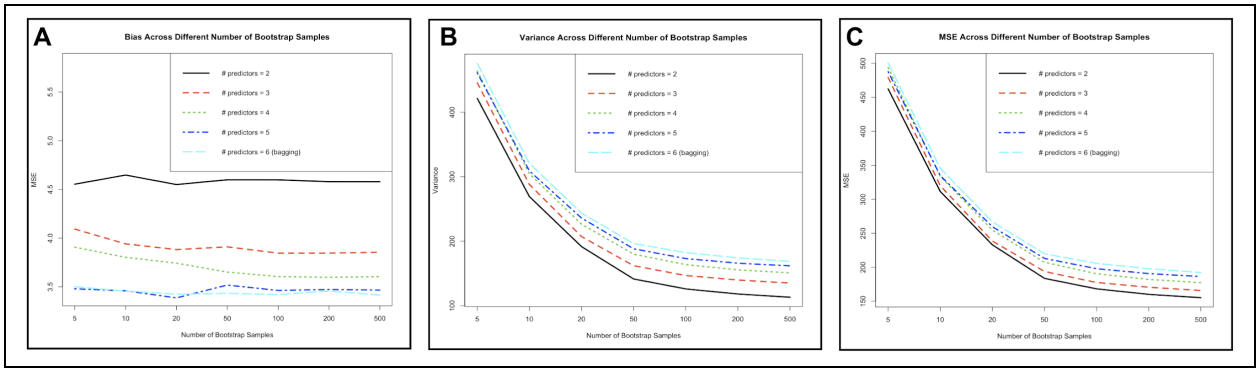

Figure 1: Comparing Bias, Variance and MSE with Different Number of Bootstrap Samples and Different Number of Predictors (Indicated by Color).

4. Conclusions

4.1 Discussion

Random forest is a powerful machine learning technique for both regression and classification. Its ability to accomodate high-dimensional feature space and model stability make it applicable to a wide range of fields including psychology, medicine, economics and bioinformatics. Through our simulation study, we formed a better understanding of the random forest algorithm and how well it performs compared to the bagging and single decision tree models. We concluded that the random forest model is the most stable among the three models and has the best bias-variance trade-off.

4.2 Further Considerations

For future work, we can investigate how the magnitude of the random error terms (i.e., the noise in the data) affects the relative performance of the models. Moreover, we may investigate the performance of the three models in a classification context.

5. References

Biau, G. & Scornet, E. (2016). A Random Forest Guided Tour. *TEST, 25* (2): 197-227.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45* (1), 5-32.

Fawagreh, K., Gaber, M. & Elyan, E. (2014). Random Forests: From Early Developments to Recent Advancements. *Systems Science & Control Engineering, 2* (1), 602-609.

Goel, E. & Abhilasha, E. (2017). Random Forest: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering, 7* (1).