Stat

The ISI's Journal for the Rapid Dissemination of Statistics Research

(wileyonlinelibrary.com) DOI: 10.1002/sta4.162

A comparison of resampling and recursive partitioning methods in random forest for estimating the asymptotic variance using the infinitesimal jackknife

Cole Brokamp[a], MB Rao[b], Patrick Ryan[a] and Roman Jandarov[b]

Received 13 July 2017; Accepted 4 September 2017

The infinitesimal jackknife (IJ) has recently been applied to the random forest to estimate its prediction variance. These theorems were verified under a traditional random forest framework that uses classification and regression trees and bootstrap resampling. However, random forests using conditional inference trees and subsampling have been found to be not prone to variable selection bias. Here, we conduct simulation experiments using a novel approach to explore the applicability of the IJ to random forests using variations on the resampling method and base learner. Test data points were simulated and each trained using random forest on one hundred simulated training data sets using different combinations of resampling and base learners. Using conditional inference trees instead of traditional classification and regression trees as well as using subsampling instead of bootstrap sampling resulted in a much more accurate estimation of prediction variance when using the IJ. The random forest variations here have been incorporated into an open-source software package for the R programming language. Copyright © 2017 John Wiley & Sons, Ltd.

Keywords: conditional inference tree; infinitesimal jackknife; prediction variance; random forest

1 Introduction

1.1 Random forest

Although random forests are commonly used in machine learning, they still remain underused for statistical inference because of a lack understanding of their statistical properties. A recent study found random forest to be the most accurate classification algorithm among 179 classifiers, based on 121 different data sets (Fernández-Delgado et al., 2014). Although proven to be more accurate, researchers are sometimes hesitant to implement random forests because they do not have parameters with a clear interpretation like regression coefficients from parametric models.

Random forest is an ensemble learning method that begins with bagging (the bootstrapped aggregation of regression tree predictions) in order to reduce the variance of the prediction function. Here, a bootstrapped sample refers to new sample taken from the original sample with replacement. Regression trees are low in bias, and although they have high variance, bagging stabilizes the predictions leading to lower variance than using one tree (Hastie et al., 2005).

[a]Division of Biostatistics and Epidemiology, Cincinnati Children's Hospital Medical Center, Cincinnati, OH 45229, USA
[b]Department of Environmental Health, University of Cincinnati, Cincinnati, OH 45220, USA
*Email: cole.brokamp@cchmc.org

The bagging procedure was modified by Brieman (1984) to use a bootstrap sample for each tree and to also select a random subset of predictors for testing at each split point in each tree. This decorrelates individual trees, further reducing the ensemble prediction variance.

The specific algorithm for random forest as used for regression is as follows:

1. For $b = 1$ to B total trees:
   - Draw a bootstrap sample from the training data
   - Grow tree T_b by repeating the following steps for each terminal node of the tree until the desired node size is reached:
     - Randomly select m_{try} of the total p variables.
     - Pick the best variable and split point from the m_{try} variables based on the best reduction in the sum of the squared errors of the predictions.
     - Split the node into two daughter nodes.
2. Output the total ensemble of all trees.
3. To predict at a new point x, average the prediction of all trees: $\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

The performance of random forests can be tuned using two parameters, m_{try} and B. B is the total number of trees and is set to 500 by default in the `randomForest` package within `R`. The number of trees should be large enough so that the error rate is stabilized. Because the random forest is grown one tree at a time, the error rate can be plotted as a function of the number of trees to visually ensure that enough trees are being used. m_{try} usually has more effect on the ensemble accuracy and is set to $\max\{\text{floor}(\frac{1}{3}p), 1\}$ as the default in the `randomForest` package within `R`. Variations in m_{try} can be auditioned, and the value producing the lowest error can be used in the final random forest model.

1.2 Estimating the variance of bagged tree predictions using the jackknife

Bootstrap sampling, subsampling and the jackknife all rely on estimating the variance of a statistic by using the variability between resamples rather than using statistical distributions. The ordinary jackknife is a resampling method useful for estimating the variance or bias of a statistic. The jackknife estimate of a statistic can be found by repeatedly calculating the statistic, each time leaving one observation from the sample out and averaging all estimates. The variance of the estimate can be found by calculating the variance of the jackknifed estimates:

$$\hat{V}_J = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\theta}_{(-i)} - \hat{\theta}_{(\cdot)}\right)^2, \tag{1}$$

where n is the total sample size, $\hat{\theta}_{(-i)}$ is the statistic estimated without using the ith observation and $\hat{\theta}_{(\cdot)}$ is the average of all jackknife estimates.

The ordinary jackknife is extended for use with bagging by applying it to the bootstrap distribution (Efron, 2014). Instead of leaving out one observation at a time, the existing bootstrap samples are used and the statistic is calculated on the basis of all resamples that do not use the ith observation:

$$\hat{V}_{JB} = \frac{n-1}{n}\sum_{i=1}^{n}\left(\bar{t}^*{}_{(-i)}(x) - \bar{t}^*{}_{(\cdot)}(x)\right)^2, \tag{2}$$

(wileyonlinelibrary.com) DOI: 10.1002/sta4.162

The ISI's Journal for the Rapid Dissemination of Statistics Research

where $\bar{t^*}_{(-i)}(x)$ is the average of $t^*(x)$ over all bootstrap samples not containing the ith example and $\bar{t^*}_{(\cdot)}(x)$ is the mean of all $\bar{t^*}_{(i)}(x)$.

1.3 Infinitesimal jackknife

As opposed to the jackknife and the jackknife after bootstrap, where the behaviour of a statistic is studied after removing one or more observations at a time, the infinitesimal jackknife (IJ) looks at the behaviour of a statistic after down-weighting each observation by an infinitesimal amount (Jaeckel, 1972). Applied to a bagged predictor, the non-parametric delta method estimate of variance for an ideal smoothed bootstrap statistic is (Efron, 2014)

$$\hat{V}_{\mathrm{IJ}} = \sum_{j=1}^{n} \mathrm{cov}_j, \tag{3}$$

where cov_j is taken with respect to the resampling distribution. Wager et al. (2014) have recently extended this idea by applying the IJ to random forest predictions. On the basis of using subsamples rather than bootstrap samples, they have shown that the variance of random forest predictions can be consistently estimated. Here, the IJ variance estimator is applied to the resampling distribution for a new prediction point:

$$\hat{V}_{\mathrm{IJ}} = \sum_{i=1}^{n} \mathrm{Cov}_* \left[T(x; Z_1^*, \ldots, Z_n^*), N_i^* \right], \tag{4}$$

where $T(x; Z_1^*, \ldots, Z_n^*)$ is the prediction of the tree T for the test point x based on the subsample $Z_1^*, \ldots, Z_n^*$ and N_i^* is the number of times Z_i appears in the subsample. Furthermore, random forest predictions are asymptotically normal given that the underlying trees are based on subsampling and that the subsample size s scales as $s(n)/n = o(\log(n)^{-p})$, where n the is number of training examples and p is the number of features (Wager & Athey, 2017).

Because $\hat{V}_{\mathrm{IJ}}$ is calculated in practice with a finite number of trees B, it is inherently associated with Monte Carlo error. Although this error can be decreased by using a large B, a correction has been suggested (Wager et al., 2014):

$$\hat{V}_{\mathrm{IJ}}^{B} = \sum_{i=1}^{n} C_i^2 - \frac{s(n-s)}{n} \frac{\hat{v}}{B}, \tag{5}$$

where $C_i = \frac{1}{B} \sum_{b=1}^{B} (N_{bi}^* - s/n)(T_b^* - \bar{T}^*)$ and $\hat{v} = \frac{1}{B} \sum_{b=1}^{B} (T_b^* - \bar{T}^*)^2$. This is essentially a Monte Carlo estimate of Equation (4) with a bias correction subtracted off. These estimates are asymptotically normal given a few key conditions, one of which is that the underlying trees are honest. Simulation experiments using sub-bagged random forests have shown that these variance estimates are biased (Wager & Athey, 2017), but the implementation of honest trees within a sub-bagged tree ensemble and its resulting prediction variance has not been studied.

1.4 Using honest trees in random forests

Athey & Imbens (2016) (and Wager & Athey (2017) within the context of random forests) define an honest tree as one in which the distribution of the predicted outcome, conditional on the explanatory variables, does not depend on the training labels. The most popular recursive partitioning algorithm and the one used in the random forest algorithm is classification and regression trees (CART) (Breiman et al., 1984). In the case of regression, this algorithm performs a search for the best possible split over all split points of all variables by minimizing the sum of squared errors between the predicted and actual values. CART are not honest because they use the same training data to both choose the tree splits and to make predictions. In contrast, conditional inference (CI) trees (Hothorn et al., 2006) are trees that are honest because they use outcomes to make predictions but use another method to find split points. CI trees implement a test statistic, like the Spearman correlation coefficient, Student's t-test or F statistic from ANOVA to pick the predictor that is most associated with the outcome based on the smallest p-value. The p-values are generated using a permutation test framework first laid out by Strasser & Weber (1999) in which the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points. To find the best split point, the standardized test statistic is then maximized. Strobl et al. (2007) showed that implementing CI trees within a random forest framework alleviates variable selection bias, which favours splitting on variables with more levels or a larger continuous range.

Each tree in the random forest algorithm is built on a resample of the original sample. By convention, the random forest uses a bootstrap sample, with size equal to the original sample size, n. Strobl et al. (2007) found that using subsampling instead of bootstrap sampling to create individual trees within a random forest also helps to reduce variable selection bias. We hypothesize that these two bias sources may also cause biased estimation of $\hat{V}_{\mathrm{IJ}}^{B}$ and explore variations on random forests that eliminate the variable selection bias to see if they perform well with the IJ variance estimator.

1.5 Overview

Although Wager et al. (2014) and Wager & Athey (2017) have proven that the IJ can be used to estimate the prediction variances of traditional random forests, their methods have not been tested using alternative individual tree types used in a random forest, such as CI trees. This variation is widely used and is important for eliminating variable selection bias. Here, we explore the applicability of the IJ to random forest variations, specifically using subsampling instead of bootstrap sampling and using CI trees instead of CART, and compare the accuracy of their estimates of prediction variances using simulation experiments.

2 Methods

2.1 Data simulation

Ten different predictor variables ($X_1, \ldots, X_{10}$) were generated by sampling from the normal distribution, with $X_1, \ldots, X_5$ having mean zero and unit variance and $X_6, \ldots, X_{10}$ having a mean of 10 and variance of 5. Eleven different simulation functions were then used to generate 11 different synthetic outcomes. Table I shows the name and corresponding simulation function used to generate each simulated data set. Here, AND and OR are used to denote the unique characteristics of these simulated data sets derived from using the indicator function, $I(\cdot)$ (1 if the argument is true; 0 otherwise). Similarly, SUM and SQ are based on the summing and summing of the squares of the predictor variables, respectively. The number in each data simulation name corresponds to the number of predictor variables included in the simulation functions. Note that less than the ten total predictor variables are used in each data simulation although all ten predictor variables are used in the construction of the random forests. To simulate the data, 100 random test points were generated from each distribution and then 100 random training sets of varying size (n = 200, 1000, 5000) were generated from each distribution. Eleven different distributions, each with three different sample sizes, resulted in 33 total types of simulated data sets.

2.2 Random forests

All possible combinations of the proposed variations on random forests were implemented on the simulated data (both CI trees and CART, as well as bootstrap and subsampling). For each variation, m_{try} was set to three different default

Table I. Ten functions used to simulate data.

Name	Simulation function
SUM1	X_1
SUM3	$X_1 + X_3 + X_5$
SUM5	$X_1 + X_3 + X_5 + X_6 + X_7$
SQ1	X_1^2
SQ3	$X_1^2 + X_3^2 + X_5^2$
SQ5	$X_1^2 + X_3^2 + X_5^2 + X_6^2$
OR1	$I(X_1 > 0.4)$
OR3	$I(X_1 > 0.4) * I(X_3 > 0.6) * I(X_5 > 0.4)$
OR5	$I(X_1 > 0.4) * I(X_2 > 0.6) * I(X_3 > 0.4) * I(X_5 > 0.4) * I(X_6 > 6)$
AND3	$\frac{1}{3}[I(X_1 > 0.4) + I(X_2 > 0.6) + I(X_3 > 0.4)]$
AND5	$\frac{1}{5}[I(X_1 > 0.4) + I(X_3 > 0.6) + I(X_5 > 0.4) + I(X_6 > 6)]$

$X_1, \ldots, X_5$ were sampled from the normal distribution with mean 0 and variance 1. $X_6, \ldots, X_{10}$ were sampled from the normal distribution with mean 10 and variance 5.

Table II. Three sample sizes (n) used for the simulated data sets and their corresponding subsample size (s), subsample fraction (s/n) and total resamples (B).

n	$s = n^{0.7}$	s/n	$B = 5n$
200	41	0.20	1,000
1,000	126	0.13	5,000
5,000	388	0.08	25,000

values: (i) max{floor($\frac{1}{3}p$), 1}, (ii) max{floor($\frac{2}{3}p$), 1} and (iii) max{p, 1} (where p is the total number of variables). For the current case of the simulations where $p = 10$, this resulted in m_{try} sizes of 3, 6 and 10. Because $p = m_{\text{try}} = 10$ is the trivial case for bagged trees, we instead opted for an m_{try} of 9. For subsampling random forest implementations, a subsample size of $n^{0.7}$ was used and all forests used $B = 5n$ total trees, as recommended by Wager & Athey (2017). Table II shows the specific numbers for subsample and resample size for each corresponding total sample size.

2.3 Simulation experiments

Figure 1 contains a diagram depicting an overview of the simulation experiments. Specifically, the four combinations of CI trees or CART with bootstrap or subsampling were implemented on all simulated data sets (Table I), each with sample sizes of 200, 1000 and 5000. $\hat{V}_{\text{IJ}}^{B}$ with the Monte Carlo correction as suggested by Wager et al. (2014) was calculated for all 100 test points, each using all 100 training samples. The empirical prediction variance for each test point was calculated as the variance of all 100 predictions of each test point on the training samples. The absolute bias in $\hat{V}_{\text{IJ}}^{B}$ was calculated as the absolute difference of each variance estimate and the empirical prediction variance. The average absolute bias for each test point was calculated by averaging the absolute bias across all 100 data sets. The average absolute bias was normalized by the empirical variance and termed as the "absolute predictive bias" in order to help in the interpretation and compare biases across different distributions and prediction ranges. Averaging the absolute predictive bias across all 100 test points resulted in the mean absolute predictive bias (MAPB) for each combination of tree type, resampling type, distribution and sample size:

$$MAPB = \frac{1}{100}\sum_{k=1}^{100} \frac{\frac{1}{100}\sum_{r=1}^{100} \left| \hat{V}_{\mathrm{IJ}}\left(x^{(k)}; Z^{(r)}\right) - \mathrm{Var}_r\left[RF_s\left(x^{(k)}; Z^{(r)}\right)\right] \right|}{\mathrm{Var}_r\left[RF_s\left(x^{(k)}; Z^{(r)}\right)\right]}, \tag{6}$$

where k represents the index of each test point $x^{(k)}$, r is the index of each training sample $Z^{(r)}$ and $RF_s\left(x^{(k)}; Z^{(r)}\right)$ is the ensemble prediction of the kth test point using the rth training set.

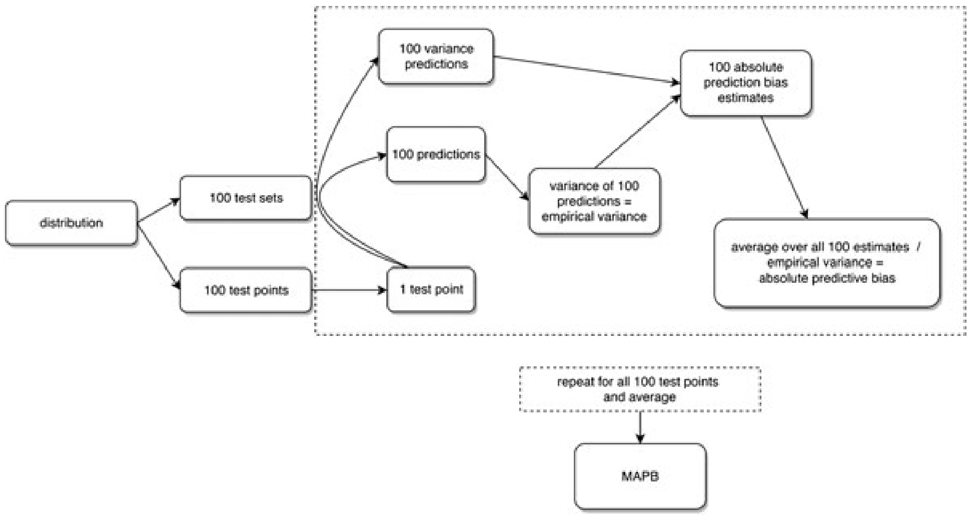
distribution
100 test sets
100 test points
100 variance predictions
100 predictions
1 test point
variance of 100 predictions = empirical variance
100 absolute prediction bias estimates
average over all 100 estimates / empirical variance = absolute predictive bias
repeat for all 100 test points and average
MAPB

Figure 1. A diagram depicting the simulation experiments. One hundred test points and 100 test tests were generated for each distribution, and these were used to calculate the mean absolute predictive bias (MAPB).

Table III. Median of the empirical variances (Var) for each distribution and sample size.

Distribution	Var ($n = 200$)	Var ($n = 1000$)	Var ($n = 5000$)
SUM1	0.0055	0.0007	0.0001
SUM3	0.0531	0.0192	0.0061
SUM5	0.8661	0.2588	0.0946
SQ1	0.0501	0.0088	0.0018
SQ3	0.3512	0.1236	0.0404
SQ5	78.8994	17.1922	6.1274
OR1	0.0018	0.0004	0.0000
OR3	0.0036	0.0007	0.0001
OR5	0.0048	0.0009	0.0001
AND3	0.0008	0.0001	0.0000
AND5	0.0009	0.0002	0.0000

(wileyonlinelibrary.com) DOI: 10.1002/sta4.162

2.4 *Statistical computing*

All statistical computing was carried out in `R`, version 3.1.2 (R Core Team, 2014), using the `randomForest` (Liaw & Wiener, 2002) and `Party` packages (Hothorn et al., 2006) wrapped into the custom package, `RFinfer` (Brokamp, 2016). A software vignette, available online with the authors' software package, describes installation and usage examples for `RFinfer`.

Table IV. The mean absolute predictive bias for each simulation, each the combination of a distribution, m_{try}, sample size, tree type and resampling method.

		CART						CI					
		Bootstrap			Subsample			Bootstrap			Subsample		
Distribution	m_{try}	200	1000	5000	200	1000	5000	200	1000	5000	200	1000	5000
	3	0.91	1.05	1.21	0.47	0.37	0.32	0.71	0.79		0.40	0.31	
	6	1.14	1.56	1.99	0.50	0.46	0.45	0.89	1.17		0.41	0.32	
SUM1	9	1.04	1.22	1.23	0.48	0.46	0.43	0.81	0.89		0.39	0.34	
	3	0.59	0.50	0.43	0.39	0.28	0.20	0.51	0.42		0.36	0.26	
	6	0.52	0.47	0.43	0.36	0.27	0.21	0.48	0.44		0.32	0.25	
SUM3	9	0.57	0.56	0.53	0.35	0.27	0.22	0.54	0.53		0.31	0.25	
	3	0.62	0.52	0.43	0.40	0.28	0.21	0.53	0.46		0.39	0.27	
	6	0.55	0.44	0.37	0.38	0.26	0.20	0.49	0.41		0.36	0.25	
SUM5	9	0.56	0.46	0.42	0.37	0.27	0.21	0.51	0.44		0.35	0.25	
	3	0.98	1.16	1.30	0.55	0.45	0.38	0.78	0.94		0.45	0.31	
	6	1.24	1.60	2.19	0.63	0.57	0.52	0.88	1.05		0.46	0.34	
SQ1	9	1.34	1.49	2.17	0.66	0.62	0.54	0.89	1.05		0.49	0.34	
	3	0.67	0.60	0.50	0.43	0.30	0.25	0.56	0.50		0.37	0.26	
	6	0.63	0.52	0.46	0.42	0.29	0.24	0.54	0.48		0.37	0.27	
SQ3	9	0.65	0.56	0.50	0.41	0.30	0.23	0.59	0.54		0.37	0.28	
	3	0.95	1.09	1.03	0.51	0.42	0.36	0.76	0.88		0.42	0.29	
	6	1.14	0.91	0.64	0.60	0.52	0.39	0.97	0.87		0.42	0.35	
SQ5	9	0.82	0.60	0.54	0.56	0.41	0.29	0.71	0.54		0.42	0.33	
	3	1.48	2.60	4.79	0.50	0.51	0.63	1.21	2.04		0.42	0.40	
	6	3.16	8.68	20.57	0.75	1.03	1.65	2.36	6.89		0.52	0.70	
OR1	9	7.12	22.94	45.56	1.31	1.71	2.09	5.70	19.24		0.91	1.44	
	3	1.30	2.64	4.82	0.44	0.46	0.61	0.95	1.90		0.37	0.35	
	6	1.87	5.21	16.36	0.50	0.72	1.10	1.31	3.56		0.37	0.48	
OR3	9	2.72	11.83	40.21	0.60	1.01	1.72	2.06	8.89		0.40	0.70	
	3	1.12	2.45	4.71	0.43	0.46	0.63	0.84	1.98		0.38	0.35	
	6	1.38	3.97	11.75	0.43	0.60	0.97	1.03	2.84		0.38	0.42	
OR5	9	1.72	7.58	24.36	0.46	0.79	1.31	1.33	4.53		0.37	0.52	
	3	1.03	1.85	3.55	0.41	0.38	0.40	0.86	1.48		0.36	0.32	
	6	1.31	3.71	9.69	0.41	0.52	0.78	1.03	2.60		0.35	0.34	
AND3	9	2.41	8.51	32.16	0.48	0.89	1.37	1.69	4.58		0.35	0.46	
	3	0.78	1.24	2.57	0.40	0.33	0.32	0.69	1.11		0.39	0.29	
	6	0.87	2.25	6.68	0.38	0.39	0.52	0.70	1.66		0.37	0.29	
AND5	9	1.38	6.27	19.41	0.40	0.54	0.96	0.98	2.82		0.37	0.33	

CART, classification and regression trees; CI, conditional inference.

3 Results

3.1 Data simulation

Ten predictor variables were generated from the normal distribution in order to simulate the data. $X_1, \ldots, X_5$ were drawn from the standard normal distribution, but $X_6, \ldots, X_{10}$ were drawn from a normal distribution with a mean of 10 and a variance of 5. The last five predictor variables were created to have a larger range in order to observe the effects of the random forest variations compared with data with only small ranging predictors. One hundred data sets were simulated for each of the 11 different simulation functions (Table I) and different sample sizes (n = 200, 1000, 5000). Furthermore, 100 test points used for prediction were generated for each simulation function.

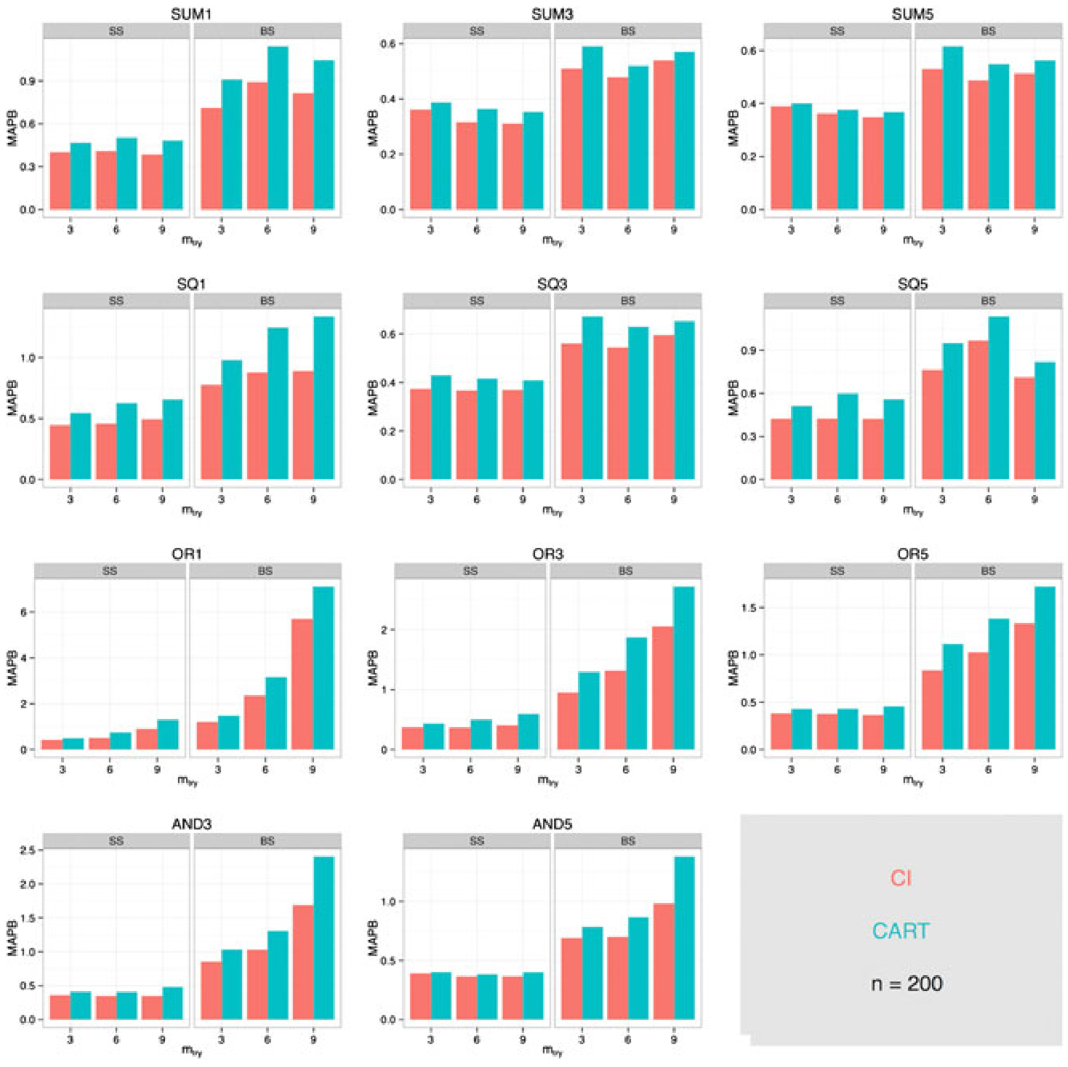

Figure 2. Mean absolute predictive bias (MAPB) for each combination of subsample (SS) or bootstrap (BS) resampling, conditional inference (CI, red bars on left) or traditional classification and regression trees (CART, green bars on right) and m_{try} for $n = 200$.

(wileyonlinelibrary.com) DOI: 10.1002/sta4.162

The ISI's Journal for the Rapid Dissemination of Statistics Research

3.2 Empirical variance

For each of the 100 test points, the variance of their predictions over all 100 test sets was calculated and termed as the empirical variance. Table III contains the median of these empirical variances for each distribution and sample size. As expected, the empirical variance increased within each type of distribution as more variables were used to generate the synthetic outcome and also decreased with increasing sample size. The *OR* and *AND* empirical variances were relatively small, all with a median less than 0.005. This is likely because the distributions utilized the indicator function, reducing the synthetic outcome to only a few possible levels and defeating the effect of using predictors $X_6, \ldots, X_{10}$, which had a larger range than $X_1, \ldots, X_5$. In contrast, the *SUM* and *SQ* distributions had a larger empirical variance, especially *SUM5* and *SQ5*, which utilized the predictors with a larger range and variation.

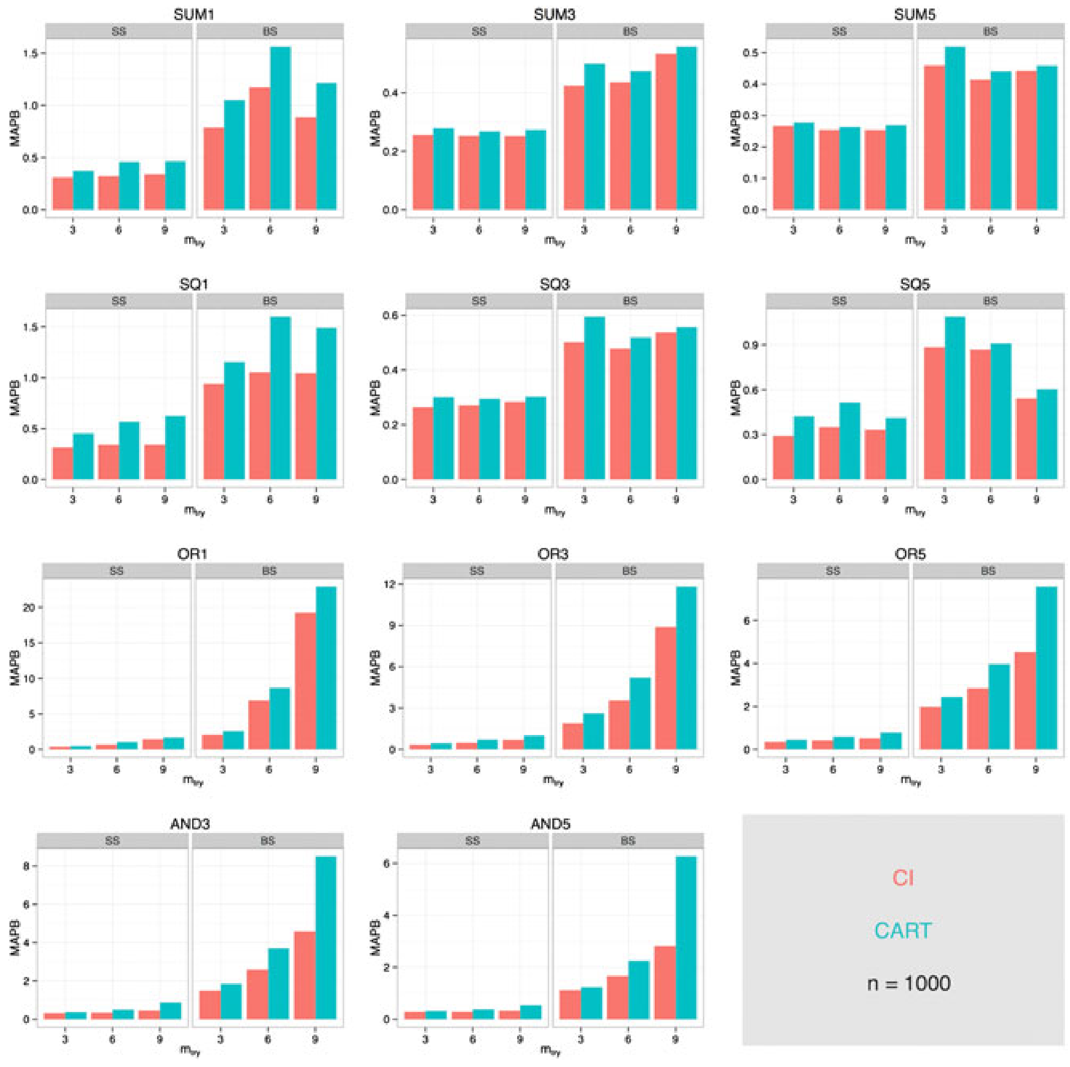

Figure 3. Mean absolute predictive bias (MAPB) for each combination of subsample (SS) or bootstrap (BS) resampling, conditional inference (CI, red bars on left) or traditional classification and regression trees (CART, green bars on right) and m_{try} for $n = 1000$.

3.3 Bias in variance predictions

The MAPB was calculated for each combination of resample type, tree type, m_{try}, sample size and distribution by calculating the absolute difference in the variance estimate and the empirical variance, normalizing this bias by the empirical variance, and averaging over all 100 data sets and all 100 prediction points (Equation (6)). See Figure 1 for a diagram depicting the simulation experiments. The full results are presented in Table IV, and Figures 2, 3 and 4 show the MAPB for sample sizes of $n = 200$, $n = 1000$ and $n = 5000$, respectively. CI trees were not created for $n = 5000$ because of computational limitations. Each of the simulation factors are explored in detail as follows.

3.3.1 Tree type.

The effect of tree type was consistent no matter the sample size, resampling method or m_{try} used; CI trees always had a lower MAPB than CART. The decrease in MAPB when using CI trees instead of CART did not seem to differ with respect to sample distribution.

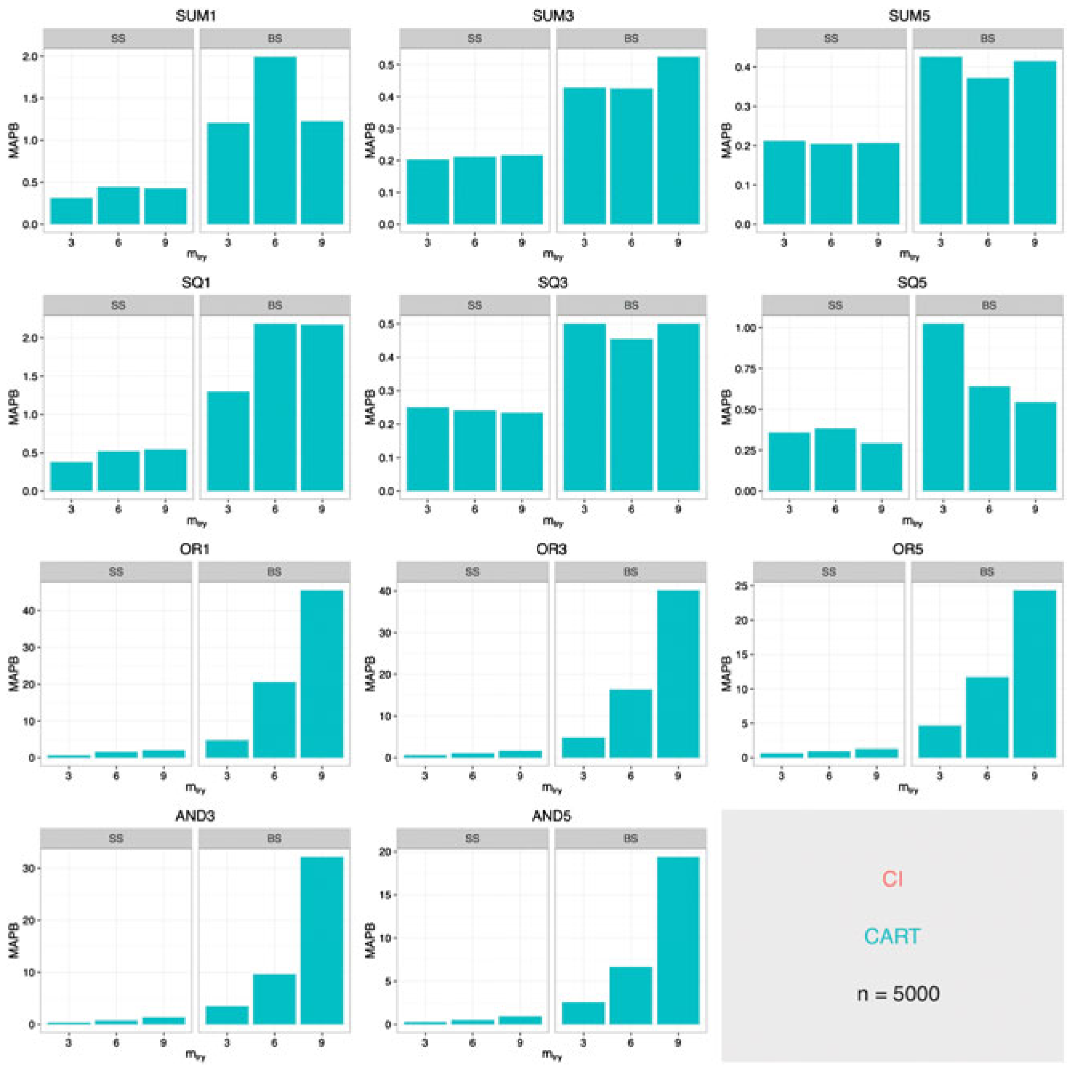

Figure 4. Mean absolute predictive bias (MAPB) for each combination of subsample (SS) or bootstrap (BS) resampling, traditional classification and regression trees (CART, green bars) and m_{try} for $n = 5000$.

3.3.2 Resampling method. The best improvement in MAPB resulted from utilizing subsampling rather than bootstrap sampling. In fact, the worst performing simulation type using subsampling always performed better than the best simulation type using bootstrap sampling for every distribution. This was again the case for all combinations of sample size, m_{try} and tree type. The difference was inflated when using a higher m_{try} in the bootstrapped *OR* and *AND* distributions.

3.3.3 Distribution. Overall, the *SUM* and *SQ* distributions performed well, with MAPB of mostly less than 1. The *AND* and *OR* distributions, however, performed much worse, especially with increasing sample size and using bootstrap resampling and high m_{try} values.

3.3.4 m_{try}. Increasing m_{try} caused a large increase in MAPB for the *OR* and *AND* distributions but caused a smaller effect with varied directions on the *SUM* and *SQ* distributions. Using subsampling with the *OR* and *AND* data sets generally exhibited a small increase in MAPB with increasing m_{try}, whereas using bootstrap resampling with the *OR* and *AND* data sets exhibited a very large increase in MAPB with increasing m_{try}. Within the *SUM* and *SQ* distributions, m_{try} had a much larger impact when bootstrap was used as the resampling method rather than subsampling. Here, m_{try} had a varied effect when using bootstrap resampling that depended on the number of variables used in each distribution and the total sample size.

3.3.5 Sample size. For the *SUM* and *SQ* distributions, increasing sample size decreased the MAPB for all combinations of m_{try}, tree type and resample type. However, the *OR* and *AND* distributions showed the opposite effect of increasing sample size, with a higher MAPB. This effect was especially large with the bootstrap resampling method; for example, the MAPB of the random forests with an m_{try} of nine trained on the $OR1$ distribution increased by an average of threefold when using $n = 1000$ instead of $n = 200$.

4 Discussion

Here, we have shown that using the IJ to estimate the variance of random forest predictions is much more accurate when using CI trees instead of traditional CART and when using subsampling instead of bootstrap sampling. These simulation experiments show that the proofs in Wager & Athey (2017) hold when using CI trees instead of traditional CART under various simulated data sets of different distributions and sizes.

Because there is no C implementation of the CI random forest method that indicates the number of times that each sample is included in each resample, the simulations using CI forests and a sample size of $n = 5000$ were computationally infeasible. In order to carry out our simulations using $\hat{V}^B_{\text{IJ}}$, we had to use a pure R implementation of CI random forests. This is different for CART random forests, where a C implementation already exists in the `randomForest` package. However, it should be noted that the difference in computational times is due to the random forest creation step, not the implementation of $\hat{V}^B_{\text{IJ}}$. This should not be an issue in the future when a C implementation of CI random forests is created.

The factor with the largest impact on MAPB was by far the resample method. Implementing sub-bagging resulted in a more accurate estimation of the prediction variance, and this eclipsed the change in MAPB caused by any other variations in m_{try}, sample size or tree type. Although using CI trees was better than using CART, the performance increases were largest when using bootstrap resampling with the non-linear *OR* and *AND* functions. However, the magnitude of improvement in MAPB was not increased for distributions utilizing the predictors with a wider range. This result was not expected given the known bias of CART utilizing wider ranging predictors (Strobl et al., 2007).

The non-linear distributions, *OR* and *AND*, had an extremely small empirical variance compared with the *SUM* and *SQ* distributions. Furthermore, the empirical variance decreased more rapidly with increasing sample size compared with the *SUM* and *SQ* distributions too. Thus, the increase in MAPB is likely due more to the decreased empirical variance rather then an increase in the absolute error of $\hat{V}_{\mathrm{IJ}}^{B}$, and this is likely why the MAPB increased with increasing sample size for the *OR* and *AND* distributions but decreased with increasing sample size for the *SUM* and *SQ* distributions.

The key to the random forest model is decorrelation of the individual trees using m_{try} and resampling. Bootstrap resampling does decorrelate trees, with each resample showing an average number of distinct observations of $0.632n$ (Hastie et al., 2005). However, using subsampling with a subsample size of $n^{0.7}$ results in a far lower number of distinct original observations per resample (see Table II for example). Thus, subsampling creates more decorrelation in individual trees than bootstrap sampling, and so m_{try} makes a large difference in the performance of bootstrapped random forests because there is room for additional decorrelation, but not in subsampled random forests. Similarly, the effects of m_{try} are greater in the *AND* and *OR* distributions when using bootstrap resampling and not subsampling because the variance of the resamples are already so small that bootstrap resampling does not sufficiently decorrelate the individual trees, unlike subsampling. Overall, this is why the worst performing subsampled simulation still outperformed the best bootstrapped simulation. Subsampling is likely more resistant to the correlation problems found in data with a lower variance and forests built with a higher m_{try} value.

Overall, our findings suggest that using CI trees instead of CART can be used in random forests to produce estimations of the prediction variance. However, it is most important to use subsampling instead of bootstrap sampling as this has the largest impact on the accuracy of $\hat{V}_{\mathrm{IJ}}^{B}$.

These simulations extend those performed previously by Wager & Athey (2017) by using different distributions, varying m_{try} values and including auxiliary noise variables in the training sets. However, in the future, it would be valuable to evaluate the performance of $\hat{V}_{\mathrm{IJ}}^{B}$ on correlated or multivariately distributed data, as well as on data with complex interactions because this is where random forest is most often used in real world settings.

References

Athey, S & Imbens, G (2016), 'Recursive partitioning for heterogeneous causal effects', *Proceedings of the National Academy of Sciences*, **113**(27), 7353–7360.

Breiman, L, Friedman, J, Stone, CJ & Olshen, RA (1984), *Classification and Regression Trees*, *CRC press*.

Brokamp, C (2016), *Rfinfer: v0.2*.

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

Fernández-Delgado, M, Cernadas, E, Barro, S & Amorim, D (2014), 'Do we need hundreds of classifiers to solve real world classification problems?', *Journal of Machine Learning Research*, **15**(1), 3133–3181.

Hastie, T, Tibshirani, R, Friedman, J & Franklin, J (2005), 'The elements of statistical learning: data mining, inference and prediction', *The Mathematical Intelligencer*, **27**(2), 83–85.

Hothorn, T, Hornik, K & Zeileis, A (2006), 'Unbiased recursive partitioning: a conditional inference framework', *Journal of Computational and Graphical Statistics*, **15**(3), 651–674.

Jaeckel, L (1972), *The infinitesimal jackknife, memorandum*, Technical Report, MM 72-1215-11, Bell Lab. Murray Hill, NJ.

Liaw, A & Wiener, M (2002), 'Classification and regression by randomForest', *R News*, **2**(3), 18–22.

(wileyonlinelibrary.com) DOI: 10.1002/sta4.162

R Core Team (2014), *R: A Language and Environment for Statistical Computing*, *R Foundation for Statistical Computing*, Vienna, Austria.

Strasser, H & Weber, C (1999), On the asymptotic theory of permutation statistics.

Strobl, C, Boulesteix, AL, Zeileis, A & Hothorn, T (2007), 'Bias in random forest variable importance measures: illustrations, sources and a solution', *BMC Bioinformatics*, **8**(1), 25.

Wager, S & Athey, S (2017), 'Estimation and inference of heterogeneous treatment effects using random forests', *Journal of the American Statistical Association, (just-accepted)*.

Wager, S, Hastie, T & Efron, B (2014), 'Confidence intervals for random forests: the jackknife and the infinitesimal jackknife', *Journal of Machine Learning Research*, **15**(1), 1625–1651.