Statistics Surveys
Vol. 3 (2009) 147–163
ISSN: 1935-7516
DOI: 10.1214/07-SS033

Navigating Random Forests and related advances in algorithmic modeling[*][†]

David S. Siroky

The MacMillan Center, Yale University,
Box 208206, New Haven, CT. 06511
e-mail: david.siroky@yale.edu

Abstract: This article addresses current methodological research on non-parametric Random Forests. It provides a brief intellectual history of Random Forests that covers CART, boosting and bagging methods. It then introduces the primary methods by which researchers can visualize results, the relationships between covariates and responses, and the out-of-bag test set error. In addition, the article considers current research on universal consistency and importance tests in Random Forests. Finally, several uses for Random Forests are discussed, and available software is identified.

AMS 2000 subject classifications: 62-02, 62-04, 62G08, 62G09, 62H30, 93E25, 62M99, 62N99.
Keywords and phrases: CART, bagging, boosting, Random Forests, algorithmic methods, non-parametrics, ensemble and committee methods.

Received December 2007.

This article addresses current methodological research on non-parametric Random Forests [14]. Random Forests are ensembles of trees grown from bootstrapped training data. For classification, the trees are combined using majority voting with one vote per tree over all the trees in the forest. For regression, forests are created by averaging over trees. Scholars tend to agree that non-parametric ensemble methods, or ‘committee methods’, such as Random Forests can offer significant improvements over any single classifier or regression tree [28, 29][38, p. 251].

In constructing the ensemble, Random Forests use two types of randomness. First, in growing any given tree, a random sample of predictors is selected at each node in choosing the best split. A further layer of randomness is added by using a random sample of observations for growing each tree in the first place. In theory, using a random sample of observations and selecting random predictors at each node should reduce dependence between covariates and thus between the resulting trees [14, p. 10-11][5].

[*] I wish to thank, without implicating, David Banks, Lenka Bustikova, Simon Lunagomez, Wendy Martinez, Sayan Mukherjee, Natesh Pillai, Jeffrey Solka, Edward Wegman and two anonymous reviewers for their comments and support. I am grateful to conference participants at the Joint Statistical Meetings in Washington DC, the Society for Political Methodology at Yale, and the American Political Science Association in Boston. All remaining errors are mine.

[†] This paper was accepted by Xiaotong Shen, Associate Editor for the IMS.

Results from the use of Random Forests have been impressive. Various studies have shown that Random Forests reduce classification and regression error on a wide array of data structures and problems from renal cell carcinoma classification and financial forecasting to genetic and bio-medical analysis [70, 80, 47]. Relative to some comparably accurate methods, Random Forests also possess a reasonably high degree of interpretability to help scientists understand the relative 'importance' of covariates, the effect of adding more variables on out-of-bag test set error, as well as proximities between observations and marginal effects. These features make the method not only accurate, but also useful for scientists, researchers and practitioners from diverse discipline areas.

The range of applications for which CART, bagging, boosting, and Random Forests are appropriate spans the natural and social sciences. The toy examples offered in this text come from political science data sets about voting and ethnic conflict. In my own research, I have applied these algorithms to predict and explain violent conflict, including outcomes such as the onset of civil war, the duration of civil war, the termination of ethnic conflict, and the incidence of armed secessionist rebellion. Applied statisticians and domain scientists in all quantitative fields should find these methods of interest as substitutes or complements for prediction, inference and description in standard regression, classification, longitudinal and censored survival settings.

The article proceeds as follows. The first section presents a brief intellectual history of Random Forests, which covers CART, boosting and bagging [30, 9, 40, 11, 32, 27]. The second section surveys some of the ways to visualize Random Forests and extract information that can help build better models. The penultimate section surveys some variations of the basic random forest algorithm which, in addition to regression and classification problems, can deal with censored survival and clustered data, the latter by bootstrapping at the subject (rather than sample) level. The article also describes some available software: front-ends for Breiman's Fortran code in the `R` language and in `MATLAB` ®, as well as `C++` and `Java`-based visualization tools. The final section summarizes the material and suggests some directions in which research on Random Forests appears headed.

1. A short intellectual history

1.1. CART

A reasonable place to begin understanding this class of models is with classification and regression trees [9]. The basic idea behind CART is straightforward. We use a set of observed predictors to partition the data recursively until the classes or values of the response variable in each sub-partition become fairly 'homogenous'. The contribution toward this homogeneity (or 'impurity') is one measure of 'variable importance,' a concept in classification settings that is typically measured as the total heterogeneity reduction produced by a given covariate on the response variable when the sample space is recursively partitioned. Homogeneity or impurity is typically defined as $i(\tau) = \theta[p(y = 1|\tau)]$, where the impurity of node τ is a non-negative function of $p(y = 1|\tau)$ in a $[0, 1]$ classification problem. θ is typically taken as the Gini $[\theta(p) = \mathtt{min}(p, 1 - p)]$, Entropy $[\theta(p) = [-p\mathtt{log}(p)] - [(1 - p)\mathtt{log}(1 - p)]$ or the Bayes error $[\theta(p) = p(1 - p)]$. More on how variable importance is measured is discussed below in the context of Random Forests following the discussion of CART, Boosting and Bagging.

The CART algorithm (for regression) has three steps.

1. At each non-terminal node, select a split to minimize the sum of squared errors: $SS_{error} = \sum_{i=1}^{N}(y_i - \hat{f}_c(x_i))^2$, where $\hat{f}_c$ is the predicted value for the relevant tree.
2. Determine when nodes are terminal (i.e., when to stop growing the tree or how much to prune) using cross-validation.
3. Estimate the outcome class or response value at each terminal node.

The first and third steps are relatively straightforward, but the second can present problems. Growing a large tree, pruning it, and tuning it using cross-validation to locate a tree with high predictive accuracy and interpretability is as much art as it is science [38, p. 270]. In short, cross validation suggests how much to prune the saturated tree by building 'ancillary trees' and calculating error rates for the saturated tree and the subtrees. There are many ways to conduct the cross-validation, including V-fold, leave-one-out or repeated random subsampling [59]. If we use V-fold cross-validation, where $V = 10$, we would grow 10 ancillary trees on a partition of the training data using 90% of the data and leaving 10% out in order to estimate the error rates. All of the error rates from the trees on which this procedure is done are then combined and used to determine how much pruning to do on the saturated tree. One popular implementation of this procedure is the `ipred` library in the `R` language [44].

Random Forests, which will be discussed shortly in more detail, obviate the need for cross-validation. Instead, they produce an unbiased estimate of the test set error internally by constructing many bootstrap samples from the original data and leaving about one-third of the cases out of the bootstrap sample and the construction of the n^{th} tree in the forest. This is one of several advantage that Random Forests possess over its predecessor, CART.

CART has other problems that scholars have duly discussed elsewhere; these include: '(1) discontinuous boundaries; (2) poor approximations of linear functions or functions which are additive in a small number of variables (cf. [38, p. 274]); (3) typically uncompetitive in low dimensions; (4) difficult to determine when a complex CART model is close to a simple model'; and (5) 'instability of trees' [2, p. 3][38, p. 272-4].

Despite these problems, CART is an attractive research tool: it not only clusters observations into groups with similar values on the response variable, but it also shows exactly how these clusters were constructed using a tree on which the branches are splits on the values of the explanatory variables. Categorical, ordered, unordered, interval–all variable types are handled seamlessly and allowed to relate in highly non-linear ways with the response [9, 60]. In some implementations of CART, prior probabilities can be assigned to the response class. In the [0,1] classification case, this would amount to specifying a prior probability of success or failure, based on one's belief about the marginal distribution of the response from previous studies. This provides a useful means of adding a cost according to whether a false positive or a false negative is more costly [9, section 4.4][5, p. 274-5].

1.2. Boosting, bagging and Random Forest

Instead of recursively partitioning smaller and smaller portions of the data set like CART, boosting considers the full data set at each potential partitioning node [67][38, p. 299ff]. The name comes from the ability 'to boost' a 'weak learning algorithm' into a stronger one using 'committee methods' [38, p. 299]. Assume two classes: -1 and 1: $Y \in \{-1, 1\}$, let X be a vector of explanatory variables and let $G(X)$ be a classifier, such that $G_m(x), m = 1, 2, \ldots, M$ is a sequence of classifiers. The predictions are aggregated using a weighted voting system: $G(x) = \texttt{sign}[\sum_{m=1}^{M} \alpha_m G_m(x)]$ [66, 68][38, p. 300]. The α_i's weigh the contribution of each $G_m(x)$, giving more accurate classifiers more weight, and vice versa. In step **2c**., α_m is the weight assigned to $G_m(x)$ in order to yield the final classifier $G(x)$. Misclassified observations are scaled by a factor $exp(\alpha_m)$, which increases their influence in the next sequence, $G_{m+1}(x)$ [38, p. 301].

Adaboost, or 'adaptive boosting' [38, p. 301], is a popular version of boosting that has performed well. Breiman speculates that Adaboost is actually a random forest [14, p. 20-21]. For a two-class classification problem with exponential loss, such as the one described above, the Adaboost.M1 algorithm is given as follows [32][38, 301]:

```
1. Initialize the observation weights, $w_i = 1/N, i = 1, 2, \ldots, N$.
2. For $m = 1$ to $M$:
      a. Fit a classifier $G_m(x)$ on the training data using weights $w_i$.
      b. Compute $err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x))}{\sum_{i=1}^{N} w_i}$.
      c. Compute $\alpha_m = log(1 - err_m/err_m)$.
      d. Set $w_i \leftarrow w_i \cdot exp[\alpha_m \cdot I(y_i \neq G_m(x_i))], i = 1, 2, \ldots, N$.
3. Output $G(x) = \texttt{sign}[\sum_{m=1}^{M} \alpha_m G_m(x)]$.
```

Boosting, however, is generally less attractive to statisticians than bagging or Random Forests, in part, because it lacks consistency and there is nothing implying convergence [54] in [5, p. 288].

Bagging, or 'bootstrap aggregation', is an ensemble method that uses a bootstrap sample of the hold out data [34] to train predictors and then combines results from several fitting attempts, assigning a predicted class or value for each observation [30, 11, 58][38, p. 246-9; 225-236]. The bootstrap sample is generated from a training set, Ω_{train}, of size n by producing m additional training sets with size $n^* : n^* \leq n$ by uniformly sampling from Ω_{train} with replacement. Bagging is one of the earliest methods to combine 'random trees' and, as such, provides a key step in the development of Random Forests [11].

Let Ω be an original data set, divided into a training Ω_{train} and a testing data set, Ω_{test}, and let B be a series of bootstrap samples from Ω_{train}. The basic bagging algorithm is given as follows [38, p. 246-7]:

1. From an original data set, Ω:
    - a. Take B bootstrap samples from the training data, Ω_{train}
    - b. Aggregate the collection of bootstrap samples: $\sum_i^B (B_i)$
2. Train predictors using these bootstrapped and aggregated (bagged) data by growing many trees without pruning and then counting the number of times (over trees) that each case is classified in each category.
3. Combine predictors using majority voting (for classification) or averaging (for regression) over the set of trees and assign cases accordingly.

The result of averaging all the classifiers is a decrease in variance for the bagged estimates [38, p. 247]. The price is an increase in bias. Assigning cases to categories using majority voting over a set of bootstrapped classification trees (step 3) reduces the chances of over-fitting. Majority voting means each tree votes: that is, assigns a class to each sample. The results are then aggregated and the class which receives the majority of votes is chosen.

For a two-class classification problem, a four cell 'confusion matrix' is produced with false positives, false negatives, and accurate class predictions. This matrix can be useful in understanding where the model falls short. Combined with an exception analysis that identifies influential outliers and observations which are poorly predicted by the model, the researcher can begin building better models [56].

Like bagging, Random Forests also grow many trees using bootstrapped samples from the training data. An unpruned classification tree is grown for each bootstrap sample and new cases are dropped down the trees. Majority voting is used to determine the terminal node to which the case should be assigned. Randomness is added at each node by choosing a random sample of predictors to consider for the split. The result is a forest constructed from randomly selected cases and randomly selected predictors. According to Breiman [14, p. 6]:

Definition 'Random Forests is a classifier consisting of a collection of tree-structured classifiers: $\{h(\mathbf{x}, \Theta_k), k = 1, \ldots\}$, where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$.'

The algorithm is given as follows [38, 14, 24, 2]:

```
1. For $B_i, i = 1, \ldots, B$
        a. Draw a bootstrap sample $S$ of size $N$ from the training data.
        b. Grow an unpruned random-forest tree, $T_b$ using the bootstrapped
    data, until the minimum terminal node size, $n_{min}$, is obtained by recursively following the sub-algortihm.
            Randomly select $p$ variables from the total set of $P$ variables.
            Select the optimal variable/split-point among the $p$ variables
            Split node into two daughter nodes.
2. Output ensemble of trees $\{T_b\}_1^B$
3. Predict new observations, or out-of-bag observations [12, 21]
        For regression: $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$
        For classification: Let $\hat{C}_b(x)$ be the class prediction of the $b^{th}$ random
    forest tree $\rightarrow \hat{C}_{rf}^B(x) =$ majority vote $\{\hat{C}_b(x)\}_1^B$
```

In addition to precision, this algorithm has the advantage of preventing over-fitting by reducing dependence between trees, which makes majority voting an effective strategy. M_{try}, the number of variables to try at each node, is the main tuning parameter. Breiman [14] suggests using $M_{try} = \sqrt{p}$, where p is the number of predictors, for regression and using $M_{try} = \frac{p}{3}$ for classification, but cross-validation can be used to optimize the choice for $M_{try} = f(p)$ [51, 10]. As for the number of trees, one can grow as many trees as one wishes without over-fitting, but can also inspect the out-of-bag error rate as a function of the number of trees (Figure 3) in order to determine how many trees to grow [18].

Randomly selecting predictors can reduce competition between similarly important factors or factors that fit some observations well but others poorly, particularly when multi-collinearity and sub-group differences are pronounced [5, p. 282]. Furthermore, Random Forests contain a built-in cross-validation method to calculate test set error using out-of-bag samples. Variables in out-of-bag samples are randomly permuted and then their impact on the test set error is measured, providing one useful method of determining 'variable importance' [14, p. 23ff]. Breiman [12] [14, p. 11] has argued that the out-of-bag test set error is as good as a test set of the same size without the need to keep aside a test set.

Following Breiman [16] and Sandri et al. [65, p. 4], we can define four prominent measures of variable importance in Random Forests:

- **Measure 1**: Randomly permute the values of the i^{th} variable. Obtain new classifications over out of bag observations (those not used to grow the tree) and record as $\hat{e}_i$. Compare with $\hat{e}$. For the i^{th} variable, calculate `arg max` $\{0; \hat{e}_i - \hat{e}\}$[72, 65]. The difference between the accuracy of the prediction before and after permutation provides the importance of the

i^{th} variable for one tree, and the important for the forest is calculated by averaging over all trees [74, p. 6-7].

- **Measure 2**: This measure takes the amount by which the margin is lowered across all trees as a measure of importance. Take an observation $(y, \mathbf{x})$, the margin function $marg(y, \mathbf{x})$ is defined as a the degree to which the proportion of the most voted incorrect classifications is exceeded by the proportion of the correct classifications:

$$\texttt{arg max}\{0; avg_s[marg(y, \mathbf{x}) - marg_i(y, \mathbf{x})]\}. \tag{1}$$

- **Measure 3**: Building on the previous two measures, this measure takes the difference of the number of lowered and raised margins as follows:

$$\texttt{arg max}\{0; \#[marg(y, \mathbf{x}) < marg_i(y, \mathbf{x})] - \#[marg(y, \mathbf{x}) \geq marg_i(y, \mathbf{x})]\}. \tag{2}$$

- **Measure 4**: This measure takes as the sum of all reductions in the Random Forests due to the i^{th} variable divided by the total number of trees in the forest:

$$I_{x_i} = \frac{1}{k} \sum_z [d(i, z) I(i, z)] \tag{3}$$

where z is a node in each tree that relies on a heterogeneity index such as the Shannon entropy or Gini index, $d(i, z)$ is the decrease in that heterogeneity index induced by x_i at node z, and $I(i, z)$ is an indicator function equal to 1 if the i^{th} variable is selected for a split at node $z.x_i$ is chosen to split on if $d(i, z) > d(w, z) \forall$ randomly selected X_w at node z.

It is prudent to consider all of these measures when using Random Forests for variable selection, since results may vary and inferences may be sensitive to one's choice. Breiman himself recommends trying all four. In addition, one has the option of considering up to date variations on these measures implemented in many statistical software languages.

In summary, Random Forests offer dramatic improvements in predictive accuracy and stability, but they do not leave intuitive trees behind to interpret. Breiman and others have suggested and developed a number of clever ways to visualize Random Forests that make them attractive methods not only for prediction but also for data description, model assessment and model improvement.

2. Visualizing results

Random Forests improve upon CART in a number of ways, including stability and accuracy, but in exchange they forgo the single interpretable tree. A number of visualization tools have been suggested to fill this gap. Figure 1 and Figure 2 are examples.

Figure 1 illustrates how variable importance is measured and ranked. The predictive accuracy is defined on the out-of-bag data for each tree and after randomly permuting each variable in the out-of-bag sample by counting the

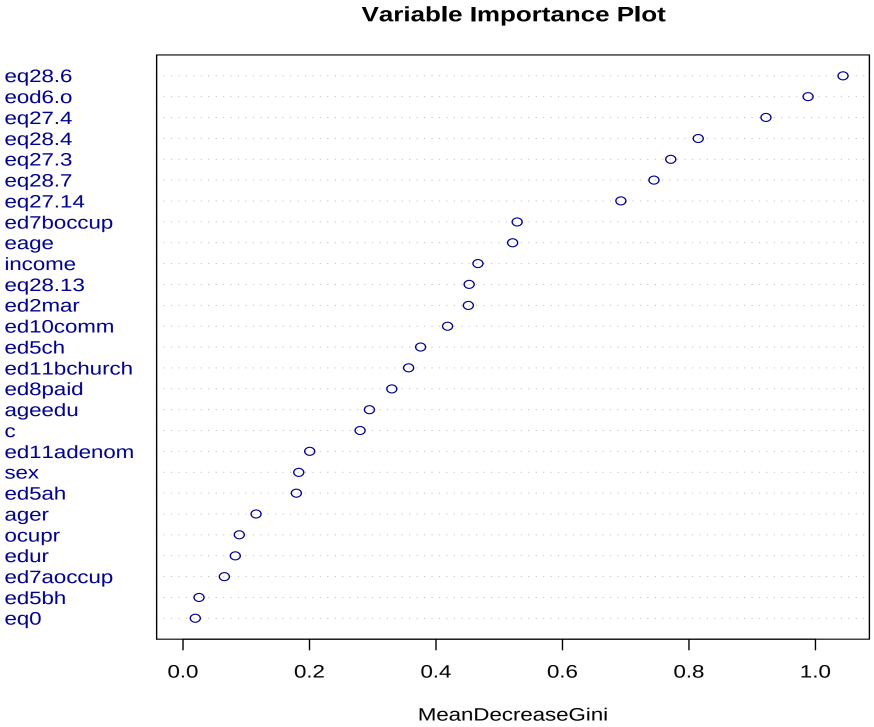

Fig 1. Variable Importance Plot. The data set used in this analysis is from a Euro-barometer survey. The response is a vote for a particular political party type. The variables label `eq28.x` all concern attitudes toward asylum seekers; all labels `eq27.x` concern attitudes toward immigrants, `eod6.o` concerns car ownership, `ed7boccup` is an occupation question, `eage` is year of birth, `income` is income deciles, `ed2mar` is marital status, `ed10comm` is a rural/urban residence question, `ed5ch` concerns the respondent's number of children and their ages, `ed11bchurch` is about church attendance, `ed8paid` concerns the source of respondent's income, `ageedu` is years of schooling, `c` is country random effect, `ed11adenom` is the respondent's denomination, `sex` is the respondent's gender, `ed5ah` is the number of adults living in the hoursehold, `ager` is age of respondent,`ocupr` is occupation of the respondent, `edur` is the education level of the respondent, `ed7aoccup` is the respondent's current occupation, `ed5bh` is the number of adults living in the household.

correct votes. The difference between the permuted and non-permuted counts is calculated, averaged over the forest and normalized using the standard error. In regression, the mean-square error is calculated for each tree in the out-of-sample forest with and without permutation. The following is more structured description of the process [38, p. 593]:

```
1. For the $b^{th}$ tree:
        a. Pass out of bag samples down the tree and record accuracy.
2. Randomly permute the values of the $j^{th}$ variable in the out of bag sample
        Recalculate and record accuracy.
```

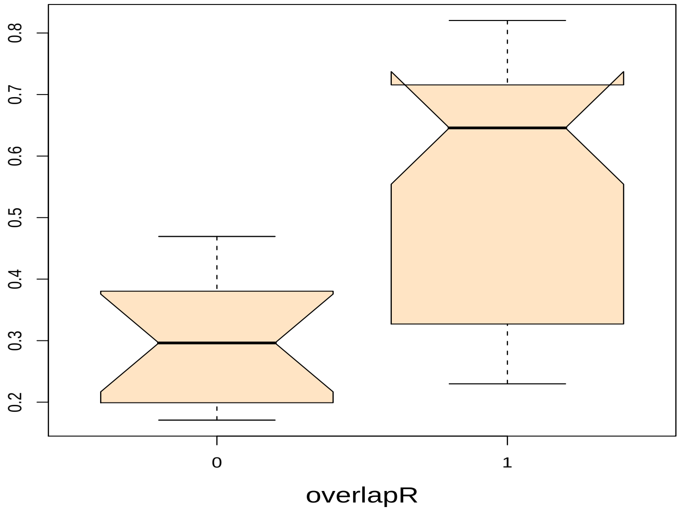

Fig 2. A Partial Plot of One Dichotomous Predictor Variable. The data set used in this analysis concerns violent irredentist conflict between post-secessionist states and the rump states [71]. Spatially, the data cover the entire globe and, temporally, includes all cases since the early 19th century. The variable in question is a simple indicator variable of whether the two states in question possess overlapping ethnic population pockets. The presence of such pockets is positively related to the response, interstate conflict.

3. Average the decrease in accuracy that resulted from randomly permuting over all the trees, $b_1, \ldots, b_B$

   Call this difference the importance of the j^{th} variable.
4. Repeat for $j_i, i = 1, \ldots, N$ and rank

Figure 1 uses a slightly different measure–the total decrease in node impurities that results from uses a given variable to split, averaged over all the trees [51, p. 7]. Since this is a classification example, node impurity is measured using the Gini index rather than the residual sum of squares [18, 51] [14, p. 10-11][5, p. 283].

Marginal or partial dependence plots (Figure 2) provide another approach to visualizing results [38, p. 331-44][18, p.12-3]. Other predictors are 'held constant', allowing one to examine the relationship between an individual predictor and the response. The partial dependence function can be estimated for regression as in [38, p. 333, 10.51] [34, 51]:

$$\tilde{f}_S(X_S) = \frac{1}{N}\sum_{i=1}^{N} f(X_S, x_{iC}), \tag{4}$$

where X_S is the predictor on which partial dependence is estimated, $\{x_{iC}\}$ are the values of X_C from the Xs in the training data set. The function and related

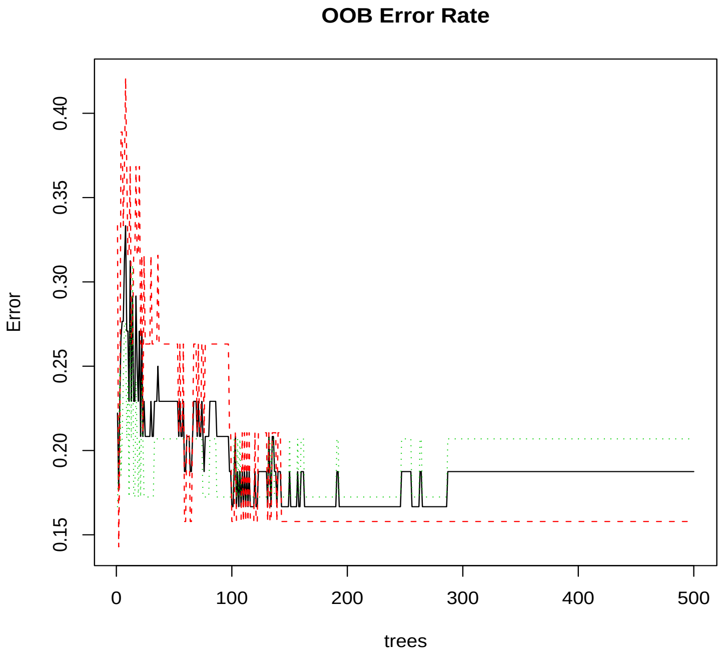

FIG 3. Out of Bag Test Set Error Rate.

TABLE 1
A ‘Confusion Matrix’ of a Data Set with 100 Observations

	No Event Predicted	Event Predicted	Error
No Event	41	9	18%
Event	6	44	12%

plot show the effect of X_S on f_X after adjusting for the effect of X_C on f_X, and not the effect of X_S on f_X ignoring the effect of X_C [38, p. 333-4]. The plot below, based on data about the occurrence of ethnic conflict, shows that when the predictor is in class 1 (versus 0) the response is much more likely to be in class 1 (versus 0) after adjusting for the other predictors.

In addition to the relationship between predictors and responses, one can examine a ‘confusion matrix’ of predictive performance for the 2-class case (see Table 1 for an example). For binary classification problems, this is a 2 X 2 table, with accurate predictions on the Northwest and Southeast diagonals, and inaccurate predictions (false negatives and positives) on the other diagonal (Table 1). The test set error rate can be visualized as a function of the number of trees grown (Figure 3). Figure 3 shows that the average test set error rate plateaus around 15% after a few hundred trees. The false positive and false negative rates are graphed separately (18% and 12% respectively), and the heavy line running between the two is the average test set error rate (15%).

3. Uses of Random Forests

Random Forests can be used for a number of purposes, such as describing the relationship between predictors and responses, forecasting, variable selection, missing data imputation and classification. Random Forests handle missing data and out-of-sample testing naturally; they can be used with mixed discrete and continuous predictors and responses, and also offer an elegant approach for survival analysis. Although variable importance measures can offer powerful insights of complex data structures, some recent research suggests a possible bias in Random Forests variable importance measures for situations in which predictors vary widely in terms of measurement or number of categories [72].

Random Forests can also be useful when combined with more conventional approaches. As Berk [5, p. 290] points out, if predictors thought to be important in one analysis do not emerge in the other, this may indicate the need for further scrutiny and can reveal new information about non-linearities and interactions. In observational studies, Berk, Li and Hickman [6] have suggested that ensemble methods can sometimes do a better (lower bias) job of modeling the selection process than propensity score matching, which usually relies upon logistic regression to determine the probability of treatment group membership [6] in [5, p. 290][23, 36, 61, 62, 63, 64, 39].

4. Universal consistency and predictor importance

The impressive performance of Random Forests has brought more attention to its properties and limitations. Breiman himself raised, but did not fully address, a number of questions about the consistency of random forest averaging rules [7, 17, 75]. For example, Breiman [14] wrote that 'Use of the Strong Law of Large Numbers shows that they [Random Forests] always converge so that over-fitting is not a problem…this result explains why Random Forests do not over-fit as more trees are added, but produce a limiting value of the out of bag error' [14, 78, 7].

To investigate the issue of 'universal consistency', Biau, Devroye and Lugosi [7] consider a binary classification problem and show that the 'purely random forest classifier' and the 'scale-invariant random forest classifiers' are consistent, but also that randomized, greedily grown tree classifiers are inconsistent and propose an alternative methodology to define a computationally feasible consistent greedily grown random forest classifier [7, p. 2028-2031] [74, Theorem 20.9] [7, p. 2018-2026]).

While Biau, Devroye and Lugosi [7] have examined the consistency of Random Forests, other research has begun to explore additional properties and limitations. For example, Strobl et al. [72, 73] demonstrated that one of the two more common variable importance measures–the Gini measure–is biased when predictors vary significantly in scale. By contrast, the same authors find that a second method of determining variable importance is unbiased under the same circumstances. Breiman et al. [9] first observed that similar measures may be biased in favor of selecting those variables with more values of the covariate, in part, because these variables provide more splits.

Strobl and Zeileis [73] investigated the Random Forests permutation importance test's ability to identify relevant predictor variables that may be correlated in a simulation study. The authors concluded that the test possesses some undesirable properties in its current form, although the issues could probably be addressed by applying a 'conditional permutation scheme'. Strobl and Zeileis [73] also question the rejection area for the null hypothesis being tested and argue that further research is necessary to determine its adequacy. Finally, the authors note that multiple testing issues will have to be taken into account when a large number of variables are under consideration. These issues, inter alia, will continue to constitute the subject of further research and debate on Random Forests, but it is already clear that they perform well in a wide range of scientific domains and have earned their place in the algorithmic canon.

5. Software

Random Forests, and other ensemble methods, are available in many software formats. CART, having been around for a few decades already, is of course widely available in both free-ware and commercial-ware. Random Forests are available in R ([51]) and MATLAB ([79]). R, a popular free-ware statistical environment, has a well functioning and frequently updated Random Forests library [50], and a version for time-to-event censored survival data [45], as well as some specialized packages, such as `varSelRF`, which implements a validated method for selecting small sets of predictors while preserving classification accuracy [26]. MATLAB also has an interface to the random forest algorithm, contributed by Ting Wang ([79]), in addition to many other statistical learning methods. Breiman and Cutler have written a java-based visualization tool called `RAFT`, which stands for *RA*ndom *F*orest *T*ool ([8]). Karpievitch, Hill, Millar, Smolka, Almeida and Hoffman [46] propose a modification of the Random Forest algorithm for clustered, repeated measure data. These data are commonplace in a wide range of application domains, including bio-statistical and social scientific applications. Karpievitch, Leclerc, Hill, and Almeida [46] recently introduced the `C++` free-ware 'RF++: Improved Random Forest for Clustered Data Classification'.

6. Conclusions

Random Forests have already gained considerable attention, despite being relatively new. In addition to impressive accuracy, some research suggests that Random Forests are more robust to noise, more stable and faster to train than Adaboost [13, 49, 22]. Random Forests—and other statistical learning methods– have often been ignored, in part, because they differ so dramatically from widely used statistical modeling tools. Hopefully, this article has uncovered some of the mystery behind these methods, making them more familiar to statisticians and applied researchers, and prompting some readers to explore further.

References

[1] Banks, D., L. House, P. Arabie, F.R. McMorris, and W. Gaul, eds. 2004. *Classification, Cluster Analysis, and Data Mining*, Springer-Verlag, Berlin. MR2112710

[2] Banks, D. 2007. *Lectures on Statistical Data Mining*, Duke University, Aug. 29–Nov. 28. http://www.stat.duke.edu/~banks/218-lectures.dir/

[3] Bauer, E. and Kohavi, R. 1999. 'An Empirical Comparison of Voting Classification Algorithms,' *Machine Learning*, 36, No. 1/2, 105–139.

[4] Buehlmann, P. and B. Yu. 2002. 'Analyzing Bagging' *The Annals of Statistics* 30: 927–61. MR1926165

[5] Berk, R. 2006. 'An Introduction to Ensemble Methods for Data Analysis.' *Sociological Methods and Research*, 34: 3, (February), 263–95. MR2247098

[6] Berk, R., A. Li and L. Hickman. 2005. 'Statistical Difficulties in Determining the Role of Race in Capital Cases', *Journal of Quantitative Criminology*, 21: 4, 365–390.

[7] Biau, G., L. Devroye, and G. Lugosi. 'Consistency of Random Forests and other averaging classifiers.' Preprint, October 10, 2007. MR2447310

[8] Breiman, L. and A. Cutler, RAF: http://www.math.usu.edu/~adele/forests/cc_graphics.htm

[9] Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone. 1984. *Classification and Regression Trees.* Monterey, CA: Wadsworth. MR0726392

[10] Breiman, L., and P. Spector. 1992. 'Submodel selection and evaluation in regression: The X-random case,' *International Statistical Review*, 60: 291–319.

[11] Breiman, L. 1996a. 'Bagging Predictors.' *Machine Learning* 26: 123–40.

[12] Breiman, L. 1996b. 'Out-of-Bag Estimation.' ftp://ftp.stat.berkeley.edu/pub/users/breiman/OOBestimation.ps.

[13] Breiman, L. 1999. 'Random Forests–Random Features.' UC Berkeley, Statistics Department, *Technical Report N. 567.*

[14] Breiman, L. 2001a. 'Random Forests.' *Machine Learning* 45: 5–32.

[15] Breiman, L. 2001b. 'Statistical Modeling: Two Cultures' (with discussion). *Statistical Science* 16: 199–231. MR1874152

[16] Breiman, L. 2001c. 'Wald Lecture I: Machine Learning' and 'Wald Lecture II: Looking Inside The Black Box' ftp://ftp.stat.berkeley.edu/pub/users/breiman/.

[17] Breiman, L. 2004a. 'Consistency For A Simple Model Of Random Forests,' Technical Report 670, Statistics Department University Of California at Berkeley, September 9, 2004.

[18] Breiman, L. and A. Cutler. 2004. 'Random Forests' http://statwww.berkeley.edu/users/breiman/RandomForests/cc_home.htm.

[19] Breitenbach, M., R. Nielsen and G. Grudic 'Probabilistic Random Forests: Predicting Data Point Specific Misclassification Probabilities,' Available at http://www.cs.colorado.edu/department/publications/

reports/docs/CU-CS-954-03.pdf. MATLAB code available at: http://markus-breitenbach.com/machine_learning_code.php.

[20] Buehlmann, P. and Bin Yu. 2002. 'Analyzing Bagging.' *The Annals of Statistics* 30: 927–61. MR1926165

[21] Bylander, T. 2002. 'Estimating Generalization Error on Two-Class Datasets Using Out-of-Bag Estimates,' *Machine Learning* 48, 1–3, p. 287–297.

[22] Chan, J.C-W. and D. Paelinckx. 2008. 'Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery,' *Remote Sensing of Environment* 112, 6, 16 June 2008, 2999–3011.

[23] Cochran, W.G., and D.B. Rubin. 1973. Controlling bias in observational studies: A review. Sankhya: *The Indian Journal of Statistics*, Series A 35(Part 4): 417–66.

[24] Cutler, A. and L. Breiman, RAFT: *RA*ndom *F*orest *T*ool, Available at: http://www.stat.berkeley.edu/users/breiman/RandomForests/.

[25] L. Devroye, L. Gyorfi, and G. Lugosi. 1996. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, New York. MR1383093

[26] Diaz-Uriarte, R. 2007. '`GeneSrF` and `varSelRF`: a web-based tool and R package for gene selection and classification using random forest, *BMC Bioinformatics*, 8: 328.

[27] Dietterich, T. 1998. 'An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization', *Machine Learning*, 1–22.

[28] Dietterich, T. 2002. 'Ensemble Learning,' In *The Handbook of Brain Theory and Neural Networks*, Second edition, (M.A. Arbib, Ed.), Cambridge, MA: The MIT Press, 405–408. MR2132702

[29] Dietterich, T. 2007. 'Ensemble Methods in Machine Learning,' Available at: eecs.oregonstate.edu/~tgd/publications/mcs-ensembles.ps.gz.

[30] Efron, B. 1979. 'Bootstrap methods: another look at the jackknife,' *The Annals of Statistics* 7: 1–26. MR0515681

[31] Efron, B. and G. Gong. 1983. 'A leisurely look at the bootstrap, the jackknife, and cross-validation,' *The American Statistician* 37: 36–48. MR0694281

[32] Freund, Y. and R. Schapire. 1996. 'Experiments with a new boosting algorithm', *Machine Learning: Proceedings of the 13th International Conference*, 148–156.

[33] Friedman, J.H., T. Hastie, and R. Tibsharini. 2000. 'Additive Logistic Regression: A Statistical View of Boosting' (with discussion). *Annals of Statistics* 28: 337–407. MR1790002

[34] Friedman, J.H., T. Hastie, and R. Tibsharini. 2001. 'Greedy Function Approximation: A Gradient Boosting Machine.' *Annals of Statistics* 29: 1189–1232. MR1873328

[35] Friedman, J.H., T. Hastie, and R. Tibsharini. 2002. 'Stochastic Gradient Boosting.' *Computational Statistics and Data Analysis* 38: 4, 367–78. MR1884869

[36] Frölich, M. 2004. 'Finite sample properties of propensity score matching and weighting estimators,' *Review of Econometrics and Statistics* 86: 77–90.

[37] Grandvalet, Y. 2004. 'Bagging Equalizes Influence.' *Machine Learning* 55: 251–70.

[38] Hastie, T., R. Tibshirani, and J. Friedman. 2001[2009]. *The Elements of Statistical Learning*. New York: Springer-Verlag. MR1851606

[39] Ho, D., K. Imai, G. King, and E. Stuart. 2007. 'Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference,' *Political Analysis*, 15: 199–236.

[40] Ho, T.K. 1995. 'Random Decision Forest'. *Proceedings of the 3rd International Conf. on Document Analysis and Recognition*, Montreal, Canada, August 14–18, 1995, 278–282.

[41] Hothorn, T. and B. Lausen. 2003. 'Double-bagging: Combining classifiers by bootstrap aggregation,' *Pattern Recognition*, 36: 6, 1303–1309.

[42] Hothorn, T., B. Lausen, A. Benner and Ma. Radespiel-Troeger. 2004. 'Bagging Survival Trees'. *Statistics in Medicine*, 23: 1, 77–91.

[43] Hothorn, T., P. Buhlmann, S. Dudoit, A. Molinaro and M.J. van der Laan. 2006. 'Survival Ensembles'. *Biostatistics*, 7: 3, 355–373.

[44] Hothorn, T. and A. Peters, 2009. ipred, http://cran.r-project.org/web/packages/ipred/index.html

[45] Ishwaran, H. and U. Kogalur. 2007. randomSurvivalForest (R software for random survival forest) Ensemble survival analysis based on a random forest of trees using random inputs. Version 3.0.1. MR2357716

[46] Karpievitch, Y.V., A.P. Leclerc, E.G. Hill, J.S. Almeida, 'RF++: Improved Random Forest for Clustered Data Classification,' http://www.ohloh.net/p/rfpp

[47] Kumar, Manish and M. Thenmozhi, 'Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest,' Indian Institute of Capital Markets 9th Capital Markets Conference Paper Available at SSRN: http://ssrn.com/abstract=876544.

[48] LeBlanc, M. and R. Tibshirani. 1996. 'Combining Estimates on Regression and Classification.' *Journal of the American Statistical Association* 91: 1641–50. MR1439105

[49] Leshem, G. 2005. 'Improvement of Adaboost Algorithm by using Random Forests as Weak Learner.' *Ph.D. Thesis, Hebrew University of Jerusalem*: shum.huji.ac.il/~gleshem/Guy_Leshem_Proposal.pdf

[50] Liaw, A. and M. Wiener. 'Classification and Regression by randomForest' *R News* (2002) Vol. 2/3 p. 18 (Discussion of the use of the random forest package for R).

[51] Liaw, A. and M. Weiner. 2007. randomForest (R software for random forest). Fortran original (L. Breiman and A. Cutler), R port (A. Liaw and M. Wiener) Version 4.5-19 and 4.5-25. http://cran.r-project.org/web/packages/randomForest/index.html

[52] Lin, Y. and Y. Jeon. 2006. 'Random Forests and adaptive nearest neighbors,' *Journal of the American Statistical Association*, 101 (474): 578–590. MR2256176

[53] Loh, W.-Y. 2002. 'Regression Trees With Unbiased Variable Selection and Interaction Detection.' *Statistica Sinica* 12: 361–86. MR1902715

[54] Mannor, S., R. Meir and T. Zhang. 2002. 'The Consistency of Greedy Algorithms for Classification,' *COLT*, 319–333. MR2040422

[55] Meinshausen, N. 2006. 'Quantile regression forests,' *Journal of Machine Learning Research*, 7: 983–999. MR2274394

[56] Nyuyen, T.T. 2008. 'Outlier and Exception Analysis in Rough Sets and Granular Computing,' in *Handbook of Granular Computing* (Eds. W Pedrycz, A. Skowron, V. Kreinovich), Wiley 2008.

[57] Opitz, D. and R. Maclin. 1999. 'Popular Ensemble Methods: An Empirical Study', *Journal of Artificial Intelligence Research*, 11, 169–198, `citeseer.ist.psu.edu/opitz99popular.html`.

[58] Peters, A. and T. Hothorn. 2007. `ipred`: Improved predictive models by indirect classification and bagging for classification, regression and survival problems as well as resampling based estimators of prediction error. (R software for random forest prediction). Version: 0.8-5

[59] , Picard, R. and D. Cook. 1984. 'Cross-Validation of Regression Models,' *Journal of the American Statistical Association* 79 (387): 575–583. MR0763576

[60] Quinlan, R. 1993. *C4.5: Programs for Machine Learning* (Morgan Kaufmann)

[61] Rosenbaum, P.R. 1984. 'The consequences of adjusting for a concomitant variable that has been affected by the treatment,' *Journal of the Royal Statistical Society*, Series A 147: 656–66.

[62] Rosenbaum, P.R. 1989. 'Optimal matching for observational studies,' *Journal of the American Statistical Association* 84: 1024–1032.

[63] Rosenbaum, P.R. 2002. *Observational studies*. 2nd ed. New York: Springer. MR1899138

[64] Rosenbaum, P.R., and D.B. Rubin. 1983. 'The central role of the propensity score in observational studies for causal effects,' *Biometrika* 70: 41–55. MR0742974

[65] Sandri, M. and P. Zuccolotto. 2009. 'Variable selection using Random Forests,' *Typescript*, 8 pages.

[66] Schapire, R.E. 1990. 'The strength of weak learnability,' *Machine Learning*, 5: 197–227.

[67] Schapire, R. E. 1999. 'A Brief Introduction to Boosting.' In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*.

[68] Schapire, R.E., Y. Freund, P. Bartlett, and W.S. Lee. 1998. 'Boosting the margin: A new explanation for the effectiveness of voting methods,' *The Annals of Statistics*, 26: 1651–1686. MR1673273

[69] Shannon, W., and D. Banks. 1997. 'An MLE Strategy for Combining CART Models,' *Computing Science and Statistics*, 29: 540–544.

[70] Shi, T., Seligson, D. Belldegrun, A.S. Palotie, A. and Horvath, S. 2005. 'Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma,' *Modern Pathology* 18: 4, 547–57.

[71] Siroky, D.S. 2009. *Secession and Survival*, Ph.D. Dissertation, Duke University.

[72] Strobl, C., A. Boulesteix, A. Zeileis and T. Hothorn. 2007. Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution. *BMC Bioinformatics*, 8, 25. http://www.biomedcentral.com/1471-2105/8/25/abstract.

[73] Strobl, C. and A. Zeileis. 2008. 'Danger: High Power! – Exploring the Statistical Properties of a Test for Random Forest Variable Importance,' Technical Report Number 017, Department of Statistics, University of Munich.

[74] Strobl, C., A-L Boulesteix, T. Augustin and A. Zeileis. 2008. 'Conditional variable importance for Random Forests,' *BMC Bioinformatics*, 9: 307.

[75] Stone, C. 1977. 'Consistent nonparametric regression,' *The Annals of Statistics*, 5: 595–645. MR0443204

[76] Su, X., M. Wang, and J. Fan. 2004. 'Maximum Likelihood Regression Trees.' *Journal of Computational and Graphical Statistics* 13: 586–98. MR2087716

[77] Therneau, T.M and B. Atkinson, '`rpart`: Recursive Partitioning' Recursive partitioning and regression trees Version 3.1-38 (CART for R).

[78] Traskin, M. 'Random Forests: classification, variable selection and consistency,' STAT900 Slides, University of Pennsylvania, Nov. 26, 2007.

[79] Wang, T. MATLAB R13. Available at: http://lib.stat.cmu.edu/matlab/

[80] Ward, M., S. Pajevic, J. Dreyfuss, and J. Malley. 2006. 'Short-term prediction of mortality in patients with systemic lupus erythematosus: classification of outcomes using Random Forests,' *Arthritis and Rheumatism* 55: 74–80.