Machine Learning, 45, 5–32, 2001

Random Forests

LEO BREIMAN
Statistics Department, University of California, Berkeley, CA 94720

Editor: Robert E. Schapire

Abstract. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favorably to Adaboost (Y. Freund & R. Schapire, *Machine Learning*: *Proceedings of the Thirteenth International conference*, ∗ ∗ ∗, 148–156), but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. These ideas are also applicable to regression.

Keywords: classification, regression, ensemble

1. Random forests

1.1. Introduction

Significant improvements in classification accuracy have resulted from growing an ensemble of trees and letting them vote for the most popular class. In order to grow these ensembles, often random vectors are generated that govern the growth of each tree in the ensemble. An early example is bagging (Breiman, 1996), where to grow each tree a random selection (without replacement) is made from the examples in the training set.

Another example is random split selection (Dietterich, 1998) where at each node the split is selected at random from among the K best splits. Breiman (1999) generates new training sets by randomizing the outputs in the original training set. Another approach is to select the training set from a random set of weights on the examples in the training set. Ho (1998) has written a number of papers on "the random subspace" method which does a random selection of a subset of features to use to grow each tree.

In an important paper on written character recognition, Amit and Geman (1997) define a large number of geometric features and search over a random selection of these for the best split at each node. This latter paper has been influential in my thinking.

The common element in all of these procedures is that for the kth tree, a random vector Θ_k is generated, independent of the past random vectors $\Theta_{1,\ldots,}\Theta_{k-1}$ but with the same distribution; and a tree is grown using the training set and Θ_k, resulting in a classifier $h(\mathbf{x}, \Theta_k)$ where $\mathbf{x}$ is an input vector. For instance, in bagging the random vector Θ is

generated as the counts in N boxes resulting from N darts thrown at random at the boxes, where N is number of examples in the training set. In random split selection Θ consists of a number of independent random integers between 1 and K. The nature and dimensionality of Θ depends on its use in tree construction.

After a large number of trees is generated, they vote for the most popular class. We call these procedures **random forests**.

Definition 1.1. A random forest 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 $\mathbf{x}$.

1.2. Outline of paper

Section 2 gives some theoretical background for random forests. Use of the Strong Law of Large Numbers shows that they always converge so that overfitting is not a problem. We give a simplified and extended version of the Amit and Geman (1997) analysis to show that the accuracy of a random forest depends on the strength of the individual tree classifiers and a measure of the dependence between them (see Section 2 for definitions).

Section 3 introduces forests using the random selection of features at each node to determine the split. An important question is how many features to select at each node. For guidance, internal estimates of the generalization error, classifier strength and dependence are computed. These are called out-of-bag estimates and are reviewed in Section 4. Section 5 and 6 give empirical results for two different forms of random features. The first uses random selection from the original inputs; the second uses random linear combinations of inputs. The results compare favorably to Adaboost.

The results turn out to be insensitive to the number of features selected to split each node. Usually, selecting one or two features gives near optimum results. To explore this and relate it to strength and correlation, an empirical study is carried out in Section 7.

Adaboost has no random elements and grows an ensemble of trees by successive reweightings of the training set where the current weights depend on the past history of the ensemble formation. But just as a deterministic random number generator can give a good imitation of randomness, my belief is that in its later stages Adaboost is emulating a random forest. Evidence for this conjecture is given in Section 8.

Important recent problems, i.e., medical diagnosis and document retrieval, often have the property that there are many input variables, often in the hundreds or thousands, with each one containing only a small amount of information. A single tree classifier will then have accuracy only slightly better than a random choice of class. But combining trees grown using random features can produce improved accuracy. In Section 9 we experiment on a simulated data set with 1,000 input variables, 1,000 examples in the training set and a 4,000 example test set. Accuracy comparable to the Bayes rate is achieved.

In many applications, understanding of the mechanism of the random forest "black box" is needed. Section 10 makes a start on this by computing internal estimates of variable importance and binding these together by reuse runs.

Section 11 looks at random forests for regression. A bound for the mean squared generalization error is derived that shows that the decrease in error from the individual trees in the forest depends on the correlation between residuals and the mean squared error of the individual trees. Empirical results for regression are in Section 12. Concluding remarks are given in Section 13.

2. Characterizing the accuracy of random forests

2.1. Random forests converge

Given an ensemble of classifiers $h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_K(\mathbf{x})$, and with the training set drawn at random from the distribution of the random vector Y, **X**, define the margin function as

$$mg(\mathbf{X}, Y) = av_k I(h_k(\mathbf{X}) = Y) - \max_{j \neq Y} av_k I(h_k(\mathbf{X}) = j).$$

where $I(\cdot)$ is the indicator function. The margin measures the extent to which the average number of votes at **X**, Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification. The generalization error is given by

$$PE^* = P_{\mathbf{X},Y}(mg(\mathbf{X}, Y) < 0)$$

where the subscripts **X**, Y indicate that the probability is over the **X**, Y space.

In random forests, $h_k(\mathbf{X}) = h(\mathbf{X}, \Theta_k)$. For a large number of trees, it follows from the Strong Law of Large Numbers and the tree structure that:

Theorem 1.2. *As the number of trees increases, for almost surely all sequences* $\Theta_{1,\ldots}$ PE^* *converges to*

$$P_{\mathbf{X},Y}(P_\Theta(h(\mathbf{X}, \Theta) = Y) - \max_{j \neq Y} P_\Theta(h(\mathbf{X}, \Theta) = j) < 0). \tag{1}$$

Proof: see Appendix I. □

This result explains why random forests do not overfit as more trees are added, but produce a limiting value of the generalization error.

2.2. Strength and correlation

For random forests, an upper bound can be derived for the generalization error in terms of two parameters that are measures of how accurate the individual classifiers are and of the dependence between them. The interplay between these two gives the foundation for understanding the workings of random forests. We build on the analysis in Amit and Geman (1997).

Definition 2.1. The margin function for a random forest is

$$mr(\mathbf{X}, Y) = P_\Theta(h(\mathbf{X}, \Theta) = Y) - \max_{j \neq Y} P_\Theta(h(\mathbf{X}, \Theta) = j) \tag{2}$$

and the strength of the set of classifiers $\{h(\mathbf{x}, \Theta)\}$ is

$$s = E_{\mathbf{X},Y} mr(\mathbf{X}, Y). \tag{3}$$

Assuming $s \geq 0$, Chebychev's inequality gives

$$PE^* \leq \text{var}(mr)/s^2 \tag{4}$$

A more revealing expression for the variance of mr is derived in the following: Let

$$\hat{j}(\mathbf{X}, Y) = \arg\max_{j \neq Y} P_\Theta(h(\mathbf{X}, \Theta) = j)$$

so

$$\begin{aligned} mr(\mathbf{X}, Y) &= P_\Theta(h(\mathbf{X}, \Theta) = Y) - P_\Theta(h(\mathbf{X}, \Theta) = \hat{j}(\mathbf{X}, Y)) \\ &= E_\Theta[I(h(\mathbf{X}, \Theta) = Y) - I(h(\mathbf{X}, \Theta) = \hat{j}(\mathbf{X}, Y)]. \end{aligned}$$

Definition 2.2. The raw margin function is

$$rmg(\Theta, \mathbf{X}, Y) = I(h(\mathbf{X}, \Theta) = Y) - I(h(\mathbf{X}, \Theta) = \hat{j}(\mathbf{X}, Y)).$$

Thus, $mr(\mathbf{X}, Y)$ is the expectation of $rmg(\Theta, \mathbf{X}, Y)$ with respect to Θ. For any function f the identity

$$[E_\Theta f(\Theta)]^2 = E_{\Theta,\Theta'} f(\Theta) f(\Theta')$$

holds where Θ, Θ' are independent with the same distribution, implying that

$$mr(\mathbf{X}, Y)^2 = E_{\Theta,\Theta'} rmg(\Theta, \mathbf{X}, Y) rmg(\Theta', \mathbf{X}, Y) \tag{5}$$

Using (5) gives

$$\begin{aligned} \text{var}(mr) &= E_{\Theta,\Theta'}(\text{cov}_{\mathbf{X},Y} rmg(\Theta, \mathbf{X}, Y) rmg(\Theta', \mathbf{X}, Y)) \\ &= E_{\Theta,\Theta'}(\rho(\Theta, \Theta') sd(\Theta) sd(\Theta')) \end{aligned} \tag{6}$$

where $\rho(\Theta, \Theta')$ is the correlation between $rmg(\Theta, \mathbf{X}, Y)$ and $rmg(\Theta', \mathbf{X}, Y)$ holding Θ, Θ' fixed and $sd(\Theta)$ is the standard deviation of $rmg(\Theta, \mathbf{X}, Y)$ holding Θ fixed. Then,

$$\begin{aligned} \text{var}(mr) &= \bar{\rho}(E_\Theta sd(\Theta))^2 \\ &\leq \bar{\rho} E_\Theta \text{var}(\Theta) \end{aligned} \tag{7}$$

where $\bar{\rho}$ is the mean value of the correlation; that is,

$$\bar{\rho} = E_{\Theta,\Theta'}(\rho(\Theta, \Theta')sd(\Theta)sd(\Theta'))/E_{\Theta,\Theta'}(sd(\Theta)sd(\Theta'))$$

Write

$$\begin{aligned} E_{\Theta}\text{var}(\Theta) &\leq E_{\Theta}(E_{\mathbf{X},Y}rmg(\Theta, \mathbf{X}, Y))^2 - s^2 \\ &\leq 1 - s^2. \end{aligned} \tag{8}$$

Putting (4), (7), and (8) together yields:

Theorem 2.3. *An upper bound for the generalization error is given by*

$$PE^* \leq \bar{\rho}(1 - s^2)/s^2.$$

Although the bound is likely to be loose, it fulfills the same suggestive function for random forests as VC-type bounds do for other types of classifiers. It shows that the two ingredients involved in the generalization error for random forests are the strength of the individual classifiers in the forest, and the correlation between them in terms of the raw margin functions. The c/s2 ratio is the correlation divided by the square of the strength. In understanding the functioning of random forests, this ratio will be a helpful guide—the smaller it is, the better.

Definition 2.4. The c/s2 ratio for a random forest is defined as

$$c/s2 = \bar{\rho}/s^2.$$

There are simplifications in the two class situation. The margin function is

$$mr(\mathbf{X}, Y) = 2P_{\Theta}(h(\mathbf{X}, \Theta) = Y) - 1$$

The requirement that the strength is positive (see (4)) becomes similar to the familiar weak learning condition $E_{\mathbf{X},Y}P_{\Theta}(h(\mathbf{X}, \Theta) = Y) > .5$. The raw margin function is $2I(h(\mathbf{X}, \Theta) = Y) - 1$ and the correlation $\bar{\rho}$ is between $I(h(\mathbf{X}, \Theta) = Y)$ and $I(h(\mathbf{X}, \Theta') = Y)$. In particular, if the values for Y are taken to be $+1$ and -1, then

$$\bar{\rho} = E_{\Theta,\Theta'}[\rho(h(\cdot, \Theta), h(\cdot, \Theta')]$$

so that $\bar{\rho}$ is the correlation between two different members of the forest averaged over the Θ, Θ' distribution.

For more than two classes, the measure of strength defined in (3) depends on the forest as well as the individual trees since it is the forest that determines $\hat{j}(\mathbf{X}, Y)$. Another approach

is possible. Write

$$PE^* = P_{\mathbf{X},Y}(P_\Theta(h(\mathbf{X},\Theta) = Y) - \max_{j \neq Y} P_\Theta(h(\mathbf{X},\Theta) = j) < 0)$$
$$\leq \sum_j P_{\mathbf{X},Y}(P_\Theta(h(\mathbf{X},\Theta) = Y) - P_\Theta(h(\mathbf{X},\Theta) = j) < 0).$$

Define

$$s_j = E_{\mathbf{X},Y}(P_\Theta(h(\mathbf{X},\Theta) = Y) - P_\Theta(h(\mathbf{X},\Theta) = j))$$

to be the strength of the set of classifiers $\{h(\mathbf{x},\Theta)\}$ relative to class j. Note that this definition of strength does not depend on the forest. Using Chebyshev's inequality, assuming all $s_{j>0}$ leads to

$$PE^* \leq \sum_j \text{var}(P_\Theta(h(\mathbf{X},\Theta) = Y) - P_\Theta(h(\mathbf{X},\Theta) = j))s_j^2 \tag{9}$$

and using identities similar to those used in deriving (7), the variances in (9) can be expressed in terms of average correlations. I did not use estimates of the quantities in (9) in our empirical study but think they would be interesting in a multiple class problem.

3. Using random features

Some random forests reported in the literature have consistently lower generalization error than others. For instance, random split selection (Dieterrich, 1998) does better than bagging. Breiman's introduction of random noise into the outputs (Breiman, 1998c) also does better. But none of these these three forests do as well as Adaboost (Freund & Schapire, 1996) or other algorithms that work by adaptive reweighting (arcing) of the training set (see Breiman, 1998b; Dieterrich, 1998; Bauer & Kohavi, 1999).

To improve accuracy, the randomness injected has to minimize the correlation $\bar{\rho}$ while maintaining strength. The forests studied here consist of using randomly selected inputs or combinations of inputs at each node to grow each tree. The resulting forests give accuracy that compare favorably with Adaboost. This class of procedures has desirable characteristics:

i Its accuracy is as good as Adaboost and sometimes better.
ii It's relatively robust to outliers and noise.
iii It's faster than bagging or boosting.
iv It gives useful internal estimates of error, strength, correlation and variable importance.
v It's simple and easily parallelized.

Amit and Geman (1997) grew shallow trees for handwritten character recognition using random selection from a large number of geometrically defined features to define the split at each node. Although my implementation is different and not problem specific, it was their work that provided the start for my ideas.

3.1. Using out-of-bag estimates to monitor error, strength, and correlation

In my experiments with random forests, bagging is used in tandem with random feature selection. Each new training set is drawn, with replacement, from the original training set. Then a tree is grown on the new training set using random feature selection. The trees grown are not pruned.

There are two reasons for using bagging. The first is that the use of bagging seems to enhance accuracy when random features are used. The second is that bagging can be used to give ongoing estimates of the generalization error (PE*) of the combined ensemble of trees, as well as estimates for the strength and correlation. These estimates are done out-of-bag, which is explained as follows.

Assume a method for constructing a classifier from any training set. Given a specific training set T, form boostrap training sets T_k, construct classifiers $h(\mathbf{x},\ T_k)$ and let these vote to form the bagged predictor. For each $y, \mathbf{x}$ in the training set, aggregate the votes only over those classifiers for which T_k does not containing $y, \mathbf{x}$. Call this the out-of-bag classifier. Then the out-of-bag estimate for the generalization error is the error rate of the out-of-bag classifier on the training set.

Tibshirani (1996) and Wolpert and Macready (1996), proposed using out-of-bag estimates as an ingredient in estimates of generalization error. Wolpert and Macready worked on regression type problems and proposed a number of methods for estimating the generalization error of bagged predictors. Tibshirani used out-of-bag estimates of variance to estimate generalization error for arbitrary classifiers. The study of error estimates for bagged classifiers in Breiman (1996b), gives empirical evidence to show that the out-of-bag estimate is as accurate as using a test set of the same size as the training set. Therefore, using the out-of-bag error estimate removes the need for a set aside test set.

In each bootstrap training set, about one-third of the instances are left out. Therefore, the out-of-bag estimates are based on combining only about one-third as many classifiers as in the ongoing main combination. Since the error rate decreases as the number of combinations increases, the out-of-bag estimates will tend to overestimate the current error rate. To get unbiased out-of-bag estimates, it is necessary to run past the point where the test set error converges. But unlike cross-validation, where bias is present but its extent unknown, the out-of-bag estimates are unbiased.

Strength and correlation can also be estimated using out-of-bag methods. This gives internal estimates that are helpful in understanding classification accuracy and how to improve it. The details are given in Appendix II. Another application is to the measures of variable importance (see Section 10).

4. Random forests using random input selection

The simplest random forest with random features is formed by selecting at random, at each node, a small group of input variables to split on. Grow the tree using CART methodology to maximum size and do not prune. Denote this procedure by Forest-RI. The size F of the group is fixed. Two values of F were tried. The first used only one randomly selected