Random Forests
10716, Spring 2020
Pradeep Ravikumar (amending notes from Larry Wasserman)

Random forests are a very simple and effective class of models, but there is still a large gap between theory and practice. Basically, a random forest is a simple average of tree estimators, each of which is learnt with some randomization (e.g. using a random subset of features) over a random subsample (typically with replacement, also called bootstrap samples). This is in contrast to boosted trees which computes an adaptive average of sequentially learnt trees. While boosted trees have become very popular as an off-the-shelf method, they come with many tuning parameters to which they are much more sensitive to, as compared to random forests. The two key ingredients, randomization, and bootstrap sampling (also called bagging), also lend themselves very easily to more complex function classes such as DNNs, so it is likely that random forests will see renewed interest in years to come.

These notes rely heavily on Biau, Gerard and Scornet (2016) as well as the other references at the end of the notes.

1 Recap: Partitions and Trees

Recall that simple and interpretable non-parametric classifiers can be derived by partitioning the range of X. Let $\Pi_n = \{A_1, \ldots, A_N\}$ be a partition of $\mathcal{X}$. Let A_j be the partition element that contains x. Then, the partition binary classifier is given as:
$\widehat{h}(x) = 1$ if $\sum_{X_i \in A_j} Y_i \geq \sum_{X_i \in A_j} (1 - Y_i)$ and $\widehat{h}(x) = 0$ otherwise.

This is nothing other than the plugin classifier based on the partition regression estimator

$$\widehat{m}(x) = \sum_{j=1}^{N} \overline{Y}_j \, I(x \in A_j)$$

where $\overline{Y}_j = n_j^{-1} \sum_{i=1}^{n} Y_i I(X_i \in A_j)$ is the average of the Y_i's in A_j and $n_j = \#\{X_i \in A_j\}$, defining $\overline{Y}_j$ to be 0 if $n_j = 0$.

Recall from the results on regression that if $m \in H_1(1, L)$ and the binwidth b of a regular partition satisfies $b \asymp n^{-1/(d+2)}$ then

$$\mathbb{E}||\widehat{m} - m||_P^2 \leq \frac{c}{n^{2/(d+2)}}. \tag{1}$$

We conclude that the corresponding classification risk satisfies $R(\widehat{h}) - R(h_*) = O(n^{-1/(d+2)})$.

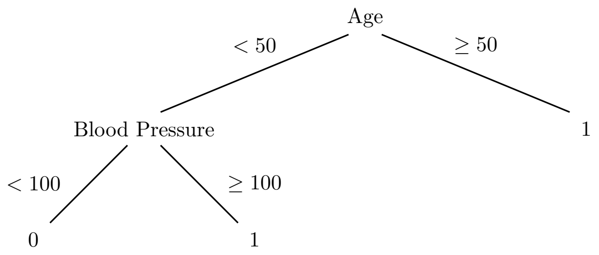

Figure 1: A simple classification tree.

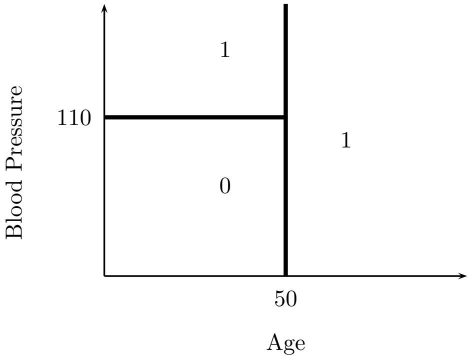

Figure 2: Partition representation of classification tree.

Regression trees and classification trees (also called decision trees) are partition classifiers where the partition is built recursively. For illustration, suppose there are two covariates, $X_1 =$ age and $X_2 =$ blood pressure. Figure 1 shows a classification tree using these variables.

The tree is used in the following way. If a subject has Age ≥ 50 then we classify him as $Y = 1$. If a subject has Age < 50 then we check his blood pressure. If systolic blood pressure is < 100 then we classify him as $Y = 1$, otherwise we classify him as $Y = 0$. Figure 2 shows the same classifier as a partition of the covariate space.

Since estimating the optimal tree is computationally hard, it is constructed greedily. Specifically, we greedily choose among a finite set of splits, and where we measure the quality of each split by the reduction in the loss, which for continuous Y (regression) is typically squared loss, and for binary Y (classification) is typically classification error, or as is more typical, a surrogate for classification error which is smoother, and hence easier to minimize. A common choice for surrogate classification error is the *Gini index*, which for binary classification, defined as

$$\gamma = 1 - \sum_{j=1}^{2} [\overline{Y}_s^2 + (1 - \overline{Y}_s)^2], \tag{2}$$

Method	Test Error
Logistic regression	0.23
SVM (Gaussian Kernel)	0.20
Kernel Regression	0.24
Additive Model	0.20
Reduced Additive Model	0.20
11-NN	0.25
Trees	0.20

Table 1: Various methods on the MAGIC data. The reduced additive model is based on using the three most significant variables from the additive model.

which can be seen to be smoother than the classification error:

$$\text{error} = 1 - \max(\overline{Y}_s, (1 - \overline{Y}_s)).$$

Each split partitions the input space as $\{A_j\}_{j=1}^k$. The overall loss of a split is the weighted average of the losses evaluated in each of the partition components.

We continue recursively splitting until some stopping criterion is met. For example, we might stop when every partition element has fewer than n_0 data points, where n_0 is some fixed number. The bottom nodes of the tree are called the *leaves*. Each leaf has an estimate $\widehat{m}(x)$ which is the mean of Y_i's in that leaf. For classification, we take $\widehat{h}(x) = I(\widehat{m}(x) > 1/2)$.

The result is a piecewise constant estimator that can be represented as a tree.

2 Example

The following data are from simulated images of gamma ray events for the Major Atmospheric Gamma-ray Imaging Cherenkov Telescope (MAGIC) in the Canary Islands. The data are from archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope. The telescope studies gamma ray bursts, active galactic nuclei and supernovae remnants. The goal is to predict if an event is real or is background (hadronic shower). There are 11 predictors that are numerical summaries of the images. We randomly selected 400 training points (200 positive and 200 negative) and 1000 test cases (500 positive and 500 negative). The results of various methods are in Table 1.

3 Bagging

Trees are useful for their simplicity and interpretability. But the prediction error can be reduced by combining many trees. In contrast to boosting, which computes the trees sequentially, and adaptively, a much simpler approach, called bagging, is as follows.

Suppose we are given a set D of n samples. A bootstrap sample is a set of n random samples drawn with replacement from D. Suppose we draw B such bootstrap samples and each time we construct a classifier. This gives tree classifiers $h_1, \ldots, h_B$. (The same idea applies to regression.) We now classify by combining them:

$$h(x) = \begin{cases} 1 & \text{if } \frac{1}{B}\sum_j h_j(x) \geq \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

This is called *bagging* which stands for *bootstrap aggregation.* A variation is sub-bagging where we use random subsamples (without replacement) instead of bootstrap samples.

To get some intuition about why bagging is useful, consider this example from Buhlmann and Yu (2002). Suppose we are given n samples $\{Y_i\}_{i=1}^n \subseteq \mathbb{R}$ drawn from some distribution with mean $\mu = \mathbb{E}[Y_i]$ and variance $\mathrm{Var}(Y_i) = 1$. Then we know that by LLN, the sample mean $\overline{Y}_n$ satisfies:

$$\overline{Y}_n \approx N(\mu, 1/n),$$

so that $\sqrt{n}(\overline{Y}_n - \mu) \approx N(0, 1)$.

Suppose that $y \in \mathbb{R}$ and consider the simple decision rule $\widehat{\theta}_n = I(\overline{Y}_n \leq y)$.

Suppose that y is close to μ relative to the sample size, so that $y \equiv y_n = \mu + c/\sqrt{n}$. Then,

$$\begin{aligned} \widehat{\theta}_n &= I(\overline{Y}_n \leq y) \\ &= I(\sqrt{n}(\overline{Y}_n - \mu) \leq c), \end{aligned}$$

which by LLN converges to $I(Z \leq c)$, where $Z \sim N(0, 1)$. So the limiting mean and variance of $\widehat{\theta}_n$ are $\Phi(c)$ and $\Phi(c)(1 - \Phi(c))$, where $\Phi(\cdot)$ is the CDF of $N(0, 1)$.

Now the bootstrap distribution of $\overline{Y}^*$ (conditional on $Y_1, \ldots, Y_n$) in turn is approximately $N(\overline{Y}, 1/n)$, so that $\sqrt{n}(\overline{Y}^* - \overline{Y}) \approx N(0, 1)$. Let E^* denote the average with respect to the bootstrap randomness. Then, if $\widetilde{\theta}_n$ is the bagged estimator, we have

$$\begin{aligned} \widetilde{\theta}_n = E^*[I(\overline{Y}^* \leq y_n)] &= E^*\left[I\left(\sqrt{n}(\overline{Y}^* - \overline{Y}) \leq \sqrt{n}(y_n - \overline{Y})\right)\right] \\ &= \Phi(\sqrt{n}(y_n - \overline{Y})) + o(1) \\ &= \Phi(c + \sqrt{n}(\mu - \overline{Y})) = \Phi(c + Z) + o(1), \end{aligned}$$

where $Z \sim N(0,1)$.

To summarize, $\widehat{\theta}_n \approx I(Z \leq c)$ while $\widetilde{\theta}_n \approx \Phi(c+Z)$ which is a smoothed version of $I(Z \leq c)$. In other words, bagging is a smoothing operator. In particular, suppose we take $c = 0$. Then $\widehat{\theta}_n$ converges to a Bernoulli with mean $\Phi(0) = 1/2$ and variance $\Phi(0)(1-\Phi(0)) = 1/4$. The bagged estimator converges to $\Phi(Z) = \mathrm{Unif}(0,1)$ which has mean 1/2 and variance 1/12. The reduction in variance is due to the smoothing effect of bagging.

4 Random Forests

Finally we get to random forests. These are bagged trees except that we also choose random subsets of features for each tree. The estimator can be written as

$$\widehat{m}(x) = \frac{1}{M}\sum_{j=1}^{M} \widehat{m}_j(x)$$

where $\widehat{m}_j$ is a tree estimator based on a subsample (or bootstrap) of size a using p randomly selected features. The trees are usually required to have some number k of observations in the leaves. There are three tuning parameters: a, p and k. You could also think of M as a tuning parameter but generally we can think of M as tending to ∞.

For each tree, we can estimate the prediction error on the un-used data. (The tree is built on a subsample.) Averaging these prediction errors gives an estimate called the *out-of-bag* error estimate.

Unfortunately, it is very difficult to develop theory for random forests since the splitting is done using greedy methods. Much of the theoretical analysis is thus done using simplified versions of random forests.

One such simplification is the so-called *centered forest* which is defined as follows. Suppose the data are on $[0,1]^d$. Choose a random feature, split in the center. Repeat until there are k leaves. This defines one tree. Now we average M such trees. Breiman (2004) and Biau (2002) proved the following.

Theorem 1 *If each feature is selected with probability* $1/d$, $k = o(n)$ *and* $k \to \infty$ *then*

$$\mathbb{E}[|\widehat{m}(X) - m(X)|^2] \to 0$$

as $n \to \infty$.

A significant step forward was made by Scornet, Biau and Vert (2015). Here is their result.

Theorem 2 *Suppose that the response is specified by an additive model:*

$$Y = \sum_j m_j(X_j) + \epsilon,$$

where $X \sim \text{Uniform}[0,1]^d$, $\epsilon \sim N(0,\sigma^2)$ and each m_j is continuous. Assume that the split is greedily chosen based on minimizing the squared loss. Let k_n be the number of leaves on each tree and let a_n be the subsample size. If $k_n \to \infty$, $a_n \to \infty$ and $k_n(\log a_n)^9/a_n \to 0$ then

$$\mathbb{E}[|\widehat{m}(X) - m(X)|^2] \to 0$$

as $n \to \infty$.

The theorem has strong assumptions, but it does allow for greedy split selection.

5 Connection to Nearest Neighbors

Lin and Jeon (2006) showed that there is a connection between random forests and k-NN methods. We say that X_i is a *layered nearest neighbor* (LNN) of x if the hyper-rectangle defined by x and X_i contains no data points except X_i. Now note that if tree is grown until each leaf has one point, then $\widehat{m}(x)$ is simply a weighted average of LNN's. More generally, Lin and Jeon (2006) call X_i a k-potential nearest neighbor k-PNN if there are fewer than k samples in the the hyper-rectangle defined by x and X_i. If we restrict to random forests whose leaves have k points then it follows easily that $\widehat{m}(x)$ is some weighted average of the k-PNN's.

Let us know return to LNN's. Let $\mathcal{L}_n(x)$ denote all LNN's of x and let $L_n(x) = |\mathcal{L}_n(x)|$. We could directly define

$$\widehat{m}(x) = \frac{1}{L_n(x)} \sum_i Y_i I(X_i \in \mathcal{L}_n(x)).$$

Biau and Devroye (2010) showed that, if X has a continuous density, Y is bounded, and m is continuous then, for all $p \geq 1$,

$$\mathbb{E}|\widehat{m}_n(X) - m(X)|^p \to 0$$

as $n \to \infty$.

Unfortunately, the rate of convergence is slow.

They also showed that if X has a continuous density,

$$\frac{(d-1)!\mathbb{E}[L_n(x)]}{2^d(\log n)^{d-1}} \to 1.$$

Suppose that $\mathrm{Var}(Y|X=x)=\sigma^2$ is constant. Then

$$\mathbb{E}|\widehat{m}_n(X)-m(X)|^p \geq \frac{\sigma^2}{\mathbb{E}[L_n(x)]} \sim \frac{\sigma^2(d-1)!}{2^d(\log n)^{d-1}}.$$

If we use k-PNN, with $k\to\infty$ and $k=o(n)$, then the results Lin and Jeon (2006) show that the estimator is consistent and has variance of order $O(1/k(\log n)^{d-1})$.

As an aside, Biau and Devroye (2010) also show that if we apply bagging to the usual 1-NN rule to subsamples of size k and then average over subsamples, then, if $k\to\infty$ and $k=o(n)$, then for all $p\geq 1$ and all distributions P, we have that $\mathbb{E}|\widehat{m}(X)-m(X)|^p\to 0$. So bagged 1-NN is universally consistent. But at this point, we have wondered quite far from random forests.

6 Connection to Kernel Methods

There is also a connection between random forests and kernel methods (Scornet 2016). Let $A_j(x)$ be the cell containing x in the j^{th} tree. Then we can write the tree estimator as

$$\widehat{m}(x)=\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{n}\frac{Y_i I(X_i\in A_j(x))}{N_j(x)}$$

where $N_j(x)$ is the number of data points in $A_j(x)$. Based on this observation, Scornet (2016) defined kernel based random forest (KeRF) by

$$\widehat{m}(x)=\frac{\sum_{i=1}^{n}\sum_{j=1}^{M} Y_i I(X_i\in A_j(x))}{\sum_{j=1}^{M} N_j(x)}.$$

With this modification, $\widehat{m}(x)$ is the average of each Y_i weighted by how often it appears in the trees. The KeRF can be written as

$$\widehat{m}(x)=\frac{\sum_{i=1}^{n} Y_i K(x,X_i)}{\sum_{i=1}^{n} K(x,X_i)}$$

where

$$K(x,z)=\frac{1}{M}\sum_{j=1}^{M} I(z\in A_j(x)).$$

The trees are random. So let us write the j^{th} tree as $T_j=T(\Theta_j)$ for some random quantity Θ_j. So the forests is built from $T(\Theta_1),\ldots,T(\Theta_M)$. And we can write $A_j(x)$ as $A(x,\Theta_j)$. Then $K(x,z)$ converges almost surely (as $M\to\infty$) to $\kappa(x,z)=P_\Theta(z\in A(x,\Theta))$ which is just the probability that x and z are connected, in the sense that they are in the same cell. Under some assumptions, Scornet (2016) showed that KeRF's and forests are close to each other, thus providing a kernel interpretation of forests.

7 Variable Importance

Let $\widehat{m}$ be a random forest estimator. How important is feature X_ℓ?

LOCO. One way to answer this question is to fit the forest with all the data and fit it again without using X_ℓ. When we construct a forest, we randomly select features for each tree. This second forest can be obtained by simply average the trees where feature ℓ was not selected. Call this $\widehat{m}_{(-\ell)}$. Let V be a hold-out sample of size m. Then let

$$\widehat{\Delta}_\ell = \frac{1}{m}\sum_{i\in V} W_i$$

where

$$W_i = (Y_i - \widehat{m}_{(-\ell)}(X_i))^2 - (Y_i - \widehat{m}(X_i))^2.$$

Then $\widehat{\Delta}_\ell$ is a consistent estimate of the prediction risk inflation that occurs by not having access to X_ℓ. Formally, if $\mathcal{T}$ denotes the training data then,

$$\mathbb{E}[\widehat{\Delta}_\ell|\mathcal{T}] = \mathbb{E}\left[(Y - \widehat{m}_{(-\ell)}(X))^2 - (Y - \widehat{m}(X))^2 \;\middle|\; \mathcal{T}\right] \equiv \Delta_\ell.$$

This approach is called LOCO (Leave-Out-COvariates).

Permutation Importance. A different approach is to permute the values of X_ℓ for the out-of-bag observations, for each tree. Let $\mathcal{O}_j$ be the out-of-bag observations for tree j and let $\mathcal{O}_j^*$ be the out-of-bag observations for tree j with X_ℓ permuted, and suppose that $m_j = |\mathcal{O}_j| = |\mathcal{O}_j^*|$. Then, let

$$\widehat{\Gamma}_\ell = \frac{1}{M}\sum_{j=1}^{M} W_j$$

where

$$W_j = \frac{1}{m_j}\sum_{i\in\mathcal{O}_j^*}(Y_i - \widehat{m}_j(X_i))^2 - \frac{1}{m_j}\sum_{i\in\mathcal{O}_j}(Y_i - \widehat{m}_j(X_i))^2.$$

This avoids using a hold-out sample. This is estimating

$$\Gamma_\ell = \mathbb{E}[(Y - \widehat{m}(X^{(\ell)}))^2] - \mathbb{E}[(Y - \widehat{m}(X))^2]$$

where $X^{(\ell)}$ has the same distribution as X except that $X_\ell^{(\ell)}$ is an independent draw from X_ℓ.

In the additive model case where $Y = \sum_{\ell=1}^d m_\ell(X_\ell) + \epsilon$ with $\mathbb{E}[\epsilon|X] = 0$ and $\mathbb{E}[\epsilon^2|X] < \infty$, they show that $\Gamma_\ell = 2\mathrm{Var}(m_\ell(X_\ell)$.

8 Summary

Random forests are considered one of the best all purpose classifiers. But it is still a mystery why they work so well. The situation is very similar to deep learning. We have seen that there are now many interesting theoretical results about forests. But the results make strong assumptions that create a gap between practice and theory. Furthermore, there is no theory to say why forests outperform other methods. The gap between theory and practice is due to the fact that forests — as actually used in practice — are complex functions of the data.

9 References

Biau, Devroye and Lugosi. (2008). Consistency of Random Forests and Other Average Classifiers. *JMLR*.

Biau, Gerard, and Scornet. (2016). A random forest guided tour. Test 25.2: 197-227.

Biau, G. (2012). Analysis of a Random Forests Model. arXiv:1005.0208.

Buhlmann, P., and Yu, B. (2002). Analyzing bagging. Annals of Statistics, 927-961.

Gregorutti, Michel, and Saint Pierre. (2013). Correlation and variable importance in random forests. arXiv:1310.5726.

Lei J, G'Sell M, Rinaldo A, Tibshirani RJ, Wasserman L. (2017). Distribution-free predictive inference for regression. Journal of the American Statistical Association.

Lin, Y. and Jeon, Y. (2006). Random Forests and Adaptive Nearest Neighbors. *Journal of the American Statistical Association*, 101, p 578.

L. Mentch and G. Hooker. (2015). Ensemble trees and CLTs: Statistical inference for supervised learning. Journal of Machine Learning Research.

Rinaldo A, Tibshirani R, Wasserman L. (2015). Uniform asymptotic inference and the bootstrap after model selection. arXiv preprint arXiv:1506.06266.

Scornet E. Random forests and kernel methods. (2016). IEEE Transactions on Information Theory. 62(3):1485-500.

Wager, S. (2014). Asymptotic Theory for Random Forests. arXiv:1405.0352.

Wager, S. (2015). Uniform convergence of random forests via adaptive concentration. arXiv:1503.06388.

Wager, S. and Athey, S. (2017). Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association.