Random Forests

Stephen Vardeman

Analytics Iowa LLC

ISU Statistics and IMSE

Bagging of special trees

This is an elaboration of bagging applied to tree prediction. Suppose that one makes B bootstrap samples of size N (random samples with replacement of size N) from the training set $\mathbf{T}$, say $\mathbf{T}_1^*, \mathbf{T}_2^*, \ldots, \mathbf{T}_B^*$. For each sample, $\mathbf{T}_b^*$, develop a corresponding regression or classification tree by

1. at each node, randomly selecting m of the p input variables and finding an optimal single split of the corresponding rectangle over the selected input variables (that reduces the splitting criterion), splitting the rectangle, and
2. repeating 1 at each node up to a fixed depth or until no single-split improvement in splitting criterion is possible without creating a rectangle with less than a small number of training cases, $n_{\min}$.

Let $\hat{f}^{*b}(\mathbf{x})$ be the corresponding tree-based predictor (taking values in $\Re$ in the regression case or in $\mathcal{G} = \{1, 2, \ldots, K\}$ in the classification case).

Random forests for regression and classification

As in any SEL bagging case, a random forest predictor is then

$$\hat{f}_B^* (\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b} (\mathbf{x})$$

and in any 0-1 loss classification case, a random forest classifier is

$$\hat{f}_B^* (\mathbf{x}) = \arg\max_k \sum_{b=1}^{B} I \left[\hat{f}^{*b} (\mathbf{x}) = k \right]$$

and out-of-bag errors are made as for any application of bagging to these contexts.

We note again that the number of bootstrap samples (and trees here) is a convergence parameter and not a complexity parameter.

Tuning/complexity parameters

The basic tuning parameters in the development of $\hat{f}_B^*(\mathbf{x})$ are then m, and $n_{\min}$, and (if used) a maximum tree depth. The standard default values of parameters are

- $m = \lfloor p/3 \rfloor$ and $n_{\min} = 5$ for regression problems, and
- $m = \lfloor \sqrt{p} \rfloor$ and $n_{\min} = 1$ for classification problems.

(The default $n_{\min} = 1$ means that splitting terminates only because of reaching a maximum depth or the impossibility of reducing the splitting criterion with a single additional split.)

Rather than using default values of parameters, those can be chosen to minimize the (large B) OOB error. **Bagging provides its own "internal" version of cross-validation and there is no need to wrap another cross-validation around a random forest in order to approximate Err.**

0 training error for random forest classifiers

The default $n_{\text{min}} = 1$ for classification problems means that splitting terminates only when reaching a maximum depth or completely homogeneous rectangles. If maximum tree depth really doesn't come into play (because it is set to some value that is large in relative terms) this produces random forest classifiers with 0 training error rate. Every training case will be missed by only about 37% of B bootstrap samples, so that about 63% of the B bootstrap trees correctly classify the case. So majority voting means that the random forest will correctly classify the case. But notice that this does **not** imply that the out-of-bag-error $\text{OOB}(B)$ will be 0. And it does **not** imply that $\text{OOB}(B)$ for large B is unreliable as an indicator of the likely performance of random forest classifier. It only implies that $\overline{\text{err}} = 0$ is completely *unreliable* as an indicator of random forest classifier efficacy.

?? No over-fitting ??

There is a fair amount of confusing discussion in the literature about the impossibility of a random forest "over-fitting" with increasing B. This seems to be related to test error *not* initially-decreasing-but-then-increasing-in-B (which is perhaps loosely related to $\mathrm{OOB}(B)$ converging to a positive value associated with the limiting predictor $\hat{f}^{\mathrm{rf}}$ and/or 0 training error rate for a random forest classifier not implying over-fit). But as HTF point out on their page 596, it is an entirely different question as to whether $\hat{f}^{\mathrm{rf}}$ itself is "too complex" to be adequately supported by the training data, $\mathbf{T}$. (And the whole discussion seems very odd in light of the fact that for any finite B, a different choice of bootstrap samples would produce a different $\hat{f}_B^*$ as a new randomized approximation to $\hat{f}^{\mathrm{rf}}$. Even for fixed $\mathbf{x}$, the value $\hat{f}_B^*(\mathbf{x})$ is a random variable. Only $\hat{f}^{\mathrm{rf}}(\mathbf{x})$ is fixed.)

m and "correlation" and "strength" of trees

There is also a fair amount of confusing discussion in the literature about the role of the *random* selection of the m predictors to use at each node-splitting (and the choice of m) in reducing "correlation between trees in the forest." The Breiman/Cutler web site `http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm` says that the "forest error rate" (presumably the error rate for $\hat{f}^{\text{rf}}$) depends upon "the correlation between any two trees in the forest" and the "strength of each tree in the forest." While this assertion is perhaps believable in some qualitative sense (if "correlation" means "similarity" and "strength" means individual "predictive effectiveness") a precise technical meaning is not clear.

Possible precise meanings

One possible technical meaning of "correlation between trees" is some version of correlation between values of $\hat{f}^{*1}(\mathbf{x})$ and $\hat{f}^{*2}(\mathbf{x})$ as one repeatedly selects *the whole training set* $\mathbf{T}$ in iid fashion from P and then makes two bootstrap samples—Section 15.4 of HTF seems to use this meaning.

A second possibility concerns "bootstrap randomization distribution" correlation (for a fixed training set and a fixed $\mathbf{x}$) between values $\hat{f}^{*1}(\mathbf{x})$ and $\hat{f}^{*2}(\mathbf{x})$.

A possible technical meaning of "tree strength" is some test error for a single $\hat{f}^{*b}$.

m and “correlation” and “strength” of trees cont.

HTF Section 15.4 goes on to suggest that increasing *m* increases both "correlation" between and "strength" of the individual trees, the first degrading error rate and the second improving it, and that the OOB estimate of error can be used to guide choice of *m* (usually in a broad range of values that are about equally attractive) if something besides the default is to be used.