Bagging, Feature Subsets and Random Forests

Motivation:

- Ensembles can help reduce variance or improve classification accuracy.
- The improvement over stage 1 models is largest if predictions from stage 1 models are uncorrelated.

Our Goal:

- Manufacture a large # of uncorrelated component models to use in an ensemble

2 Strategies:

1) Train different component models on different sets of rows of the data (bagging)
2) Train different component models on different sets of columns of the data (feature subsets)
   → or at least don't use columns in the same way.

Bagging (bootstrap aggregation):

- Train B (often B = 500 or 1000) models, each to its own data set of observations drawn with replacement from the original data.

  ↑ a bootstrap sample

Algorithm:

1. Allocate space to save test set predictions from all B models
2. For b = 1, ..., B:
   a. Draw a bootstrap sample from the original data
   b. Fit model to the bootstrap sample from step a.
   c. Obtain test set predictions and save them
3. Ensemble combines component model predictions by majority vote or average.

Feature subsets:

- Similar to bagging, but each model is trained on a different randomly selected subset of the explanatory variables.

Random Forests:

Combine the following

~~bagging~~

- component models are classification or regression trees
- bagging: each tree is estimated using a different bootstrap sample
- when growing trees, for each possible split consider only splits that can be made with a randomly selected subset of explanatory variables.

  ↳ In caret::train, parameter is mtry

  * smaller mtry means more bias (may omit important explanatory variables)
  * smaller mtry means lower variance (less of a chance different trees in the forest will use the same variable to split
    → less correlated predictions from different trees
    → lower variance)

Estimating Test Set Performance with Bagging:

- Each tree in random forest trained using a bootstrap sample (sampled with replacement)
- For each tree, some observations not used to fit that tree.
  ↳ use these to estimate test set performance

"Out-of-bag" (OOB) procedure for estimating test set error:

1) Obtain the out-of-bag prediction for each training set observation $i = 1, \ldots, n$
   - there will be about $B/3$ bootstrap samples that did not include observation i
   - $\hat{y}_i^{OOB}$ is the mean (regression) or majority vote (classification) prediction from models not trained using observation i.

2) Use the OOB predictions $\hat{y}_1^{OOB}, \ldots, \hat{y}_n^{OOB}$ to estimate test set performance:
   - ~~Est~~ Estimate test set MSE: $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{OOB})^2$
   - Estimate test set classification error rate:
     $$\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i \neq \hat{y}_i^{OOB})$$

What proportion of observations are not in ~~any~~ a particular bootstrap samples?

$P(\text{Obs. } i \text{ not in bootstrap sample})$

$= P(\text{Obs. 1 in bootstrap sample is not } i)$
$\quad * P(\text{Obs. 2 in bootstrap sample is not } i)$
$\quad * \cdots$
$\quad * P(\text{Obs. } n \text{ in bootstrap sample is not } i)$

$= \left(\frac{n-1}{n}\right)^n$

For example $0.99^{100} = 0.366$ so a given bootstrap sample will tend to contain about 63% of the observations in the original data set.

Also, $\lim_{n\to\infty} \left(\frac{n-1}{n}\right)^n = \lim_{n\to\infty} \exp\left[\log\left\{\left(\frac{n-1}{n}\right)^n\right\}\right]$

$= \lim_{n\to\infty} \exp\left[n\left\{\log(n-1) - \log(n)\right\}\right]$

$= \lim_{n\to\infty} \exp\left[\frac{\log(n-1) - \log(n)}{1/n}\right]$

$= \lim_{n\to\infty} \exp\left[\frac{\frac{1}{n-1} - \frac{1}{n}}{-\frac{1}{n^2}}\right]$ l'Hopital's rule

$= \lim_{n\to\infty} \exp\left(\frac{-n^2}{n-1} + \frac{n(n-1)}{n-1}\right)$

$= \lim_{n\to\infty} \exp\left(\frac{-n}{n-1}\right) = e^{-1} = \frac{1}{e} \approx 0.368$

Variable importance from Bagging:

2 approaches: ~~here is the default~~:

Total decrease in RSS (for regression)
or Gini index (classification)
associated with splits on a certain variable,
averaged across, all trees in the forest.