Bootstrap Aggregation (Bagging) and Random Forests

Instructor: Haoran LEI

Hunan University

Pros of Tree-based Methods

- Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
- Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches seen in previous chapters.
- Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small).
- Trees can easily handle qualitative predictors without the need to create dummy variables.

From single-tree to many-trees

- However, trees generally do not have the same level of predictive accuracy as some of the other regression (and classification approaches) covered before.
- By aggregating many decision trees, the predictive performance of trees can be substantially improved. We introduce these concepts next.

Bagging

- *Bootstrap aggregation*, or *bagging*, is a general-purpose procedure for reducing the variance of a statistical learning method
  - it is particularly useful and frequently used in the context of decision trees.
- Recall that given a set of n independent observations $Z_1, \ldots, Z_n$, each with variance σ^2, the variance of the mean $\bar{Z}$ of the observations is given by σ^2/n.
  - In other words, averaging a set of observations reduces variance. (But we usually only have one sample set)

Bagging continued

- Instead, we can bootstrap, by taking repeated samples from the (single) training data set.
- In this approach we generate B different (bootstrapped) training data sets.
  - We then train our method on the bth bootstrapped training set in order to get $\hat{f}^{*b}(x)$, the prediction at a point x.
- Average all the predictions to obtain

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

Bagging regression trees

- The above prescription applied to regression trees:
  - For each $b \in \{1, ..., B\}$, we compute the prediction $\hat{f}^{*b}(x)$ by buiding the b-th tree as discussed before
  - The final prediction is the average of all the B predictions.

Out-of-Bag Error

- There is a very straightforward way to **estimate the test error of a bagged model**.
- Recall that the key to bagging is that trees are repeatedly fit to bootstrapped subsets of the observations.
  - One can show that on average, each bagged tree makes use of **around two-thirds** of the observations. (**Why?**)
  - The remaining one-third of the observations not used to fit a given bagged tree are referred to as *the out-of-bag (OOB)* observations.

Out-of-Bag Error Estimation

- We can predict the response for the i-th observation using each of the trees in which that observation was OOB. This will yield around $B/3$ predictions for the i-th observation, which we average.
- This estimate is essentially the *LOO cross-validation error* for bagging, if B is large.
- Therefore, use the magic formula for LOO cross-validation.

Random Forests

- *Random forests* provide an improvement over bagged trees by way of a small tweak that *decorrelates* the trees. This reduces the variance when we average the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- But when building these decision trees, each time a split in a tree is considered, *a random selection* of m predictors is chosen as split candidates from the full set of p predictors.
    - The split is allowed to use only one of those m predictors.

Random Forests Cont.

- A fresh selection of m predictors is taken at each split, and typically we choose $m \approx \sqrt{p}$
  - That is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors.
- By focusing on $m(\approx \sqrt{p})$ predictors, each time we grow a very small tree. This *decorrelates* the trees.
  - In practice, this method also can prevent over-fitting.