CHL5212H Assignment 4: Random Forests and Gradient Boosting

The goal of this assignment is to practice fitting and tuning random forest and gradient boosting models. Split the data into a training and test set using a 50/50 split.

Random Forest:
Tuning of the random forest is determined by two parameters; the number of trees fit in the ensemble (ntree) and the number of predictors sampled for each tree (mtry). The number of trees is determined by the OOB error of the bootstrap, but it is necessary to fit different random forests for each value of mtry.

Fit a random forest using all predictors and the default mtry value (square-root of the number of variables) on the training set. Set the ensemble size to 1200 and fit the RF using 0.10, 0.25, 0.50, 0.75, 1.0 and 1.25 times the default mtry value and determine the best number of variables to sample in the ensemble. Estimate the prediction error on the test set.

Gradient Boosting:
Tuning of gradient boosting is determined by the number of trees, the depth of the trees and the weight of the trees (shrinkage). The number of trees can be obtained by cross-validation from a particular fit, but it is necessary to do a grid search for the optimal shrinkage-tree depth combination.

Fit a gradient boosting model using a package of your choice and try tree depths of 1 (stump), 2, 3 and 4 and shrinkage values of 0.05, 0.10, 0.25 and 0.50. Determine the optimal number of trees and the best combination of parameters using 10 fold cross-validation from the training set. Pick the best model using CV and estimate the error on the test set. For each tree depth, plot the test set errors vs. number of trees for different shrinkage values.

Note: A stump is a tree with a single split, some packages set this depth at 0, in that case use depths of 0, 1, 2 and 3.

Conclusions:

Random Forest:

How would you adopt the RF fitting algorithm to use cross-validation, instead of OOB?

In what instance would using OOB become problematic, compared to CV?

What was the effect of choosing different values of mtry on the number of trees selected? What type of data structure would work best with high/low mtry values?

Conceptually, what is the implicit tradeoff between the values of ntree and mtry in a random forest? By tuning these parameters, what assumptions of the algorithm are we trying to satisfy?

Why is it not necessary to prune trees in a RF model?

What can be done improve the smoothness of the error function?

Gradient Boosting:

How much effect did the different tuning parameters have on performance? What were your optimal parameters and what do they tell you about the structure of the signal?

Was the optimal number of trees in the ensemble comparable to RF? Would this be expected?

What is the implicit trade-off between the shrinkage and tree depth parameters? What type of signal do you expect when setting a high value for shrinkage? A high value for tree depth?

How is the effect of the existing trees incorporated into the fit?

Compare the variable ranks from the two approaches. Is there a lot of overlap? Would these results be expected based on the models fit in A2 and A3?

Under what scenario/signal structure would you expect GBM to outperform linear/parametric models? When would it under-perform? (Consider the advantages/disadvantages of using trees.)

Evaluation:
Results (20%)
Coding style (20%)
Interpretation (60%): Answers need to be interpretive rather than descriptive. Use clear and precise language and reference the results to support your claims when appropriate. Split is 30% for RF and 30% for GBM.

Due: June 16, 2020 at 12:00pm