**Bagging & Random Forest Cheat Sheet**

Duncan Wingfield, Jessy Sun, Nada Liang, Travis Li

Team 17, Section 2

## "Ensemble" Model

The basic idea of “ensemble” model is to combine multiple models into a single "ensemble" model. The more ‘decorrelated’ the models, the better.

* Examples:
  + Boosting
  + Bagging
  + random forests
* Advantage:
  + reduces variability
  + more robust to changes in the data
  + tolerant of noise

## Bagging

This technique is widely used by random forest. Therefore, we mainly explain its usage in random forest here.

#### Definition

* bagging is the idea of collecting a random sample of observations (with replacement) into a bag (though the term is actually an abbreviation of "bootstrap aggregation").

#### Process

* A "bag" of observations is randomly selected from the original training dataset.
  + The sample size is often the same as for the full training dataset.
  + In general, about two-thirds of the observations will be included in the bag (with repeats) and one-third will be left out .
  + The reason is that, if there are n observations, for each observation, the probability of not being chosen is 1-1/n; therefore, the probability of not being chosen for n times is (1-1/n)^n. When n approaches to positive infinite, this probability will approaches 1/3.
  + Each bag of observations is then used as the training dataset for building a decision tree, and those left out make up the test set.
* The observations left out is called “out-of-bag(OOB): observations”.
  + The OOB test error is equivalent to LOOCV.
  + The OOB test error is a great estimate of test error rate.

## 

## Random Forests

Random forest is to combine multiple decisions trees into a single "ensemble" model (forest).

#### When To Use：

* Random forests handle underrepresented classification tasks well.
  + Underrepresentation occurs when one class has very few (e.g., 5% or fewer) observations compared with the other class(es).
* Random forests are particularly suitable when there are very many input variables and only a limited number of observations.
* Random forest can be used in both regression and classification models.

#### Randomness In Random Forest:

* there are two types of randomness here:
  + The selection observations to use (bagging)
  + The variables to consider for splits
* advantage of introducing randomness:
  + It is this randomness that delivers considerable robustness to noise, outliers, and overfitting, when compared with a single-tree classier.
  + Randomness delivers substantial computational efficiencies.

#### How To Build Single Trees In The Random Forest Algorithm:

* The random forest algorithm will build many single decision trees. For each decision tree, the algorithm will select a random subset of the observations available in the training dataset.
  + This uses “Bagging”.
* At each node in the process of building the decision tree, only a small fraction of all of the available variables (chosen randomly) are considered when determining how to best partition the dataset.
  + As a general rule of thumb, the number of predictors for each split(m) should be equal to m=√p (p is the total number of predictors). m is a hyperparameters here.
* Each decision tree is built to its maximal depth, not pruning the individual decision trees.
  + Pruning was necessary to avoid overfitting the data. However, a random forest of overfitted trees can deliver a very good model that performs well on new data.
  + we can end up with a model that is also less biased.

#### How To Aggregate The Decisions From All Trees Into One Final Decision:

* For classification forests, there are 3 alternatives:
  + Simple majority rules

If 80 out of 100 trees in the random forest say that the customer will defect, then we will go with that decision.

Even if 51 of the 100 trees say that the customer will defect, we might go with that, although perhaps with less certainty.

* + Computing a weighted score where the weight of each individual tree corresponds to the quality of the accuracy of that tree.
  + By default, the random forest algorithm that we will employ uses the "majority rules" alternative.
* For regression forests:
  + the result is the average value (or weighted average value as above) over the ensemble of regression trees.

#### Variable Importance:

* Although random forest is more difficult to interpret than a single tree, it can obtain an overall summary of the importance of each predictor.
* A large value indicates an important predictor

Process

* Train Random Forest with the original dataset and get a matric statistic: accuracy (for classification) or RSS (for regression).
* Take one column and randomly permute it. Then use the Random Forest model trained before to make prediction and get a new metric statistic.
* Repeat that with all the columns. Compare the decrease values of the metric statistics. The bigger the difference means the more importance of that permuted column.
* You can also use other metric to compare, for instance, GINI index for the classification problem.

ROC curve:

An incredibly useful tool in evaluating and comparing predictive models is the ROC curve. It’s especially good when the dataset is imbalanced (since accuracy will be a weaker metric).

#### Definition

* The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. It equals to 1-Type II error against Type I error.

Process

* The threshold decides how many predicted values are classified as positive or negative. For example, there are three predicted probabilities 0.3,0.6,0.9; If the cutoff (threshold) is 0.5, then there are 2 positive 1 negative values; When the cutoff changes to 0.7, the number of positive value decreases to 1 while the negative values to 2;
* For each threshold, we can get a confusion table with predicted classes against ground truth classes, and Type I and Type II error thereafter. WIth those statistics, we may draw a point in the coordinate (with Type I error at the x-axis, while 1-Type II at the y-axis). As the threshold changes (0,1), the points will form a curve, which is the ROC curve.

Interpretation

The more the curve inclines to the left upper corner, the better the model is; Another way to look is the bigger the area under the curve (AUC), the better the model is;

## In R:

* Use library(“randomForest”)
* randomForest(y ~ ., x, ntree = # of trees, mtry = # of predictors each split,

importance=TRUE, localImp=TRUE, na.action=na.roughfix, replace = TRUE)

* ntree: number of trees of the model to grow, don’t set this number too small
* mtry: number of variables randomly sampled as candidates at each split. Default value for classification is sqrt(# of predictors), for regression is (# of predictors) / 3
* importance = TRUE makes sure importance of predictors is assessed
* localImp = TRUE makes sure casewise importance is assessed (override *importance*)
* replacement = TRUE, sampling with replacement (concept of bagging)

When modeling bagging trees model, set mtry = all predictors

When modeling random forest model, set mtry = number less than number of predictors

* importance() is a builtin function of randomForest() which assesses variable importance if importance = TRUE
* varImpPlot(), plot the variable importance