Ensemble Methods

Nisha Iyer

April 1, 2016

### I will use the well known Boston Housing Data set to build out a random forest model, followed

### by an AdaBoost model. I want to use the same data set for comparison of these two methods.

library(MASS)  
library(caret)

## Loading required package: lattice  
## Loading required package: ggplot2

library(randomForest)

## randomForest 4.6-12  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library(ROCR)

## Loading required package: gplots  
##   
## Attaching package: 'gplots'  
##   
## The following object is masked from 'package:stats':  
##   
## lowess

library(miscTools)  
  
#load Boston Housing data   
#This example will be regression trees since the target variable is continuous (median value).  
data(Boston)  
head(Boston)

## crim zn indus chas nox rm age dis rad tax ptratio black  
## 1 0.00632 18 2.31 0 0.538 6.575 65.2 4.0900 1 296 15.3 396.90  
## 2 0.02731 0 7.07 0 0.469 6.421 78.9 4.9671 2 242 17.8 396.90  
## 3 0.02729 0 7.07 0 0.469 7.185 61.1 4.9671 2 242 17.8 392.83  
## 4 0.03237 0 2.18 0 0.458 6.998 45.8 6.0622 3 222 18.7 394.63  
## 5 0.06905 0 2.18 0 0.458 7.147 54.2 6.0622 3 222 18.7 396.90  
## 6 0.02985 0 2.18 0 0.458 6.430 58.7 6.0622 3 222 18.7 394.12  
## lstat medv  
## 1 4.98 24.0  
## 2 9.14 21.6  
## 3 4.03 34.7  
## 4 2.94 33.4  
## 5 5.33 36.2  
## 6 5.21 28.7

dim(Boston)

## [1] 506 14

#Use createDataPartition from the caret package to create train and test sets  
set.seed(123)  
split <- createDataPartition(y=Boston$medv, p = 0.7, list=FALSE)  
train <- Boston[split,]  
test<- Boston[-split,]  
  
#A note about 'createDataPartition; createResample can be used to make a simple bootstrap and createFolds for   
#cross-validation groupings.

### OK, training and test sets have been created. Now we will run randomForest on the training data.

### The randomForest algorithm below includes target variable (medv) '.' which is all the predictors,

### the data is BostonTrain, number of trees to create using 'bagging' method is 100 and what makes the

### random forest algorithm set apart from boosting; number of predictors randomly chosen from full set of

### predictors is 5. In regression trees, the recommended number for mtry is the total number of predictors

### divided by three. In classification (an example to follow) the recommended number is the square root of

### predictors.

#Build the model:  
rf <- randomForest(medv~., data=train, mtry=6, importance = TRUE)  
rf

##   
## Call:  
## randomForest(formula = medv ~ ., data = train, mtry = 6, importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 6  
##   
## Mean of squared residuals: 11.21406  
## % Var explained: 86.21

#Now make predictions on the test set:  
yhat <- predict(rf, test)  
  
#The MSE:  
mean((yhat - test$medv)^2)

## [1] 16.23085

#Look at variable importance; looking at increase in MSE by looking at  
#mean decrease in accuracy in predictions on out  
#of bag samples, when the given variable is excluded from the model and  
#increase in node purity by looking at the total decrease in node purity   
#resulting from the given variable, averaged over all trees.  
importance(rf)

## %IncMSE IncNodePurity  
## crim 13.905168 1144.45272  
## zn 2.721400 72.83503  
## indus 8.463850 1213.97948  
## chas 3.270861 147.91585  
## nox 14.871973 1227.67252  
## rm 43.496630 11880.58675  
## age 10.555111 611.36956  
## dis 12.314255 1285.96179  
## rad 4.030756 128.15980  
## tax 9.742466 652.82292  
## ptratio 15.170913 1619.17704  
## black 6.569988 420.56903  
## lstat 25.927797 8111.12283

#Now plot it:  
varImpPlot(rf)

