Machine Learning PSET 3

Krish Suchak

Decision Trees

1. Setup

```r
set.seed(45)
survey <- read.csv('nes2008.csv')
feats <- names(survey)[-1]
p <- length(feats)
lambda <- seq(0.0001, 0.04, 0.001)
```

2. Train and Test Splits

```r
samples <- sample(nrow(survey), .75 * nrow(survey), replace = FALSE)
train <- survey[samples, ]
test <- survey[-samples, ]
```

3. Plot Train/Test MSEs after Boosting

```r
library(gbm)
```

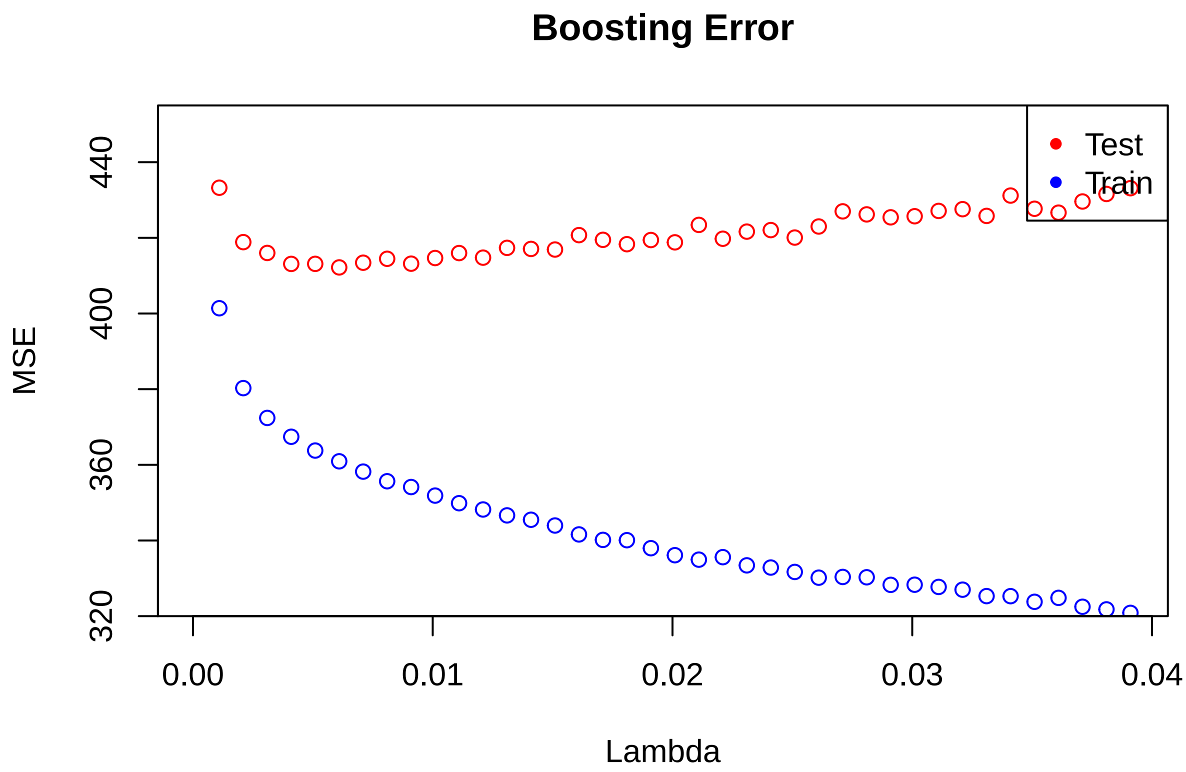
```
## Loaded gbm 2.1.5
```

```r
train_mse <- c()
test_mse <- c()

for (l in lambda) {
  boost <- gbm(biden ~ .,
              data = train,
              distribution = "gaussian",
              n.trees = 1000,
              shrinkage = l,
              interaction.depth = 4)
  train_preds <- predict(boost, newdata = train, n.trees = 1000)
  test_preds <- predict(boost, newdata = test, n.trees = 1000)
  train_mse <- append(train_mse, mean((train_preds - train$biden)^2))
  test_mse <- append(test_mse, mean((test_preds - test$biden)^2))

}
```

```r
plot(lambda, train_mse, col = 'blue',
     ylab = 'MSE', xlab = 'Lambda', main = 'Boosting Error', ylim = c(325, 450))
points(lambda, test_mse, col = 'red')
legend(x='topright', legend=c('Test', 'Train'), col=c('red', 'blue'), pch = 20)
```



4. MSE at lambda = 0.01

```
boost <- gbm(biden ~ .,
             data = train,
             distribution = "gaussian",
             n.trees = 1000,
             shrinkage = 0.01,
             interaction.depth = 4)
pred <- predict(boost, newdata = test, n.trees = 1000)
mse <- mean((pred - test$biden)^2)
mse
```

```
## [1] 416.5368
```

```
min(test_mse)
```

```
## [1] 412.1839
```

These values are quite comparable (test MSE at lambda = 0.01 and minimum test MSE value in boosting). La

5. Bagging

```
library(ipred)
bagg <- bagging(biden ~ .,
                data = train)
```

```r
pred <- predict(bagg, newdata = test)
mse = mean((pred - test$biden)^2)
mse
```

```
## [1] 418.6089
```

This test MSE is larger than the one from previous parts (less accurate model).

6. Random Forest

```r
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.
```

```r
rf <- randomForest(biden ~ ., data = train)
pred <- predict(rf, newdata = test)
mse <- mean((pred - test$biden)^2)
mse
```

```
## [1] 421.5164
```

This test MSE is similar to the one achieved through bagging.

7. Linear Regression

```r
model <- lm(biden ~ ., data = train)
pred <- predict(model, newdata = test)
mse <- mean((pred - test$biden)^2)
mse
```

```
## [1] 416.2959
```

8. Model Comparison

The linear regression yielded a test MSE smaller than the bagging and random forest models but larger than a boosted model with a low learning rate. Therefore, the boosted model is the best approach.

SVMs

1. Setup

```r
library(ISLR)
oj <- OJ
oj$Purchase <- as.factor(oj$Purchase)
set.seed(45)
samples <- sample(nrow(oj), 800, replace = FALSE)
train <- oj[samples,]
test <- oj[-samples,]
```

2. SVM Classifier

```r
library(e1071)
clf <- svm(Purchase ~ .,
           data = train,
           kernel = 'linear',
           cost = 0.01)
summary(clf)
```

```
##
## Call:
## svm(formula = Purchase ~ ., data = train, kernel = "linear", cost = 0.01)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.01
##
## Number of Support Vectors:  434
##
##  ( 216 218 )
##
##
## Number of Classes:  2
##
## Levels:
##  CH MM
```

The SVM model has 434 support vectors with 216 belonging to Citrus Hill and 211 belonging to Minute Maid.