Chapter 8 Tree Based Methods - Problems 7

Kartheek Raj

12/23/2019

7. In the lab, we applied random forests to the Boston data using mtry=6 and using ntree=25 and ntree=500. Create a plot displaying the test error resulting from random forests on this data set for a more comprehensive range of values for mtry and ntree. You can model your plot after Figure 8.10. Describe the results obtained.

Required packages : MASS,tree,randomForest and ISLR.

Answer

Boston data pulling from MASS package and data snapshots.

```
require(MASS)
```

```
## Loading required package: MASS
```

```
require(tree)
```

```
## Loading required package: tree
```

```
## Warning: package 'tree' was built under R version 3.6.2
```

```
require(ISLR)
```

```
## Loading required package: ISLR
```

```
require(randomForest)
```

```
## Loading required package: randomForest
```

```
## Warning: package 'randomForest' was built under R version 3.6.2
```

```
## randomForest 4.6-14
```

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

```
library(MASS)
library(tree)
library(ISLR)
library(randomForest)



boston<-data.frame(Boston)
head(boston)
```

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

```
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```
str(boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
summary(boston)
```

```
##       crim                zn              indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00
```

```
##  Max.   :37.97   Max.   :50.00
```

Test and train data splits

```
set.seed(1)
sd<-sample(1:nrow(boston), round(nrow(boston)/2))
train.x<-boston[sd,-14]
test.x<-boston[-sd,-14]
train.y<-boston[sd,14]
test.y<-boston[-sd,14]
```

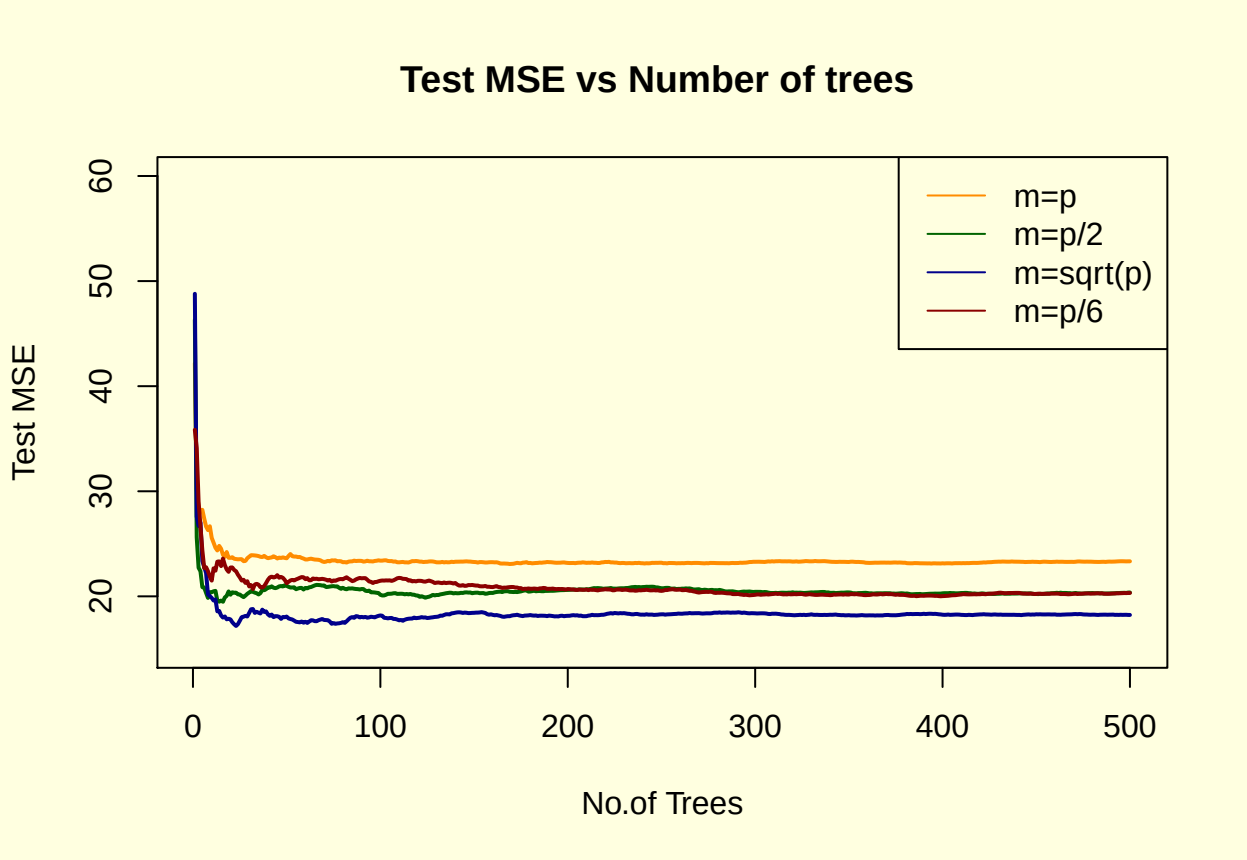
Building Randomforest Models with more **mtry** and **ntree**.

```
mtry1<-randomForest(train.x,train.y,test.x,test.y,mtry = round(ncol(boston)-1),ntree = 500)
mtry2<-randomForest(train.x,train.y,test.x,test.y,mtry = round(ncol(boston)/2),ntree = 500)
mtry3<-randomForest(train.x,train.y,test.x,test.y,mtry = round(sqrt(ncol(boston))),ntree = 500)
mtry4<-randomForest(train.x,train.y,test.x,test.y,mtry = round(ncol(boston)/6),ntree = 500)
```

Plotting the test errors above generate models.

```
par(mfrow=c(1,1),bg="lightyellow")
plot(1:500,mtry1$test$mse,col="darkorange",type = "l",lwd=2,main="Test MSE vs Number of trees ",ylim = c
lines(1:500,mtry2$test$mse,col="darkgreen",type = "l",lwd=2)
lines(1:500,mtry3$test$mse,col="darkblue",type = "l",lwd=2)
lines(1:500,mtry4$test$mse,col="darkred",type = "l",lwd=2)

legend("topright",c("m=p","m=p/2","m=sqrt(p)","m=p/6"),col=c("darkorange","darkgreen","darkblue","darkr
```



Insight 1 : All models droped Test MSE around the similar number of tress.M=sqrt(p) model yiels lowest

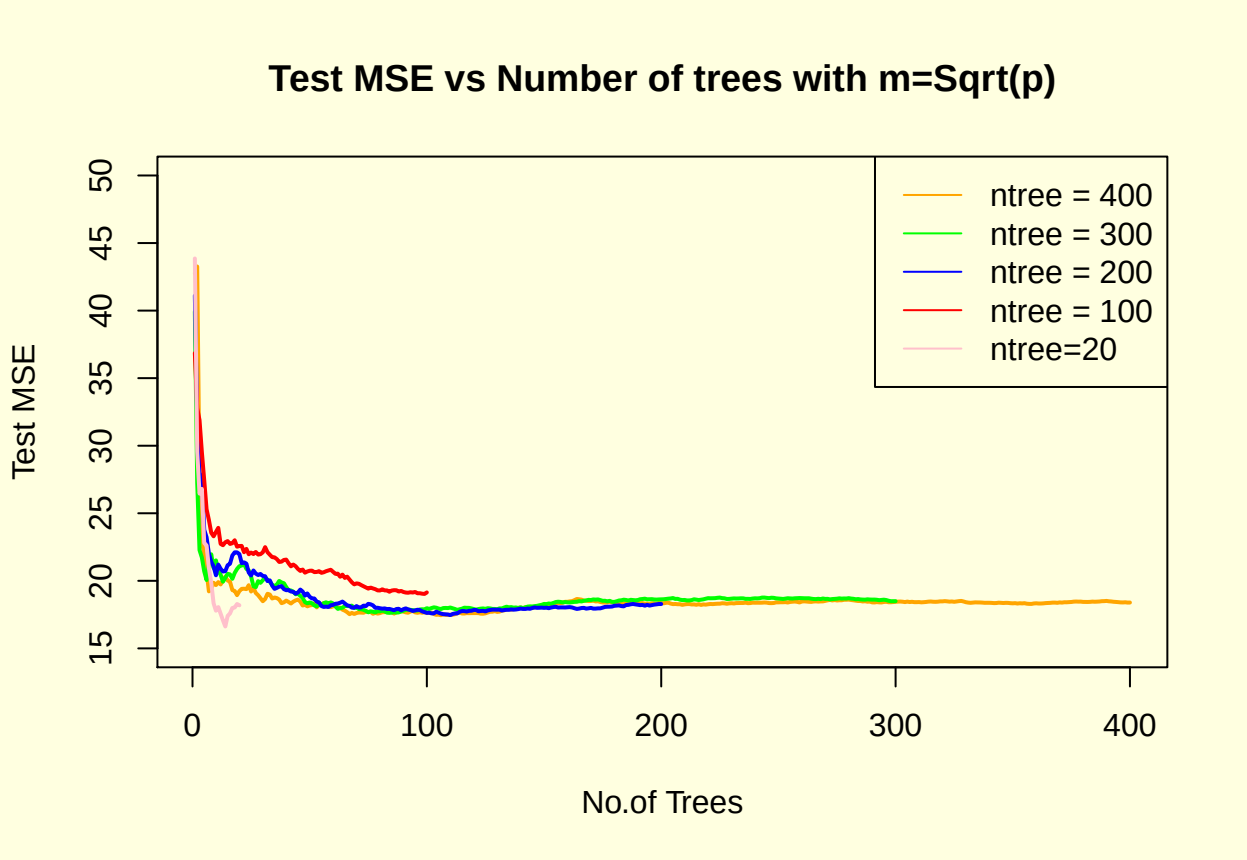
MSE

```
ntree1<-randomForest(train.x,train.y,test.x,test.y,mtry = round(sqrt(ncol(boston))),ntree = 100)
ntree2<-randomForest(train.x,train.y,test.x,test.y,mtry = round(sqrt(ncol(boston))),ntree = 200)
ntree3<-randomForest(train.x,train.y,test.x,test.y,mtry = round(sqrt(ncol(boston))),ntree = 300)
ntree4<-randomForest(train.x,train.y,test.x,test.y,mtry = round(sqrt(ncol(boston))),ntree = 400)
ntree5<-randomForest(train.x,train.y,test.x,test.y,mtry = round(sqrt(ncol(boston))),ntree = 20)
```

Plotting the test errors above generate models with m=sqrt(P)

```
par(mfrow=c(1,1),bg="lightyellow")
plot(1:400,ntree4$test$mse,col="orange",type = "l",lwd=2,main="Test MSE vs Number of trees with m=Sqrt(
lines(1:300,ntree3$test$mse,col="green",type = "l",lwd=2)
lines(1:200,ntree2$test$mse,col="blue",type = "l",lwd=2)
lines(1:100,ntree1$test$mse,col="red",type = "l",lwd=2)
lines(1:20,ntree5$test$mse,col="pink",type = "l",lwd=2)

legend("topright",c("ntree = 400","ntree = 300","ntree = 200","ntree = 100","ntree=20"),col=c("orange",
```



Insight2 : ntree=300 records lowest Test MSE then others. However,ntree=20 records proxmity to othe rest of the trees.After developing 9 models expermenting with no of trees and varaibles i,e m narrowed to model yields lowest MSE is m=sqrt(p) and 300 trees.