STAT542 Statistical Learning Homework 4

Huamin Zhang

Nov 14, 2017

Name: Huamin Zhang (huaminz2@illinois.edu)

Question 2

a) [20 points]

Answer:

To estimate the degree of freedom for each tree, we use the formula:

$$df(\hat{f}) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\mathrm{Cov}(\hat{y}_i, y_i).$$

To estimating $\mathrm{Cov}(\hat{y}_i, y_i)$, we fix X and do 20 times simulation.(Generate Y, fit the model, and predict $\hat{Y}$. Then use the sample covariance to estimate the degree of freedom.

```
library(MASS)
library(randomForest)
# Set the sedd, number of observation and dimension
set.seed(0); P = 20; N = 200
# Function generate_data: Generate the data, input the number of observation N,
# dimension P, and randam seed, return the data X and the response variable Y
# with standard normal errors.
generate_data<-function(N,P,seed_x,seed_y){
  I = diag(nrow = P)
  set.seed(seed_x); X = as.matrix(mvrnorm(N, mu=rep(0,P), Sigma=I))
  set.seed(seed_y); Y = 1 + 0.5 * (X[,1] + X[,2] + X[,3] + X[,4]) +  rnorm(N)
  return(list(X = X, Y = Y))
}
# N: The number of observation
# P: The dimension of the data
# mtry: A seq of mtry parameters to estimate degree of freedom
# nodesize: A seq of nodesize parameters to estimate degree of freedom
# iter: The number of simulations we will perform
# Output: result: A matrix, the row name is the nodesize, the column name
#         is the mtry, and the value is the estimation of Dof
DoF_RF_mtry_nodesize<-function(N,P,mtry,nodesize,iter){
  mtry_n = length(mtry); nodesize_n = length(nodesize)
  result = matrix(NA,nodesize_n,mtry_n)
  rownames(result) = nodesize; colnames(result) = mtry
  for(i in 1:nodesize_n){
```

```r
    for(j in 1:mtry_n){
      Y.pred = NULL; Y.ture = NULL
      for(m in 1:iter){
        data = generate_data(N,P,0,m); X = data$X; Y = data$Y
        rf.fit = randomForest(X, Y, mtry = mtry[j], nodesize = nodesize[i])
        Y.ture = cbind(Y.ture,Y); Y.pred = cbind(Y.pred, predict(rf.fit, X))
      }
      # Calculate the degree of freedom
      result[i,j] = sum(sapply(1:N, function(x) cov(Y.ture[x,],Y.pred[x,])))
    }
  }
  return(result)
}
mtry = seq(1,19,3); nodesize =c(seq(3,30,3),50,100)
```

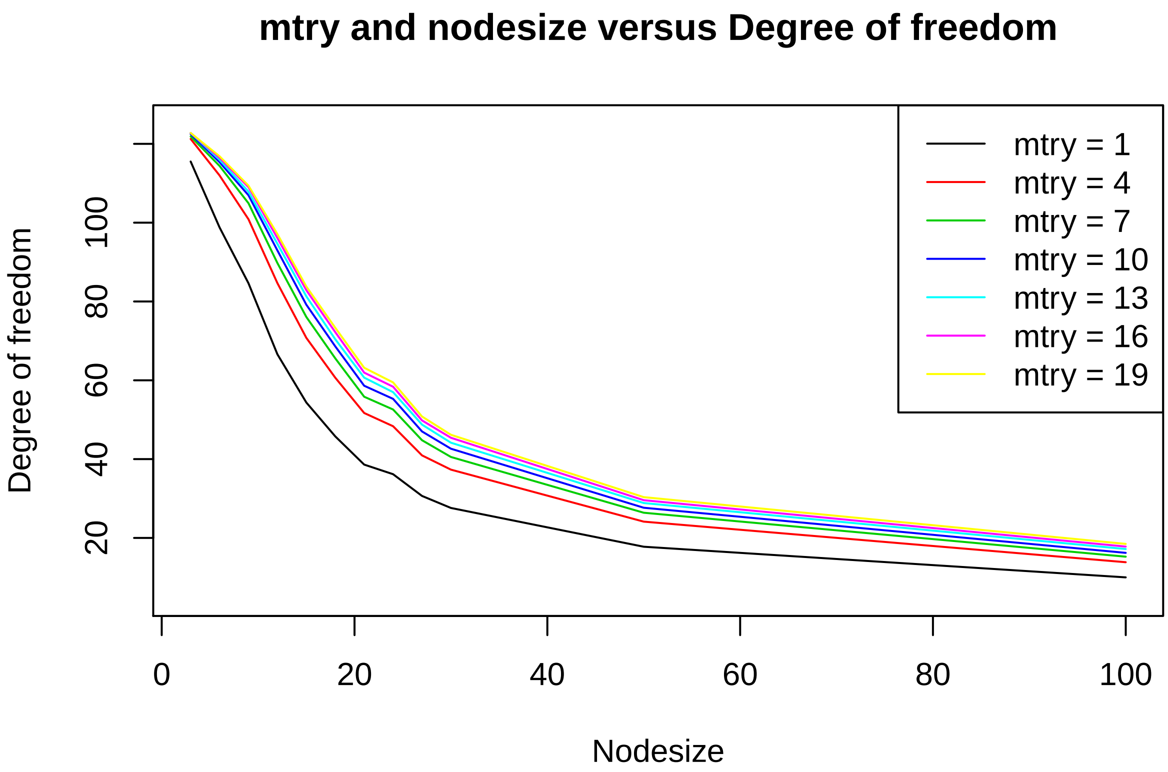
```r
mtry_nodesize_result = DoF_RF_mtry_nodesize(N,P,mtry,nodesize,20)
```

```r
load("Q2.Rdata")
mtry_nodesize_result
```

```
##              1         4         7        10        13        16        19
## 3   115.51934 121.25021 121.88871 122.50217 122.77360 122.76103 122.75918
## 6    98.79494 111.98541 114.39870 115.34318 115.96799 116.46944 116.84956
## 9    84.61932 100.86155 104.93131 106.94828 107.86491 109.00974 109.39100
## 12   66.60074  84.65600  89.75726  92.95979  94.64055  96.10129  97.10857
## 15   54.36219  70.76812  76.07349  79.18427  81.30993  82.87040  83.79552
## 18   45.77555  60.67018  65.57971  68.61897  70.42426  72.22953  73.34406
## 21   38.60715  51.70024  55.83294  58.61812  60.67903  61.94778  63.17651
## 24   36.17059  48.34493  52.58692  55.28993  56.99306  58.37114  59.46388
## 27   30.65929  40.94482  44.79441  47.02051  48.75816  49.81399  50.80682
## 30   27.60301  37.33243  40.54553  42.64159  44.16493  45.41675  46.18183
## 50   17.76732  24.13320  26.39450  27.65740  28.85019  29.57759  30.35121
## 100   9.98839  13.83514  15.24167  16.20943  17.17651  17.80109  18.48164
```

In the matrix, the row name is the nodesize, the column name is the mtry, and the value is the estimation of DOF. According to the matrix, we make a plot to summary the relation between Degree of freedom and mtry, nodesize. **We find that when the nodesize parameter increases, the DOF of Random Forest decreses. And when the mty parameter increases, the DOF of Random Forest increases.**

```r
plot(x = NULL, y= NULL,xlim=c(3,100),ylim=c(5,125),xlab = "Nodesize",
     ylab = "Degree of freedom")
for(i in 1:dim(mtry_nodesize_result)[2]){
  lines(rownames(mtry_nodesize_result),mtry_nodesize_result[,i],type='l', col = i)
}
legend("topright", c("mtry = 1","mtry = 4","mtry = 7","mtry = 10", "mtry = 13",
       "mtry = 16","mtry = 19"), col = 1:7, cex = 1, lty = 1)
title(main = "mtry and nodesize versus Degree of freedom")
```



b) [15 points]

Answer:

To estimate the variance of this estimator, we use the formula:

$$\frac{1}{n}\sum_{i}^{n} E_{\hat{f}}(\hat{f}(x_i) - E[\hat{f}(x_i)])^2$$

```
# N: The number of observation
# P: The dimension of the data
# ntree: A seq of ntree parameters to estimate degree of freedom
# iter: The number of simulations we will perform
# Output: result: A matrix of the ntree parameters and the corresponding
#          degree of freedom
Var_RF_ntree<-function(N,P,ntree,iter){
  ntree_n = length(ntree); var = rep(NA,ntree_n)
  for(i in 1:ntree_n){
    Y.pred = NULL; Y.ture = NULL
    for(m in 1:iter){
      data = generate_data(N,P,0,0)
      X = data$X; Y = data$Y; set.seed(m)
      rf.fit = randomForest(X, Y, ntree = ntree[i])
      Y.ture = cbind(Y.ture,Y); Y.pred = cbind(Y.pred, predict(rf.fit, X))
```

```
    }
    # Calculte the variance
    var[i] = sum(sapply(1:N, function(x) mean((Y.pred[x,] - mean(Y.pred[x,]))^2))) / N
  }
  result = rbind(ntree,var)
  return(result)
}
```

```
ntree = c(5,10,50,100,200,500,1000,2000,3000,4000)
ntree_result = Var_RF_ntree(N,P,ntree,20)
```

```
ntree_result
```

```
##              [,1]        [,2]         [,3]         [,4]         [,5]
## ntree 5.00000000 10.00000000 50.000000000 1.000000e+02 2.000000e+02
## var   0.09791961  0.04977433  0.009915496 5.199607e-03 2.547287e-03
##               [,6]         [,7]         [,8]         [,9]        [,10]
## ntree 5.000000e+02 1.000000e+03 2.000000e+03 3.000000e+03 4.000000e+03
## var   1.050329e-03 5.177913e-04 2.545992e-04 1.690471e-04 1.235088e-04
```

According to the matrix, we make a plot to summary the relation between the variance of this estimator and ntree. **We find that when the ntree parameter increases, the variance of this estimator decreases. We can shrink the estimator's variance using ntree parameter.**

```
plot(x = ntree_result[1,][3:10], y= ntree_result[2,][3:10],xlab = "ntree",
     ylab = "Variance of RF estimator",col = "red", type = 'l')
points(ntree_result[1,][3:10], ntree_result[2,][3:10])
title(main = "ntree versus Variance of RF estimator")
```