STAT 542 / CS 598: Homework 4

Fall 2019, by Paul Nel (paulnel2)

Due: Monday, Oct 14 by 11:59 PM Pacific Time

Table of Contents

## Question 1 [70 Points] Tuning Random Forests in Virtual Twins

### Set up of data sets

set.seed(101)  
require(randomForest)  
mydata = read.csv('Sepsis.csv')  
n = nrow(mydata)  
train.id = sample(1:n, round(n\*0.75,0))  
train.data = mydata[train.id,-15]  
test.data = mydata[-train.id,-15]

### Prediction and comparison functions

get\_prediiction <- function(therapy, mtry, nodesize){  
 model.data = train.data[train.data[,"THERAPY"] == therapy,]  
 rforest = randomForest(model.data$Health ~ .-THERAPY , data=model.data, mtry=mtry, nodesize=nodesize)  
 return (predict(rforest, test.data[,-2]))  
}  
  
get\_treatment <- function(test\_sample, mtry, nodesize){  
 pred.0 = get\_prediiction(0, mtry, nodesize)  
 pred.1 = get\_prediiction(1, mtry, nodesize)  
 return (ifelse(pred.0>pred.1, 0, 1))  
}  
  
get\_accuracy <- function(prediction, truth) {  
 mean(prediction == truth)  
}

### Iteration loop

reps = 2  
mtries = c(2,5,13)  
nodesizes = c(1,10,100)  
accuracies.ave = matrix(rep(0, length(mtries)\*length(nodesizes)),ncol=length(mtries))  
for (i in 1:reps){  
 accuracies = matrix(rep(0, length(mtries)\*length(nodesizes)),ncol=length(mtries))  
 for (j in 1:length(mtries)){  
 for (k in 1:length(nodesizes)){  
 prediction = get\_treatment(test.data, mtry=mtries[j], nodesize=nodesizes[k])  
 accuracies[k,j] = get\_accuracy(prediction, mydata[-train.id, "BEST"])  
 }  
 }  
 accuracies.ave = accuracies.ave + accuracies  
}  
accuracies.ave = accuracies.ave/reps

This leads to the following results with rows representing different nodesize and the coloumns different mtry values.

Model performance for combinations of mtry and nodesize

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | 5 | 13 |
| 1 | 0.6864407 | 0.7076271 | 0.7161017 |
| 10 | 0.7033898 | 0.7076271 | 0.7203390 |
| 100 | 0.6864407 | 0.7330508 | 0.8093220 |

### Intuition

Normally when one has highly correlated data, it helps to reduce the number of predictors (mtry) available for selection at each split. When all predictors are used (13 in this case if we ignore THERAPY) this process is similar to simple bagging. The result of the above analysis would suggest that the predictors are largely uncorrelated since the accuracy improves as we allow more predictors into the selection at each split, up to all 13.

The parameter nodesize determins the minimum size of nodes, in other words it determines the depth of each tree. For small values we can expect very deep trees. In this example I vary this from a minimum of 1 (largest possible tree) to 100. This parameter has a significant impact for larger values of mtry suggesting that smaller trees results in better aggregation of related samples.

## Question 2 [30 Points] Second Step in Virtual Twins

The second step in a virtual twins model is to use a single tree model (CART) to describe the choice of the best treatment. Perform the following: \* Based on your optimal tuning parameter, fit the Virtual Twins model described in Question 1. Again, you should not use the BEST variable. \* For each subject, obtain the predicted best treatment of the training data itself \* Treating the label of best treatment as the outcome, and fit a single tree model to predict it. Be careful which variables should be removed from this model fitting. \* Consider tuning the tree model using the cost-complexity tuning.