Statistical Machine Learning

Shivani Batus 19203249

14/04/2020

### Problem:

Use multinomial logistic regression and random forests to predict the classification of the images. In doing so, compare appropriately the two models. Evaluate the generalization performance of the best model on the test data. Write a report about your findings, discussing in particular the following questions/tasks: • Which of the two models performs better? • Comment on the predictive performance of the best classifier.

### Solution:

We begin with loading the necessary libraries and the dataset. The aim is to predict the classification of a satellite image, given the multi-spectral values. Prepare the data running the following lines of code.The data matrix dat will be employed for training and model comparison (validation), while the data matrix dat\_test will be employed for testing. The last column is the response variable.

library(mlbench)

## Warning: package 'mlbench' was built under R version 3.6.3

library(nnet)  
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.6.3

## randomForest 4.6-14

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

library(mclust)

## Warning: package 'mclust' was built under R version 3.6.2

## Package 'mclust' version 5.4.5  
## Type 'citation("mclust")' for citing this R package in publications.

data("Satellite")  
# this will re-order alphabetically class labels and remove spacing  
Satellite$classes <- gsub(" ", "\_", Satellite$classes)  
Satellite$classes <- factor( as.character(Satellite$classes) )  
# to have the same initial split  
set.seed(777222)  
D <- nrow(Satellite)  
keep <- sample(1:D, 5500)  
test <- setdiff(1:D, keep)  
dat <- Satellite[keep,]  
dat\_test <- Satellite[test,]  
colnames(dat)

## [1] "x.1" "x.2" "x.3" "x.4" "x.5" "x.6" "x.7"   
## [8] "x.8" "x.9" "x.10" "x.11" "x.12" "x.13" "x.14"   
## [15] "x.15" "x.16" "x.17" "x.18" "x.19" "x.20" "x.21"   
## [22] "x.22" "x.23" "x.24" "x.25" "x.26" "x.27" "x.28"   
## [29] "x.29" "x.30" "x.31" "x.32" "x.33" "x.34" "x.35"   
## [36] "x.36" "classes"

we will consider a 5-fold cross validation. Note that setting K = 5 correspond to a 80%/20% split between training and test set at each step of the cross validation procedure. Because at each istance we randomly split the data into the folds, we replicate the cross validation procedure 100 times to obtain a better estimate of the accuracy of the two models. We create a validation dataset from the training set.

N=nrow(dat) #No. of observations.  
K=5 #No. of folds  
R=50 #No.of replicates.  
out=vector("list",R) # store accuracy output.  
  
# out is a list, each slot of this list will contain a matrix where each column.  
# corresponds to the accuracy of each classifier in the K fold.  
  
best = matrix(NA, R, K) # store best classifier.  
  
for(r in 1:R){  
   
 #print(r) #iteration number.  
   
 acc= matrix(NA,K,2) # accuracy of the two classifiers. in the K fold  
   
 folds = rep( 1:K, ceiling(N/K) )   
 folds = sample(folds) # random permute.   
 folds = folds[1:N]  
   
 for ( k in 1:K ) {   
 train = which(folds != k)   
 val = setdiff(1:N, train)  
   
 #Multinomial Regression.  
 fit1=multinom(classes~.,data=dat,subset=train,maxit=300,trace=F)  
 #Random Forest Classification.  
 fit2=randomForest(classes~.,data=dat,subset=train)  
   
 #predict the classification of the test data observations in the dropped fold.  
 pred1=predict(fit1,type="class",newdata=dat[val,]) #Multinomial Regression.  
 tab1=table(dat$classes[val],pred1)  
 acc[k,1]=sum(diag(tab1))/sum(tab1)  
   
 pred2=predict(fit2,type="class",newdata=dat[val,]) #Random Forest classification.  
 tab2=table(dat$classes[val],pred2)  
 acc[k,2]=sum(diag(tab2))/sum(tab2)  
   
 best[r,k]=ifelse(acc[k,1]>acc[k,2],"Multinomial","Random Forest")  
 }  
 out[[r]] = acc   
}

The object out is a list where the slots are the replications of the cross validation procedure. Each slot contains the accuracy of the two classifiers across the folds.The first column contains the accuracy of the classification tree in the 5 folds and the second column the accuracy of the logistic regression.

We can calculate the average fold accuracy for classification tree and logistic regression model in all replications.

avg <-as.data.frame( t( sapply(out, colMeans) ))   
colnames(avg)<-c("Multinomial","Random Forest")  
head(avg)

## Multinomial Random Forest  
## 1 0.8623636 0.9147273  
## 2 0.8601818 0.9156364  
## 3 0.8592727 0.9123636  
## 4 0.8572727 0.9152727  
## 5 0.8627273 0.9185455  
## 6 0.8614545 0.9154545

We can calculate the mean classification accuracy on all the replications and produce a plot to visually compare the estimated accuracy of the two classifiers.

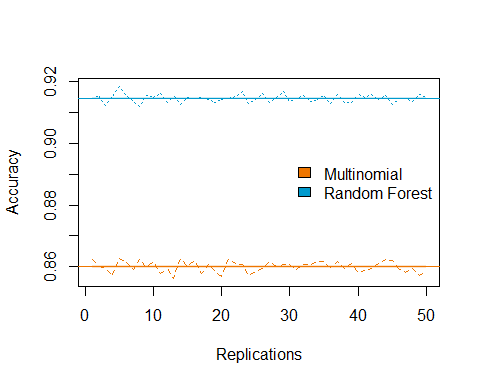
meanAcc <- colMeans(avg) #Estimated mean accuracy.  
meanAcc

## Multinomial Random Forest   
## 0.8600109 0.9146873

sdAcc <- apply(avg, 2, sd)/sqrt(R)  
sdAcc

## Multinomial Random Forest   
## 0.0002410642 0.0001843453

matplot(avg, type = "l", lty = c(2,3), col = c("darkorange2", "deepskyblue3"), xlab = "Replications", ylab = "Accuracy")  
  
bounds1 <- rep( c(meanAcc[1] - 2\*sdAcc[1], meanAcc[1] + 2\*sdAcc[1]), each = R )  
bounds2 <- rep( c(meanAcc[2] - 2\*sdAcc[2], meanAcc[2] + 2\*sdAcc[2]), each = R )   
  
polygon(c(1:R, R:1), bounds1, col = adjustcolor("darkorange2", 0.2), border = FALSE)   
polygon(c(1:R, R:1), bounds2, col = adjustcolor("deepskyblue3", 0.2), border = FALSE)  
  
abline(h = meanAcc, col = c("darkorange2", "deepskyblue3"))   
  
legend("right", fill = c("darkorange2", "deepskyblue3"), legend = c("Multinomial", "Random Forest"), bty = "n")



The random forest classifier has a higher average performance accuracy and near about the same variability as compared to multinomial logistic regression.

The object best contains the number of times each of the two classifier was selected as the best one in all the replications and folds. We can calculate the overall proportion of times each of the two classifiers was chosen as the best one.

prop.table(table(best))

## best  
## Random Forest   
## 1

pred = predict(fit2, newdata=dat\_test)  
tab=table(pred,dat\_test$classes)  
tab

##   
## pred cotton\_crop damp\_grey\_soil grey\_soil red\_soil  
## cotton\_crop 96 0 0 0  
## damp\_grey\_soil 0 62 6 0  
## grey\_soil 1 18 172 2  
## red\_soil 0 2 2 223  
## vegetation\_stubble 0 0 0 0  
## very\_damp\_grey\_soil 0 18 1 0  
##   
## pred vegetation\_stubble very\_damp\_grey\_soil  
## cotton\_crop 0 0  
## damp\_grey\_soil 0 13  
## grey\_soil 0 4  
## red\_soil 4 0  
## vegetation\_stubble 85 5  
## very\_damp\_grey\_soil 10 211

sum(diag(tab))/sum(tab)

## [1] 0.9080214