Predicting the Classification of the Satellite Images

Introduction

In this report, random forests and multinomial logistic regression are used on the dataset Satellite available in package mlbench contains 6435 images displaying different scenes recorded by the Landsat satellite program. Each row of the data correspond to an image, which encodes a scene in terms of 36 features extracted from multi-spectral values of pixels in 3x3 neighborhoods of the image.

The task is to predict the classification of the satellite images. In doing so, both models are compared and based on the generalization performance evaluation, we can find the best model.

Here is the detail of both methods and predictive performance for the data.

1. **Data Preparation**

```
library(mlbench);library(randomForest) # load the libraries
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
require(foreign)
## Loading required package: foreign
require(nnet)
## Loading required package: nnet

# load the data
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,]      #training and validation
dat_test = Satellite[test,] #testing
N=nrow(dat) # store number of observations
K=5 # set number of folds
R=50 # set the replicates
out = vector("list",R) # store accuracy output
best = matrix(NA, R, K) # store best classifier
```

2. **Fitting Both Models**

```
for(r in 1:R)
{
  acc= matrix(NA,K,2) # accuracy of the two classifiers in the K folds
  folds = rep( 1:K, ceiling(N/K) )
  folds = sample(folds) # random permute
  folds = folds[1:N] # ensure we got N data points
    for ( k in 1:K ) {
    train = which(folds != k) # train data
    val = setdiff(1:N, train) # validation data

    # fitting the random forest model on the training data
    fit1=randomForest(classes~.,data=dat,subset=train)

    # fitting the multinomial logistic regression on the training data
    fit2=multinom(classes~.,data=dat,subset=train)

    # accuracy for the random forest model
    pred1=predict(fit1,type="class",newdata=dat[val,])
    tab1=table(dat$classes[val],pred1)
    acc[k,1]=sum(diag(tab1))/sum(tab1)

    # accuracy for the multinomial logistic regression model
    pred2=predict(fit2,type="class",newdata=dat[val,])
    tab2=table(dat$classes[val],pred2)
    acc[k,2]=sum(diag(tab2))/sum(tab2)

    # best model having higher accuracy
    best[r,k]=ifelse(acc[k,1]>acc[k,2],"Random Forest","MLR")

    }
  out[[r]] = acc
}
```

Here is the average fold accuracy for random forest and multinomial logistic regression model in all replications. The random forest has average around 91.4% while the multinomial logistic regression has a lower accuracy around 80.6%.

```
avg_fold = t(sapply(out, colMeans))
head(avg_fold)
```

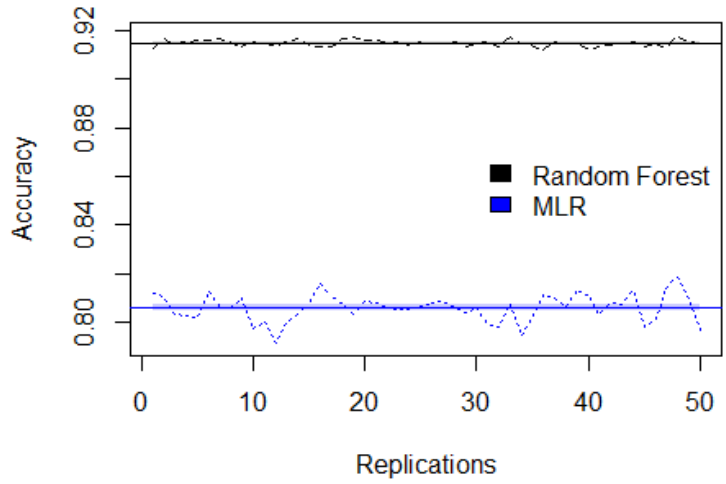
```
##           [,1]      [,2]
## [1,] 0.9123636 0.8120000
## [2,] 0.9169091 0.8110909
## [3,] 0.9138182 0.8034545
## [4,] 0.9154545 0.8029091
## [5,] 0.9158182 0.8020000
## [6,] 0.9160000 0.8125455
```

```
estmean = colMeans(avg_fold);estmean
## [1] 0.9146073 0.8061455
```

And, the standard deviation of mean accuracy for all replications are good, below 1%.

```
sd = apply(avg_fold,2,sd)/sqrt(R)
sd
## [1] 0.0001930211 0.0007861510 # estimated mean accuracy

# plot the mean accuracies over 50 replications
matplot(avg_fold, type = "l", lty = c(2,3), xlab = "Replications", ylab = "Ac
curacy",col = c("black", "blue"))
bounds1 <- rep( c(estmean[1] - 2*sd[1], estmean[1] + 2*sd[1]), each = R )
bounds2 <- rep( c(estmean[2] - 2*sd[2], estmean[2] + 2*sd[2]), each = R )
polygon(c(1:R, R:1), bounds1, col = adjustcolor("black", 0.2), border = FALSE
)
polygon(c(1:R, R:1), bounds2, col = adjustcolor("blue", 0.2), border = FALSE)
abline(h = estmean, col = c("black", "blue"))
legend("right", fill = c("black", "blue"), legend = c("Random Forest", "MLR")
, bty = "n")
```



Overall, the random forest model has a better average accuracy and a lower variability in the estimates compared to the multinomial logistic regression. In every comparison, we store the best model in variable best. In every replication, **Random forest is the best model.**

```
prop.table(table(best))

## best
## Random Forest
##             1
```

Predictive performance

After applying the best model to the dat_test, it shows that the predictive performance of the Random Forest is very good with 91% accuracy. We can use it for the better prediction of the satellite images classification.

```
# accuracy of the best model on the unseen test data
bestpred = predict(fit1,type="class",newdata=dat_test)
besttab=table(dat_test$classes,bestpred)
besttab
```

```
##                       bestpred
##                        cotton_crop damp_grey_soil grey_soil red_soil
##   cotton_crop                   95              0         2        0
##   damp_grey_soil                 1             61        16        2
##   grey_soil                      0              6       173        1
##   red_soil                       0              0         2      222
##   vegetation_stubble             0              1         0        3
##   very_damp_grey_soil            0              7         5        0
##                       bestpred
##                        vegetation_stubble very_damp_grey_soil
##   cotton_crop                           0                   0
##   damp_grey_soil                        0                  20
##   grey_soil                             0                   1
##   red_soil                              1                   0
##   vegetation_stubble                   86                   9
##   very_damp_grey_soil                   4                 217
```

```
sum(diag(besttab))/sum(besttab)
```

```
## [1] 0.913369
```

The code is adapted from:
Statistical Machine Learning – Practical Lab 7 (with additional modification).