Team14\_Project\_Perceptron

Prajwal Gonnade; Supreet Nayak

December 9, 2016

library(boot)

## Warning: package 'boot' was built under R version 3.2.5

library(caret)

## Warning: package 'caret' was built under R version 3.2.5

## Warning: package 'ggplot2' was built under R version 3.2.5

library(class)  
library(ROCR)

## Warning: package 'ROCR' was built under R version 3.2.5

## Warning: package 'gplots' was built under R version 3.2.5

library(MASS)  
library(monmlp)

## Warning: package 'monmlp' was built under R version 3.2.5

Occupancy\_Train <- read.csv(file.choose(),header=T)  
Occupancy\_Test1 <- read.csv(file.choose(),header=T)  
Occupancy\_Test2 <- read.csv(file.choose(),header=T)  
Occupancy\_subset <- Occupancy\_Train[,c(2,4,5,6)]  
Occupancy\_subset <- data.matrix(Occupancy\_subset)  
Occupancy\_response <- Occupancy\_Train[,7]  
Occupancy\_response <- data.matrix(Occupancy\_response)

# Perceptron

# Fit the model and compute the predictions  
Occupancy.monmlp <- monmlp.fit(Occupancy\_subset, Occupancy\_response, hidden1=3, n.ensemble=15, monotone=1, bag=TRUE)

## \*\* Ensemble 1   
## \*\* Bagging on  
## 1 0.05689021   
## \*\* 0.05689021   
##   
## \*\* Ensemble 2   
## \*\* Bagging on  
## 1 0.06621595   
## \*\* 0.06621595   
##   
## \*\* Ensemble 3   
## \*\* Bagging on  
## 1 0.0762087   
## \*\* 0.0762087   
##   
## \*\* Ensemble 4   
## \*\* Bagging on  
## 1 0.0714974   
## \*\* 0.0714974   
##   
## \*\* Ensemble 5   
## \*\* Bagging on  
## 1 0.07462134   
## \*\* 0.07462134   
##   
## \*\* Ensemble 6   
## \*\* Bagging on  
## 1 0.07064936   
## \*\* 0.07064936   
##   
## \*\* Ensemble 7   
## \*\* Bagging on  
## 1 0.05917637   
## \*\* 0.05917637   
##   
## \*\* Ensemble 8   
## \*\* Bagging on  
## 1 0.0565633   
## \*\* 0.0565633   
##   
## \*\* Ensemble 9   
## \*\* Bagging on  
## 1 0.05639147   
## \*\* 0.05639147   
##   
## \*\* Ensemble 10   
## \*\* Bagging on  
## 1 0.0667447   
## \*\* 0.0667447   
##   
## \*\* Ensemble 11   
## \*\* Bagging on  
## 1 0.05977946   
## \*\* 0.05977946   
##   
## \*\* Ensemble 12   
## \*\* Bagging on  
## 1 0.06211387   
## \*\* 0.06211387   
##   
## \*\* Ensemble 13   
## \*\* Bagging on  
## 1 0.06509851   
## \*\* 0.06509851   
##   
## \*\* Ensemble 14   
## \*\* Bagging on  
## 1 0.07589035   
## \*\* 0.07589035   
##   
## \*\* Ensemble 15   
## \*\* Bagging on  
## 1 0.05706769   
## \*\* 0.05706769

monmlp.test1 <- monmlp.predict(x = data.matrix(Occupancy\_Test1[,c(2,4,5,6)]), weights = Occupancy.monmlp)  
mean(round(monmlp.test1) != Occupancy\_Test1$Occupancy)

## [1] 0.02138837

confusionMatrix(Occupancy\_Test1$Occupancy, round(monmlp.test1))

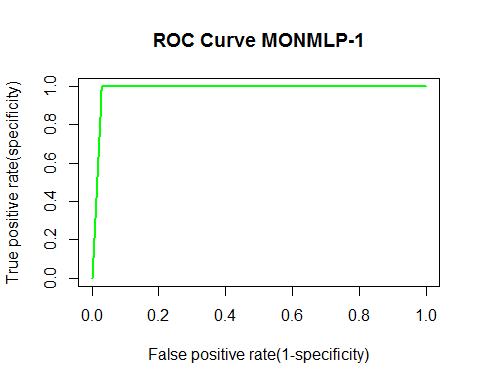
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1638 55  
## 1 2 970  
##   
## Accuracy : 0.9786   
## 95% CI : (0.9724, 0.9838)  
## No Information Rate : 0.6154   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9544   
## Mcnemar's Test P-Value : 5.675e-12   
##   
## Sensitivity : 0.9988   
## Specificity : 0.9463   
## Pos Pred Value : 0.9675   
## Neg Pred Value : 0.9979   
## Prevalence : 0.6154   
## Detection Rate : 0.6146   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.9726   
##   
## 'Positive' Class : 0   
##

# Compute the AUC  
roc.curve=function(s,print=FALSE){  
Ps=(monmlp.test1>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 0))/sum(Occupancy\_Test1$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 1))/sum(Occupancy\_Test1$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test1$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 1638 55  
## 1 2 970

## FPR TPR   
## 0.03248671 0.99794239

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,], xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve MONMLP-1', col="green",lwd=2,type="l")



monmlp.test2 <- monmlp.predict(x = data.matrix(Occupancy\_Test2[,c(2,4,5,6)]), weights = Occupancy.monmlp)  
mean(round(monmlp.test2) != Occupancy\_Test2$Occupancy)

## [1] 0.01066448

confusionMatrix(Occupancy\_Test2$Occupancy, round(monmlp.test2))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7609 94  
## 1 10 2039  
##   
## Accuracy : 0.9893   
## 95% CI : (0.9871, 0.9913)  
## No Information Rate : 0.7813   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9683   
## Mcnemar's Test P-Value : 3.992e-16   
##   
## Sensitivity : 0.9987   
## Specificity : 0.9559   
## Pos Pred Value : 0.9878   
## Neg Pred Value : 0.9951   
## Prevalence : 0.7813   
## Detection Rate : 0.7803   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.9773   
##   
## 'Positive' Class : 0   
##

# Compute the AUC  
roc.curve=function(s,print=FALSE){  
Ps=(monmlp.test2>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 0))/sum(Occupancy\_Test2$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 1))/sum(Occupancy\_Test2$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test2$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 7609 94  
## 1 10 2039

## FPR TPR   
## 0.01220304 0.99511957

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,], xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve MONMLP-2', col="green",lwd=2,type="l")

