Lab 9.6.3

Team 17

3/22/2021

Lab 9.6.3 ROC Curves

Receiver Operator Characteristic (ROC) curves are a graphical plot showing the diagnostic ability of binary classifiers. The objective of this lab is to show how to plot and analyze ROC curves with different gamma values.

The first step is generating data with a non-linear class boundary. "Y" is a factor with 2 levels, '1' and '2'.

```
set.seed(1)
x <- matrix(rnorm(200*2), ncol = 2)
x[1:100, ] <- x[1:100, ] + 2
x[101:150, ] <- x[101:150, ] - 2
y <- c(rep(1, 150), rep(2, 50))
dat <- data.frame(x = x, y = as.factor(y))
summary(dat)
```

```
##       x.1                x.2          y
##  Min.   :-3.9144   Min.   :-4.2649   1:150
##  1st Qu.:-1.0789   1st Qu.:-0.9259   2: 50
##  Median : 0.9699   Median : 0.7546
##  Mean   : 0.5355   Mean   : 0.5406
##  3rd Qu.: 2.1849   3rd Qu.: 1.9965
##  Max.   : 4.4016   Max.   : 4.6492
```

```
plot(x, col = y, pch = 20)
```

Plotting the data shows that the class boundary is non-linear. Next, the data is split into a training and a testing set.

```
train.index <- sample(1:nrow(dat),nrow(dat)*.5)
train <- dat[train.index,]
test <- dat[-train.index,]
```

ROC (Receiver Operating Characteristic) Curve

Load the "ROCR" and "e1071" libraries.

```
library(ROCR)
library(e1071)# Library that contains svm function
```

Next, a function is created to generate an ROC plot. The function inputs are “pred” and “truth”.

“Pred” is a vector containing the numerical score for each observation.

“Truth” is a vector containing the class label for each observation.

“Predobject” creates the prediction object for evaluation using ROCR. Label.ordering specifies the direction, which determines how the negativity or positivity of an observation is determined. “2” is the negative class label and “1” is the positive class label.

“Perf” evaluates performance using tpr (true positive rate) and fpr (false positive rate).

```
rocplot =function(pred , truth , ...){
  predob = prediction(pred , truth, label.ordering = c(2,1))
  perf = performance(predob , "tpr", "fpr")
  plot(perf ,...)}
```

Next, we will obtain fitted values, which are numerical scores used to obtain class labels.

In this exercise, we will be comparing models with a gamma set at 2 and another set at 50. The model with gamma set at 2 will be called “svmfit.opt” and the one with 50 will be called “svmfit.flex”.

```
svmfit.opt <- svm(y~., data = train, kernel = 'radial', gamma = 2, cost = 1,
decision.values = TRUE)
fitted.g2.train <- attributes(predict(svmfit.opt, train,
decision.values=TRUE))$decision.values

svmfit.flex=svm(y~., data=train,kernel
="radial",gamma=50,cost=1,decision.values =TRUE)
fitted.g50.train=attributes(predict(svmfit.flex,train,decision.values=TRUE))$
decision.values
head(fitted.g50.train)

##            1/2
## 148  1.0004240
## 192 -0.5934886
## 87   0.9998178
## 20   1.0132808
## 112  1.0000652
## 177 -0.9998983
```

The sign of the fitted value determines on which side of the decision boundary the observation lies. If the fitted value exceeds zero then the observation is assigned to one class, and if it is less than zero then it is assigned to the other.

Next, we produce the ROC plot.

```
par(mfrow = c(1, 1))
par(pty = "s")       # plot as a square

rocplot(fitted.g2.train,train$y,main="Train Data", col = "blue")
rocplot(fitted.g50.train,train$y,add=TRUE,col="red ")         # add = TRUE
means add to existing graph

legend(.4, .3, legend=c("Gamma = 2", "Gamma = 50"),col=c("blue", "red"),
lty=1:1, cex=0.8)
```

The training model with a gamma of 50 produced more accurate results. We know this because the red line hugs the upper left corner of the graph, meaning that the true positives are maximized.

Test Data

Next, we will fit the values for the test set and plot the ROC curves.

```
fitted.g2.test=attributes(predict(svmfit.opt,test,decision.values=TRUE))$deci
sion.values
fitted.g50.test=attributes(predict(svmfit.flex,test,decision.values=TRUE))$de
cision.values

par(pty = "s")
rocplot(fitted.g2.test,test$y,main="Test Data", col = "blue")
rocplot(fitted.g50.test,test$y,add=T,col="red")

legend(.5, .5, legend=c("Gamma = 2", "Gamma = 50"),col=c("blue", "red"),
lty=1:1, cex=0.8)
```

The model where gamma = 2 produced the best results.

Another ROC plotting method

Another way to plot the ROC curves is using the “pROC” library. The inputs are the same as the “rocplot” function inputs. However, one difference is adding “direction”. With direction = “>”, each observation will be considered positive if it is less than 0 and negative otherwise. If you change the direction, you reverse the positive and negative predictions and therefore reverse the ROC curve.

The plots have Area Under the Curve (AUC) labels, which help evaluate the model performance. The best models will have the highest AUC score.

```
library(pROC)

par(mfrow = c(1, 2))
par(pty = "s")

### Train Data
plot(roc(train$y,fitted.g2.train,direction = ">"), col = "blue",
     legacy.axes= TRUE, lwd = 3, main = "Train Data", print.auc=TRUE,
#print.auc=TRUE: prints AUC label
     xlab = "False Positive Rate", ylab= "True Positive Rate")

plot(roc(train$y,fitted.g50.train,direction = ">"), col = "red", add = TRUE,
     lwd = 3, print.auc = TRUE, print.auc.x = 0.5, print.auc.y = 0.4)
#print.auc.x/y: coordinates for label

legend(.5, .2, legend=c("Gamma = 2", "Gamma = 50"),col=c("blue", "red"),
lty=1:1, cex=0.65)

### Test Data
plot(roc(test$y,fitted.g2.test,direction = ">"), col = "blue",
     legacy.axes= TRUE, lwd = 3, main = "Test Data", print.auc=TRUE,
     xlab = "False Positive Rate", ylab= "True Positive Rate")

plot(roc(test$y,fitted.g50.test,direction = ">"), col = "red", add = TRUE,
     lwd = 3, print.auc = TRUE, print.auc.x = 0.5, print.auc.y = 0.4)

legend(.5, .2, legend=c("Gamma = 2", "Gamma = 50"),col=c("blue", "red"),
lty=1:1, cex=0.65)
```

As we saw earlier, a gamma of 50 works best for the train data and a gamma of 2 works best for the test data.