

## Lab 9.6.3

Team 17

3/22/2021

### Lab 9.6.3 ROC Curves

First, we generate data with a non-linear class boundary.

```
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))
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 set.

```
train = sample(200, 100)
```

### ROC (Receiver Operating Characteristic) Curve

Load the ROCR library

```
library(ROCR)
## Warning: package 'ROCR' was built under R version 4.0.4
library(e1071)# Library that contains svm function
```

Create a function to generate an ROC plot

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

The function inputs are: pred : a vector containing the numerical score for each observation  
truth: a vector containing the class label for each observation  
Predobject creates the prediction object for evaluation using ROCR  
Perf evaluates performance using tpr (true positive rate) and fpr (false positive rate)

Next, we obtain fitted values from SVM using 'decision.values = T'.

```

svmfit.opt <- svm(y~., data = dat[train,], kernel = 'radial', gamma = 2, cost = 1, decision.values = T)

fitted <- attributes(predict(svmfit.opt, dat[train,], decision.values = TRUE))$decision.values
head(fitted)

##           1/2
## 148  1.130044
## 192 -1.038648
## 87   1.188065
## 20   1.204742
## 112  1.195899
## 177 -1.096820

```

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, 2))
```

If we increase gamma we can produce a more flexible fit and get further improvements in accuracy

```

svmfit.flex=svm(y~., data = dat[train, ], kernel = "radial", gamma = 50, cost = 1, decision.values = T)
fitted = attributes(predict(svmfit.flex, dat[train, ], decision.values = T))$decision.values
rocplot(fitted, dat[train, "y"], main = "Training Data")
rocplot(fitted, dat[train, "y"], add = T, col = "red ")

```

## Test data

```

fitted = attributes(predict(svmfit.opt, dat[-train, ], decision.values = T))$decision.values
rocplot(fitted, dat[-train, "y"], main = "Test Data")

fitted = attributes(predict(svmfit.flex, dat[-train, ], decision.values = T))$decision.values
rocplot(fitted, dat[-train, "y"], add = T, col = "red")

```

The model where gamma = 2 produces the most accurate results