Project – Machine Learning with R

1. Recreate the R part of this experiment using your computer.

library("e1071", lib.loc="~/anaconda3/lib/R/library")

> setwd("~/Documents/RProjects")

> srd1 = read.csv("srd1.csv", header=FALSE)

> srd2 = read.csv("srd2.csv", header=FALSE)

> y1 = rep(0,1000)

> y2 = rep(1,1000)

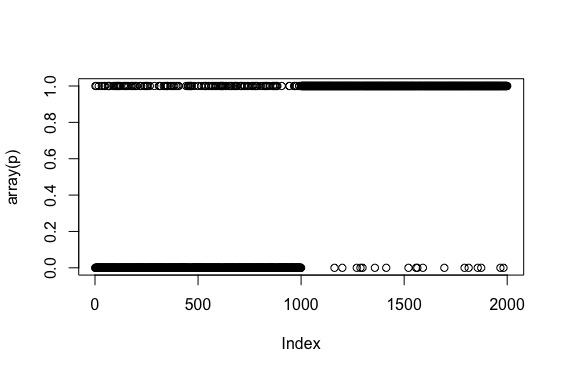
> x = rbind(srd1,srd2)

> y = factor(c(y1,y2))

> model = svm(x,y)

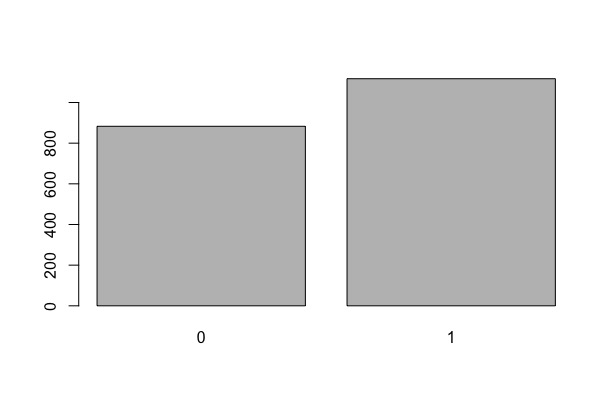
> p = predict(model,x)

> plot(array(p))



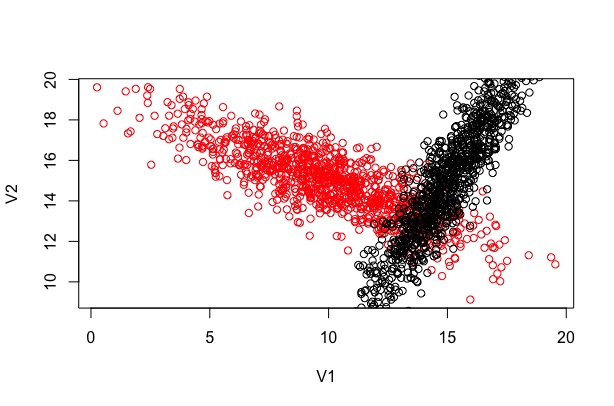
1. See if you can find other ways to display the predictions.

> plot(p)



> f = plot(srd1, col="red")

> points(srd2, col="black")



> install.packages("scatterplot3d")

trying URL 'https://cran.rstudio.com/src/contrib/scatterplot3d\_0.3-41.tar.gz'

Content type 'application/x-gzip' length 460912 bytes (450 KB)

==================================================

downloaded 450 KB

\* installing \*source\* package ‘scatterplot3d’ ...

\*\* package ‘scatterplot3d’ successfully unpacked and MD5 sums checked

\*\* R

\*\* inst

\*\* preparing package for lazy loading

\*\* help

\*\*\* installing help indices

\*\* building package indices

\*\* installing vignettes

\*\* testing if installed package can be loaded

\* DONE (scatterplot3d)

The downloaded source packages are in

‘/private/var/folders/yz/f\_tg879j23j29b6dllvby5380000gq/T/Rtmp3KqPSg/downloaded\_packages’

Updating HTML index of packages in '.Library'

Making 'packages.html' ... done

> library(scatterplot3d)

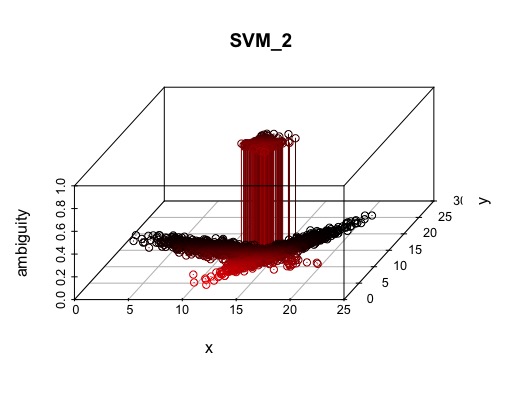
> i = as.matrix(as.numeric(predict(model,x[1:1000,]))) -1

> j = as.matrix(as.numeric(predict(model,x[1001:2000,]))) -2

> j = abs(j)

> z = c(I,j)

> scatterplot3d(x[,1],x[,2],z,pch=20, highlight.3d = TRUE, type="h", main ="3D View", xlab="x", ylab="y", zlab="ambiguity", angle=65)



1. Train the svm model using the 1000 length data and then predict the 10,000 length data. Compare the predictions with the data. See the ideas of cross validation.

> rd1 = read.csv("rd1.csv", header=FALSE)

> rd2 = read.csv("rd2.csv", header=FALSE)

> y1 = rep(0,10000)

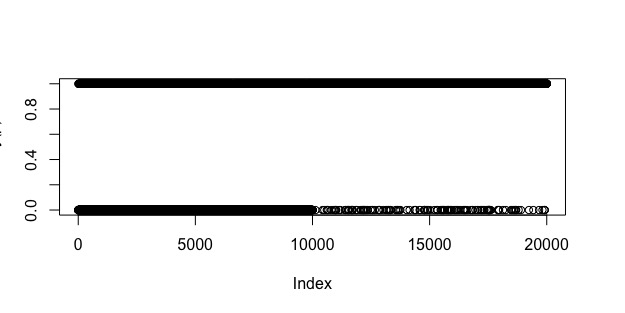
> y2 = rep(1,10000)

> y = factor(c(y1,y2))

> x = rbind(rd1,rd2)

> p = predict(model,x)

> plot(array(p))



The prediction compared to the data, shows significant errors. In particular, the first 10,000 predictions appear to have been miss-predicted more visibly than the later 10,000 predictions.

1. Is there anyway to remove the ambiguity between the two classes given the existing data? What about if you could add additional measurements for each data point?

Answer: The ambiguity here can be minimized by considering more variables. The code below attempts to introduce a third set of variable for each observation that can help the svm function do a more accurate—if not a perfect—prediction in this case.

> srd1 = read.csv("srd1.csv", header=FALSE)

> srd2 = read.csv("srd2.csv", header=FALSE)

> y1 = rep(0,1000)

> y2 = rep(1,1000)

> z = c(y1,y2) ##This line concatenates y1 and y2 as numbers

> y = factor(z) ## This line changes z as two level factor

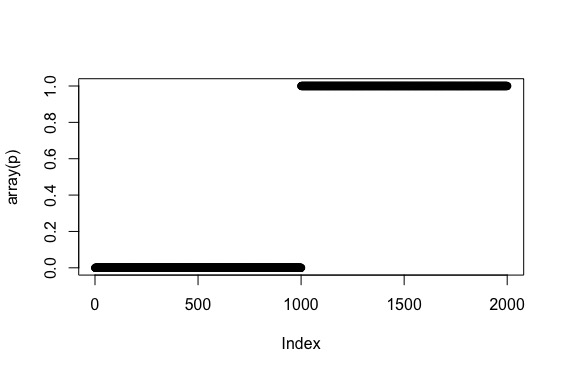
> x = rbind(srd1,srd2)

> x = cbind(x,z)

> model = svm(x,y)

> p = predict(model,x)

> plot(array(p))



The same can be applied to the data set with more variable and predict more accurately as well.

> rd1 = read.csv("rd1.csv", header=FALSE)

> rd2 = read.csv("rd2.csv", header=FALSE)

> x = rbind(rd1,rd2)

> y1 = rep(0,10000)

> y2 = rep(1,10000)

> z = c(y1,y2)

> y = factor(z)

> x = cbind(x,z)

> p = predict(model,x)

> plot(array(p))

