Shetty\_S\_M8\_SVM

Sooraj Shetty

March 20, 2017

library(kernlab)  
library(e1071)  
library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:kernlab':  
##   
## alpha

library(openxlsx)

Loading my data - Abalone Data Set

data\_url <- 'http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data'  
ABA <- read.csv(url(data\_url),sep=",",header = FALSE)  
colnames(ABA) <- c("Sex","Length","Diameter","Height","WholeWeight","ShuckedWeight","VisceraWeight","ShellWeight","Rings")  
head(ABA)

## Sex Length Diameter Height WholeWeight ShuckedWeight VisceraWeight  
## 1 M 0.455 0.365 0.095 0.5140 0.2245 0.1010  
## 2 M 0.350 0.265 0.090 0.2255 0.0995 0.0485  
## 3 F 0.530 0.420 0.135 0.6770 0.2565 0.1415  
## 4 M 0.440 0.365 0.125 0.5160 0.2155 0.1140  
## 5 I 0.330 0.255 0.080 0.2050 0.0895 0.0395  
## 6 I 0.425 0.300 0.095 0.3515 0.1410 0.0775  
## ShellWeight Rings  
## 1 0.150 15  
## 2 0.070 7  
## 3 0.210 9  
## 4 0.155 10  
## 5 0.055 7  
## 6 0.120 8

summary(ABA)

## Sex Length Diameter Height   
## F:1307 Min. :0.075 Min. :0.0550 Min. :0.0000   
## I:1342 1st Qu.:0.450 1st Qu.:0.3500 1st Qu.:0.1150   
## M:1528 Median :0.545 Median :0.4250 Median :0.1400   
## Mean :0.524 Mean :0.4079 Mean :0.1395   
## 3rd Qu.:0.615 3rd Qu.:0.4800 3rd Qu.:0.1650   
## Max. :0.815 Max. :0.6500 Max. :1.1300   
## WholeWeight ShuckedWeight VisceraWeight ShellWeight   
## Min. :0.0020 Min. :0.0010 Min. :0.0005 Min. :0.0015   
## 1st Qu.:0.4415 1st Qu.:0.1860 1st Qu.:0.0935 1st Qu.:0.1300   
## Median :0.7995 Median :0.3360 Median :0.1710 Median :0.2340   
## Mean :0.8287 Mean :0.3594 Mean :0.1806 Mean :0.2388   
## 3rd Qu.:1.1530 3rd Qu.:0.5020 3rd Qu.:0.2530 3rd Qu.:0.3290   
## Max. :2.8255 Max. :1.4880 Max. :0.7600 Max. :1.0050   
## Rings   
## Min. : 1.000   
## 1st Qu.: 8.000   
## Median : 9.000   
## Mean : 9.934   
## 3rd Qu.:11.000   
## Max. :29.000

ABAsvm <- ABA[c("Sex","Length","Diameter","Height","WholeWeight","ShuckedWeight","VisceraWeight","ShellWeight","Rings")]  
ABAsvm$Sex <- as.factor(ABAsvm$Sex)  
head(ABAsvm)

## Sex Length Diameter Height WholeWeight ShuckedWeight VisceraWeight  
## 1 M 0.455 0.365 0.095 0.5140 0.2245 0.1010  
## 2 M 0.350 0.265 0.090 0.2255 0.0995 0.0485  
## 3 F 0.530 0.420 0.135 0.6770 0.2565 0.1415  
## 4 M 0.440 0.365 0.125 0.5160 0.2155 0.1140  
## 5 I 0.330 0.255 0.080 0.2050 0.0895 0.0395  
## 6 I 0.425 0.300 0.095 0.3515 0.1410 0.0775  
## ShellWeight Rings  
## 1 0.150 15  
## 2 0.070 7  
## 3 0.210 9  
## 4 0.155 10  
## 5 0.055 7  
## 6 0.120 8

str(ABAsvm)

## 'data.frame': 4177 obs. of 9 variables:  
## $ Sex : Factor w/ 3 levels "F","I","M": 3 3 1 3 2 2 1 1 3 1 ...  
## $ Length : num 0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475 0.55 ...  
## $ Diameter : num 0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37 0.44 ...  
## $ Height : num 0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...  
## $ WholeWeight : num 0.514 0.226 0.677 0.516 0.205 ...  
## $ ShuckedWeight: num 0.2245 0.0995 0.2565 0.2155 0.0895 ...  
## $ VisceraWeight: num 0.101 0.0485 0.1415 0.114 0.0395 ...  
## $ ShellWeight : num 0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165 0.32 ...  
## $ Rings : int 15 7 9 10 7 8 20 16 9 19 ...

Divide Training Data into x (containing all features) and y (containing classes)

x <- ABAsvm[c("Sex","Length","Diameter","Height","WholeWeight","ShuckedWeight","VisceraWeight","ShellWeight","Rings")]  
y <- ABAsvm$Sex

Creating SVM Models and summary of each model

svm\_model\_rad <- svm(ABAsvm$Sex ~ ., data=ABAsvm, kernel="radial")  
summary(svm\_model\_rad)

##   
## Call:  
## svm(formula = ABAsvm$Sex ~ ., data = ABAsvm, kernel = "radial")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.125   
##   
## Number of Support Vectors: 3400  
##   
## ( 1451 1285 664 )  
##   
##   
## Number of Classes: 3   
##   
## Levels:   
## F I M

svm\_model\_lin <- svm(ABAsvm$Sex ~ ., data=ABAsvm, kernel="linear")  
summary(svm\_model\_lin)

##   
## Call:  
## svm(formula = ABAsvm$Sex ~ ., data = ABAsvm, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.125   
##   
## Number of Support Vectors: 3463  
##   
## ( 1467 1299 697 )  
##   
##   
## Number of Classes: 3   
##   
## Levels:   
## F I M

svm\_model\_sig <- svm(ABAsvm$Sex ~ ., data=ABAsvm, kernel="sigmoid")  
summary(svm\_model\_sig)

##   
## Call:  
## svm(formula = ABAsvm$Sex ~ ., data = ABAsvm, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.125   
## coef.0: 0   
##   
## Number of Support Vectors: 2262  
##   
## ( 792 1001 469 )  
##   
##   
## Number of Classes: 3   
##   
## Levels:   
## F I M

svm\_model\_pol <- svm(ABAsvm$Sex ~ ., data=ABAsvm, kernel="polynomial")  
summary(svm\_model\_pol)

##   
## Call:  
## svm(formula = ABAsvm$Sex ~ ., data = ABAsvm, kernel = "polynomial")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: polynomial   
## cost: 1   
## degree: 3   
## gamma: 0.125   
## coef.0: 0   
##   
## Number of Support Vectors: 3622  
##   
## ( 1470 1297 855 )  
##   
##   
## Number of Classes: 3   
##   
## Levels:   
## F I M

See the confusion matrix result of prediction, using command table to compare the result of SVM prediction and the class data in y variable.

predR <- predict(svm\_model\_rad,x)  
table(predR,y)

## y  
## predR F I M  
## F 354 34 240  
## I 180 1075 308  
## M 773 233 980

a1 <- predR == y  
table(a1)

## a1  
## FALSE TRUE   
## 1768 2409

predL <- predict(svm\_model\_lin,x)  
table(predL,y)

## y  
## predL F I M  
## F 239 16 193  
## I 225 1123 365  
## M 843 203 970

a2 <- predL == y  
table(a2)

## a2  
## FALSE TRUE   
## 1845 2332

predS <- predict(svm\_model\_sig,x)  
table(predS,y)

## y  
## predS F I M  
## F 322 37 291  
## I 335 878 463  
## M 650 427 774

a3 <- predS == y  
table(a3)

## a3  
## FALSE TRUE   
## 2203 1974

predP <- predict(svm\_model\_pol,x)  
table(predP,y)

## y  
## predP F I M  
## F 50 2 26  
## I 73 773 146  
## M 1184 567 1356

a4 <- predP == y  
table(a4)

## a4  
## FALSE TRUE   
## 1998 2179

We can see from all the above confusion matrices that svm\_model\_rad is showing most correct predictions, so we will further tune svm\_model\_rad to find best cost and gamma

Tuning SVM to find the best cost and gamma.

# I am facing the following error when I run this section of the code  
  
#NAs introduced by coercion  
#Error in if (any(co)) { : missing value where TRUE/FALSE needed  
  
# Could not rectify this chunk of the code.  
  
svm\_tune <- tune(svm, train.x=x, train.y=y, kernel="radial", ranges=list(cost=c(0.01,0.1,1,2,5,10), gamma=c(0.1,0.2,0.5,1,1.5,2)))  
print(svm\_tune)

You can create svm model again and try to run again

svm\_model\_after\_tune <- svm(ABAsvm$Sex ~ ., data=ABAsvm, kernel="radial", cost=1, gamma=0.125)  
  
summary(svm\_model\_after\_tune)

##   
## Call:  
## svm(formula = ABAsvm$Sex ~ ., data = ABAsvm, kernel = "radial",   
## cost = 1, gamma = 0.125)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.125   
##   
## Number of Support Vectors: 3400  
##   
## ( 1451 1285 664 )  
##   
##   
## Number of Classes: 3   
##   
## Levels:   
## F I M

Run Prediction again with new model See the confusion matrix result of prediction, using command table to compare the result of SVM prediction and the class data in y variable.

pred <- predict(svm\_model\_after\_tune,x)  
table(pred,y)

## y  
## pred F I M  
## F 354 34 240  
## I 180 1075 308  
## M 773 233 980

a5 <- pred == y  
table(a5)

## a5  
## FALSE TRUE   
## 1768 2409