**ASSIGNMENT 6**

**Name: Jain Samkitkumar Hasmukhlal**

**Roll No.: 20 (BEIT)**

> n<-150

> p<-2

> sigma<-1

> meanpos<-0

> meanneg<-3

> npos<-round(n/2)

> nneg<-n-npos

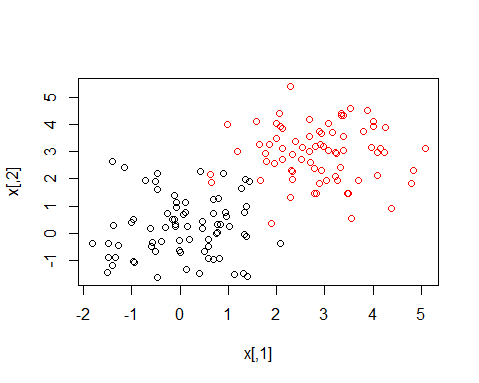
> xpos<-matrix(rnorm(npos\*p,mean=meanpos,sd=sigma), npos,p)

> xneg<- matrix(rnorm(nneg\*p,mean=meanneg,sd=sigma), npos,p)

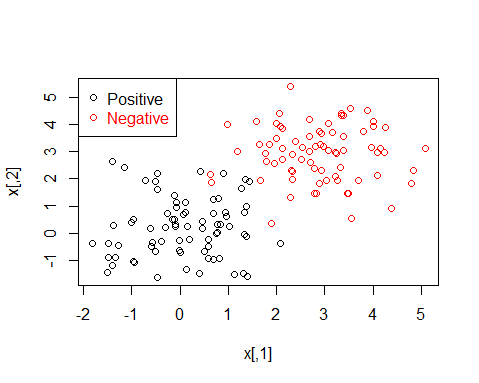
> x <- rbind(xpos,xneg)

> y <- matrix(c(rep(1,npos),rep(-1,nneg)))

> plot(x,col=ifelse(y>0,1,2))



> legend("topleft",c('Positive','Negative'),col=seq(2),pch=1,text.col=seq(2))



> ntrain<-round(n\*0.8)

> tindex<-sample(n,ntrain)

> xtrain <- x[tindex,]

> xtest <- x[tindex,]

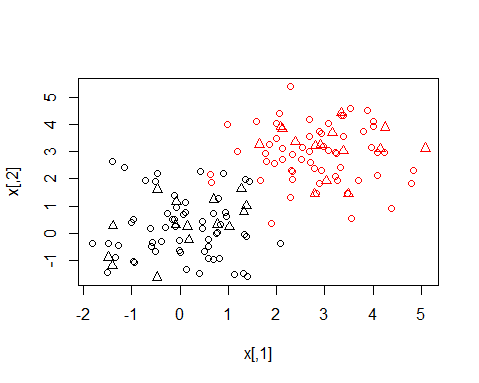
> ytrain <- y[tindex]

> ytest <- y[tindex]

> istrain=rep(0,n)

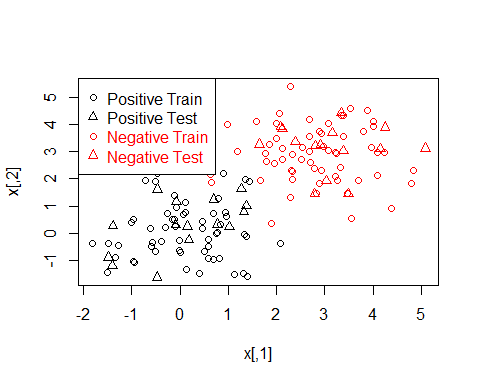
> istrain[tindex]=1

> plot(x,col=ifelse(y>0,1,2),pch=ifelse(istrain==1,1,2))



> legend("topleft",c('Positive Train','Positive Test','Negative Train','Negative Test'),

+ col=c(1,1,2,2),pch=c(1,2,1,2),text.col=c(1,1,2,2))



> install.packages("C:/Users/Solaris lap/Downloads/New folder/kernlab\_0.9-22.zip", repos = NULL, type = "win.binary")

Installing package into ‘C:/Users/Solaris lap/Documents/R/win-library/3.2’

(as ‘lib’ is unspecified)

package ‘kernlab’ successfully unpacked and MD5 sums checked

> library(kernlab)

> svp<-ksvm(xtrain,ytrain,type="C-svc",kernel='vanilladot',C=100,scaled=c())

Setting default kernel parameters

> svp

Support Vector Machine object of class "ksvm"

SV type: C-svc (classification)

parameter : cost C = 100

Linear (vanilla) kernel function.

Number of Support Vectors : 3

Objective Function Value : -33.1733

Training error : 0

> attributes(svp)

$param

$param$C

[1] 100

$scaling

`\001NULL\001`

$coef

$coef[[1]]

[1] -33.165111 32.060974 1.104137

$alphaindex

$alphaindex[[1]]

[1] 63 82 106

$b

[1] -14.05152

$obj

[1] -33.17334

$SVindex

[1] 63 82 106

$nSV

[1] 3

$prior

$prior[[1]]

$prior[[1]]$prior1

[1] 60

$prior[[1]]$prior0

[1] 60

$prob.model

$prob.model[[1]]

NULL

$alpha

$alpha[[1]]

[1] 33.165111 32.060974 1.104137

$type

[1] "C-svc"

$kernelf

function (x, y = NULL)

{

if (!is(x, "vector"))

stop("x must be a vector")

if (!is(y, "vector") && !is.null(y))

stop("y must be a vector")

if (is(x, "vector") && is.null(y)) {

crossprod(x)

}

if (is(x, "vector") && is(y, "vector")) {

if (!length(x) == length(y))

stop("number of dimension must be the same on both data points")

crossprod(x, y)

}

}

<environment: 0x0000000005e172b8>

attr(,"kpar")

list()

attr(,"class")

[1] "vanillakernel"

attr(,"class")attr(,"package")

[1] "kernlab"

$kpar

list()

$xmatrix

$xmatrix[[1]]

X1 X2

63 1.8351521 1.2484765

82 1.8111008 0.9931633

106 -0.2475164 1.8312248

$ymatrix

[1] -1 1 1 1 -1 -1 -1 1 -1 -1 -1 1 -1 -1 1 1 1 -1 -1 1 -1 1 1 -1 -1 1 -1 -1 -1 -1 -1 -1

[33] 1 1 -1 1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 1 -1 1 1 -1 -1 -1 1 -1 -1 -1 1 1 -1 -1 1

[65] -1 1 1 1 1 1 -1 -1 1 1 -1 1 1 1 1 -1 1 1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 1 -1 -1 -1

[97] 1 -1 1 -1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 1 -1 -1 1 1 -1 -1 1

$fitted

[1] -1 1 1 1 -1 -1 -1 1 -1 -1 -1 1 -1 -1 1 1 1 -1 -1 1 -1 1 1 -1 -1 1 -1 -1 -1 -1 -1 -1

[33] 1 1 -1 1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 1 -1 1 1 -1 -1 -1 1 -1 -1 -1 1 1 -1 -1 1

[65] -1 1 1 1 1 1 -1 -1 1 1 -1 1 1 1 1 -1 1 1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 1 -1 -1 -1

[97] 1 -1 1 -1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 1 -1 -1 1 1 -1 -1 1

$lev

[1] -1 1

$nclass

[1] 2

$error

[1] 0

$cross

[1] -1

$n.action

function (object, ...)

UseMethod("na.omit")

<bytecode: 0x000000000bd8a3d8>

<environment: namespace:stats>

$terms

`\001NULL\001`

$kcall

.local(x = x, y = ..1, scaled = ..5, type = "C-svc", kernel = "vanilladot",

C = 100)

$class

[1] "ksvm"

attr(,"package")

[1] "kernlab"

> alpha(svp)

[[1]]

[1] 33.165111 32.060974 1.104137

> alphaindex(svp)

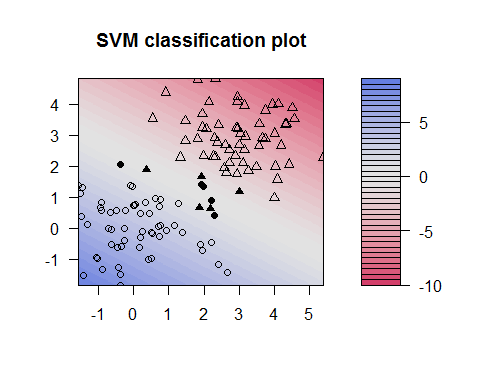
[[1]]

[1] 63 82 106

> b(svp)

[1] -14.05152

> plot(svp,data=xtrain)



> ypred = predict(svp,xtest)

> table(ytest,ypred)

ypred

ytest -1 1

-1 60 0

1 0 60

> sum(ypred==ytest)/length(ytest)

[1] 1

> ypredscore = predict(svp,xtest,type="decision")

> table(ypredscore > 0,ypred)

ypred

-1 1

FALSE 15 0

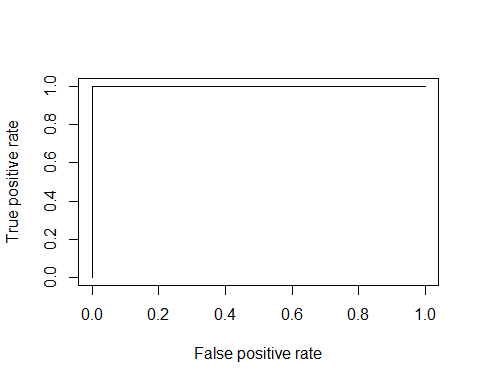
TRUE 0 15

> library(ROCR)

> pred <- prediction(ypredscore,ytest)

> perf <- performance(pred, measure = "tpr", x.measure ="fpr")

> plot(perf)



> perf <- performance(pred, measure = "prec", x.measure ="rec")

> plot(perf)

