

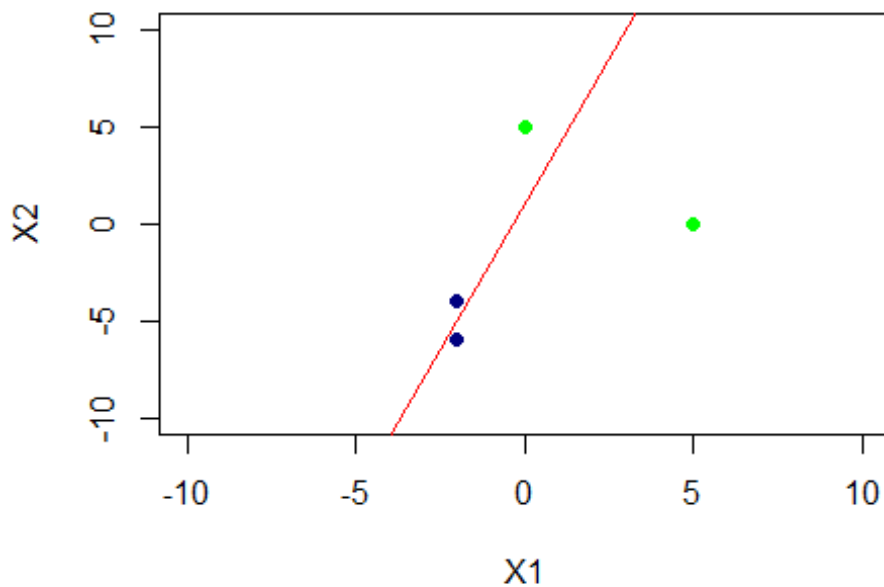
# Support Vector Machines

Vinit Sanjay Patil

2025-09-29

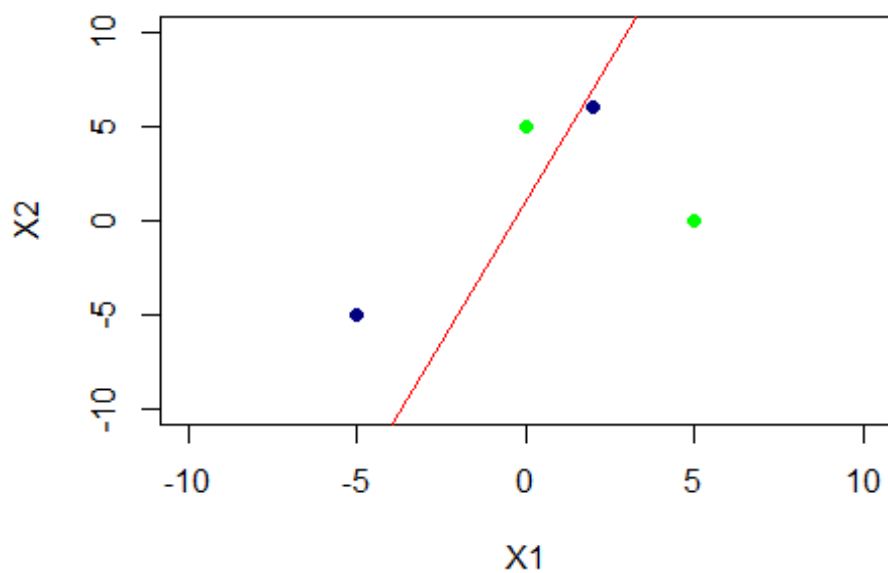
```
plot(0,type="n",xlab="X1",ylab="X2", ylim=c(-10,10),xlim=c(-10,10))
abline(1,3,col = 'red',lwd = 1.5) #line  $1+3X_1-X_2=0$ 
```

```
#points where line > 0 and line < 0
points(-2,-4,col = 'navyblue',pch = 19)
points(-2,-6,col = 'navyblue',pch = 19)
points(0,5,col = 'green',pch = 19)
points(5,0,col = 'green',pch = 19)
```



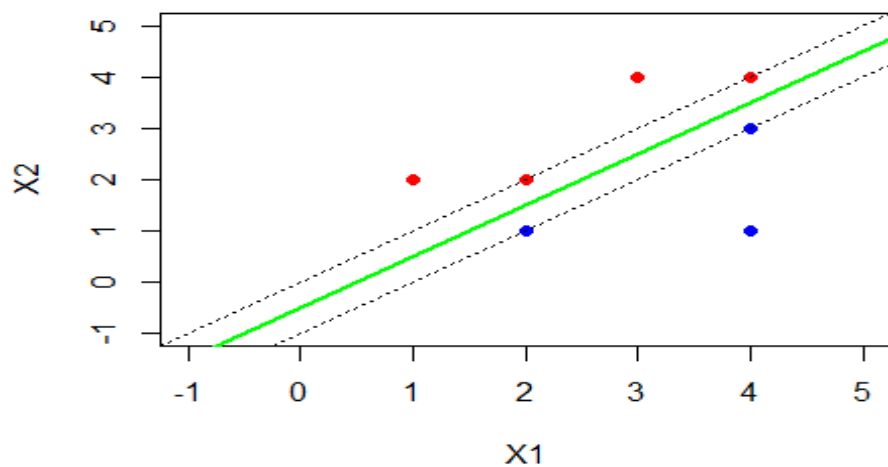
```
#b)
plot(0,type="n",xlab="X1",ylab="X2", ylim=c(-10,10),xlim=c(-10,10))
abline(1,3,col = 'red',lwd = 1.5) #line  $1+3X_1-X_2=0$ 
```

```
#points where line > 0 and line < 0
points(-5,-5,col = 'navyblue',pch = 19)
points(2,6,col = 'navyblue',pch = 19)
points(0,5,col = 'green',pch = 19)
points(5,0,col = 'green',pch = 19)
```

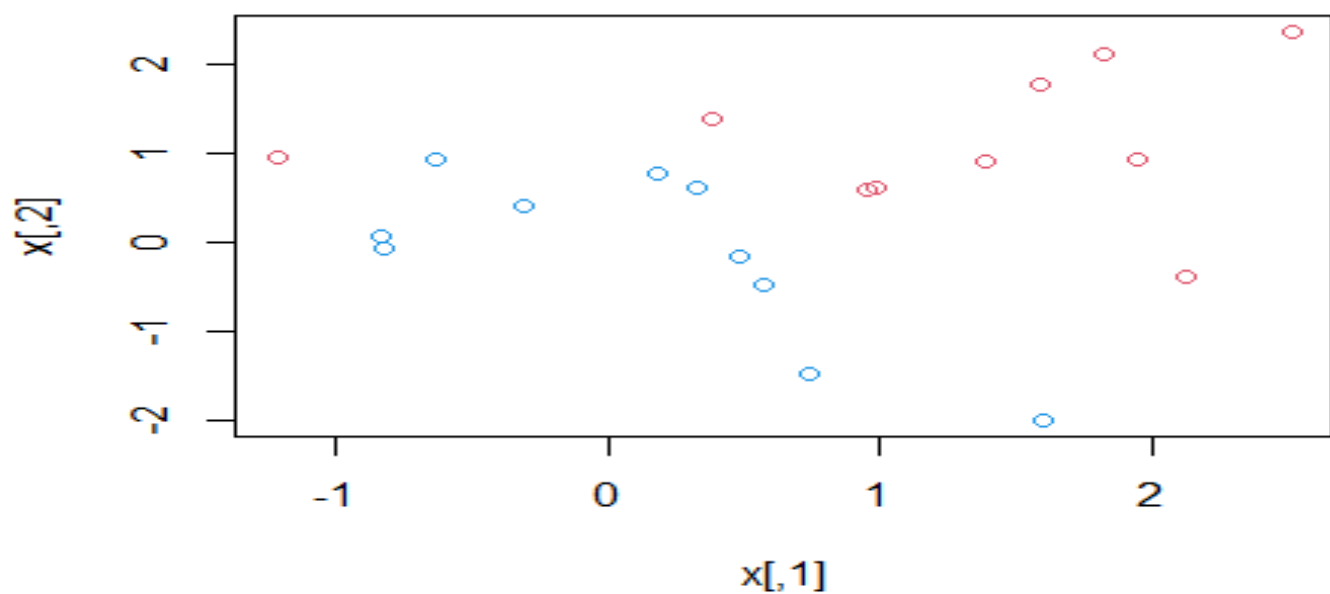


```
plot(-1:5, -1:5, type="n", xlab="X1", ylab="X2", main="Example 2")
points(c(3,2,4,1), c(4,2,4,2), col="red", pch=19)
points(c(2,4,4), c(1,3,1), col="blue", pch=19)
abline(-0.5, 1, col="green", lwd=2)
abline(-1, 1, col="black", lty="dotted")
abline(0, 1, col="black", lty="dotted")
```

## Example 2

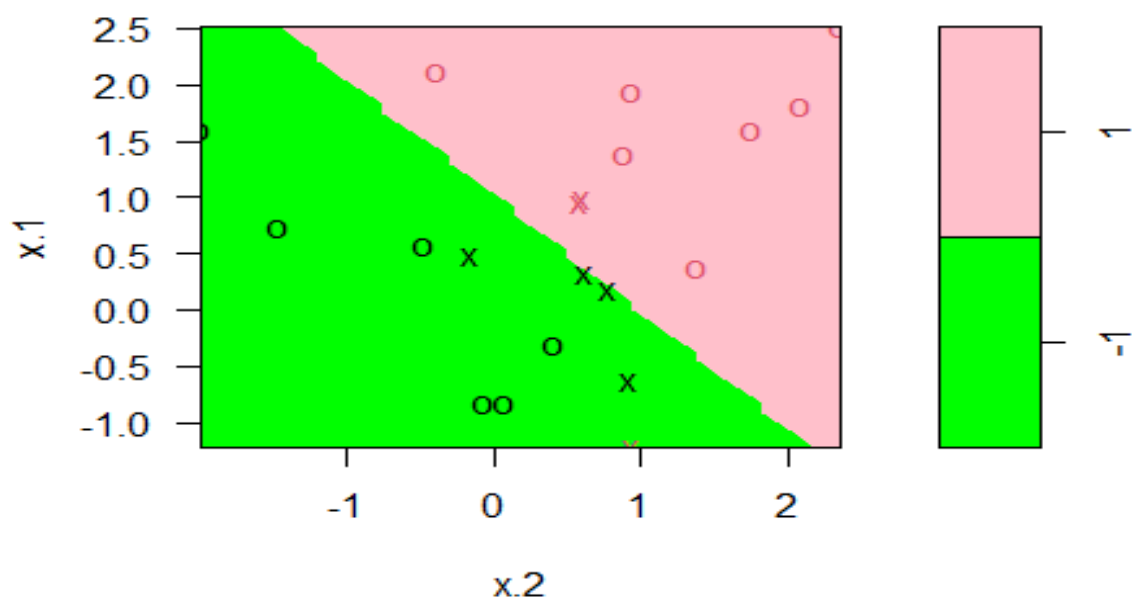


```
set.seed(1)
x=matrix(rnorm(20*2), ncol=2)
y=c(rep(-1,10), rep(1,10))
x[y==1,]=x[y==1,]+1
plot(x, col=(3-y))
```



```
dat=data.frame(x=x, y=as.factor (y))
library (e1071)
svmfit =svm(y ~ ., data=dat , kernel = "linear", cost =10,
scale =FALSE )
plot(svmfit , dat, col = c('green','pink'), pch = 20)
```

### SVM classification plot



```
svmfit$index
```

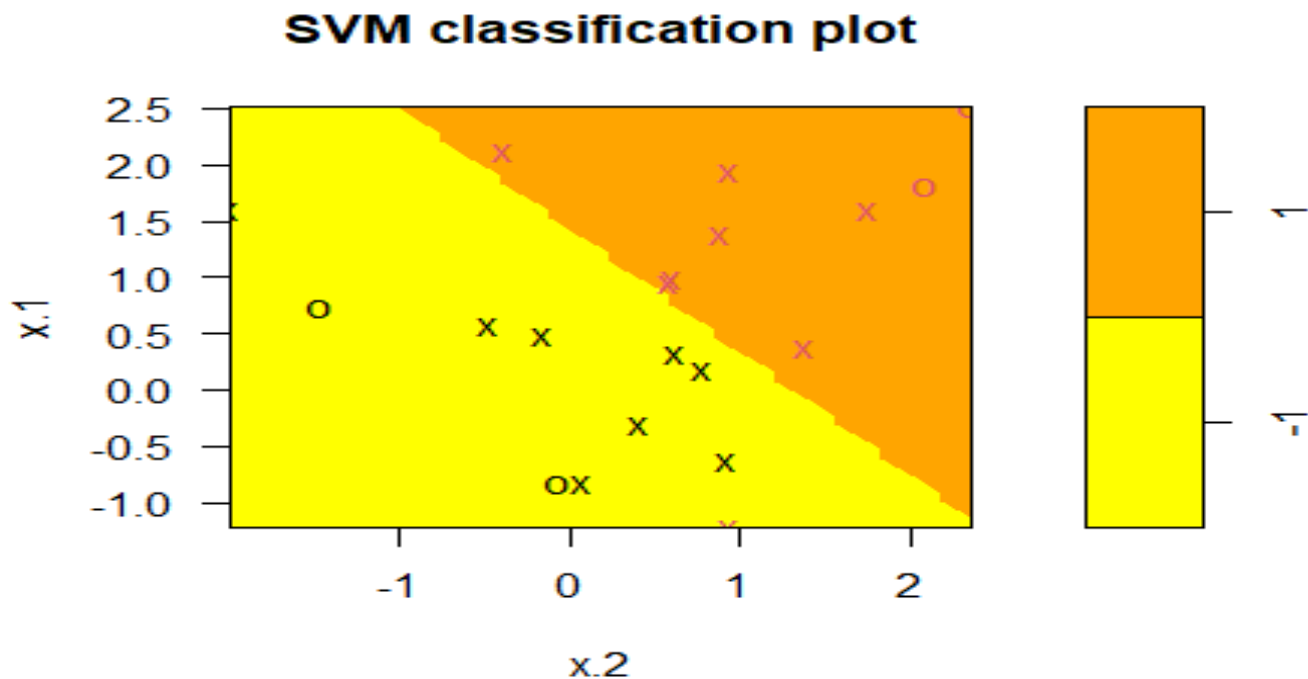
```
## [1] 1 2 5 7 14 16 17

summary(svmfit)

## Parameters:
##   SVM-Type: C-classification
##   SVM-Kernel: linear
##   cost: 10
## Number of Support Vectors: 7
## ( 4 3 )

## Number of Classes: 2
## Levels:
## -1 1

svmfit = svm(y ~., data=dat, kernel = "linear", cost = 0.1,
scale = FALSE)
plot(svmfit, dat, col = c('yellow', 'orange'), pch = 20)
```



```
svmfit$index

## [1] 1 2 3 4 5 7 9 10 12 13 14 15 16 17 18 20

set.seed(1)
tune.out = tune(svm, y ~., data=dat, kernel = "linear",
ranges = list(cost=c(0.001, 0.01, 0.1, 1, 5, 10, 100)))
summary(tune.out)

## - best parameters:
## cost
## 0.1
## - best performance: 0.05
```

```

bestmod = tune.out$best.model
summary(bestmod)

## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: linear
##           cost: 0.1
## Number of Support Vectors: 16
## ( 8 8 )
## Number of Classes: 2
## Levels:
## -1 1

xtest=matrix (rnorm (20*2) , ncol =2)
ytest=sample (c(-1,1) , 20, rep=TRUE)
xtest[ytest == 1 ,]= xtest[ytest ==1,] + 1
testdat = data.frame (x=xtest , y=as.factor(ytest))

ypred = predict(bestmod, testdat)
table(predict = ypred, truth= testdat$y )

##           truth
## predict -1 1
##          -1 9 1
##           1 2 8

svmfit = svm(y~. , data=dat, kernel ="linear", cost =0.01,
scale =FALSE)
ypred = predict(svmfit ,testdat )
table(predict = ypred, truth = testdat$y )

##           truth
## predict -1 1
##          -1 11 6
##           1 0 3

```