Chapter 9 Support Vector Machines problems 5

kartheek raj mulasa

12/28/2019

5 We have seen that we can fit an SVM with a non-linear kernel in order to perform classification using a non-linear decision boundary. We will now see that we can also obtain a non-linear decision boundary by performing logistic regression using non-linear transformations of the features.

(a) Generate a data set with n = 500 and p = 2, such that the observations belong to two classes with a quadratic decision boundary between them. For instance, you can do this as follows:

Answer

Data generated as given in the question.

```
set.seed(5)

X1=runif (500) -0.5
X2=runif (500) -0.5
Y=1*(X1^2-X2^2 > 0)
Y<-as.factor(Y)

head(Y)
```

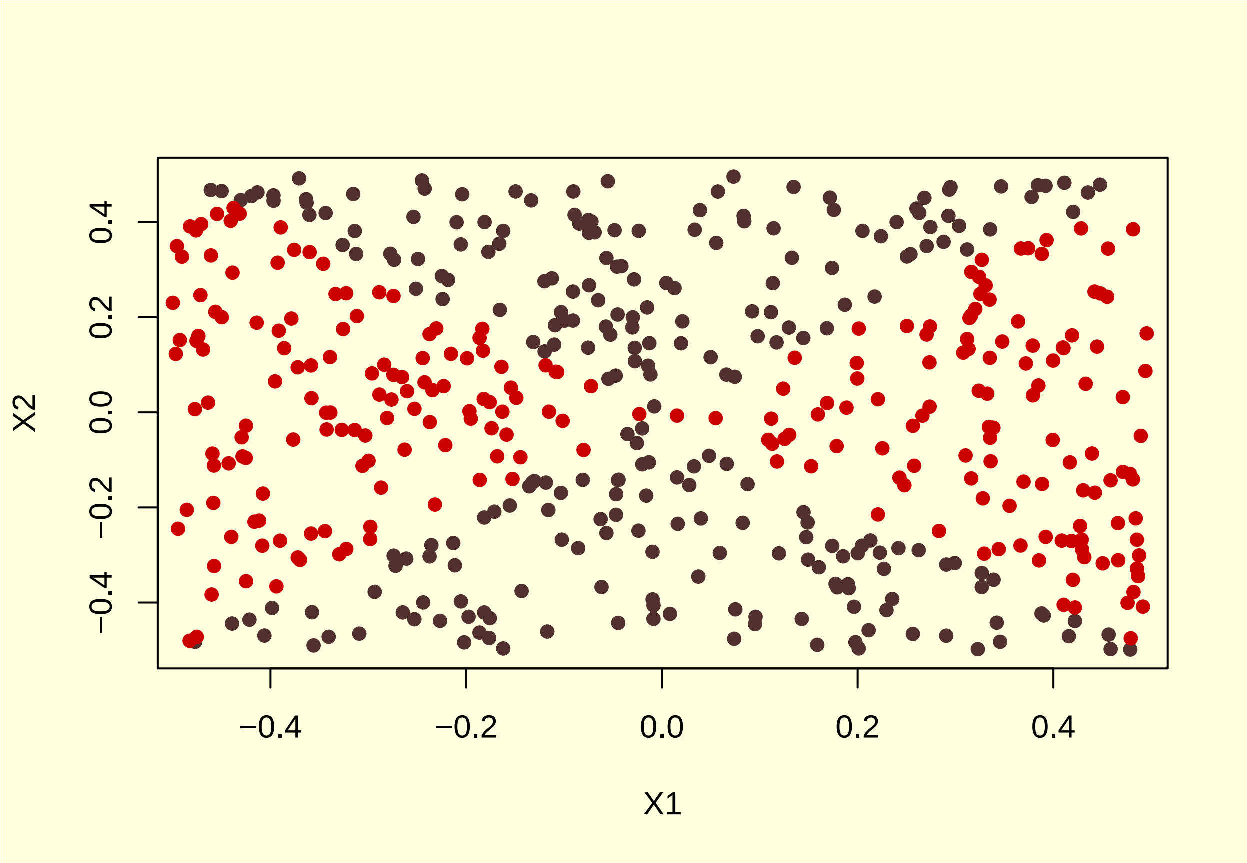
```
## [1] 1 0 1 1 1 0
## Levels: 0 1
```

(b) Plot the observations, colored according to their class labels. Your plot should display X1 on the x-axis, and X2 on the yaxis.

Answer

Plotting the data points as mentioned in the question.

```
par(mfrow=c(1,1),bg="lightyellow")

plot(X1[Y==0],X2[Y==0],xlab = "X1",ylab = "X2",col="#50312F",cex=1,pch=16,type="p")
points(X1[Y!=0],X2[Y!=0],col="#CB0000",cex=1,pch=16)
```



(c) Fit a logistic regression model to the data, using X1 and X2 as predictors.

Answer

Building the logi model to above data .

```
logi<-glm(Y~X1+X2,family = "binomial")

summary(logi)
```

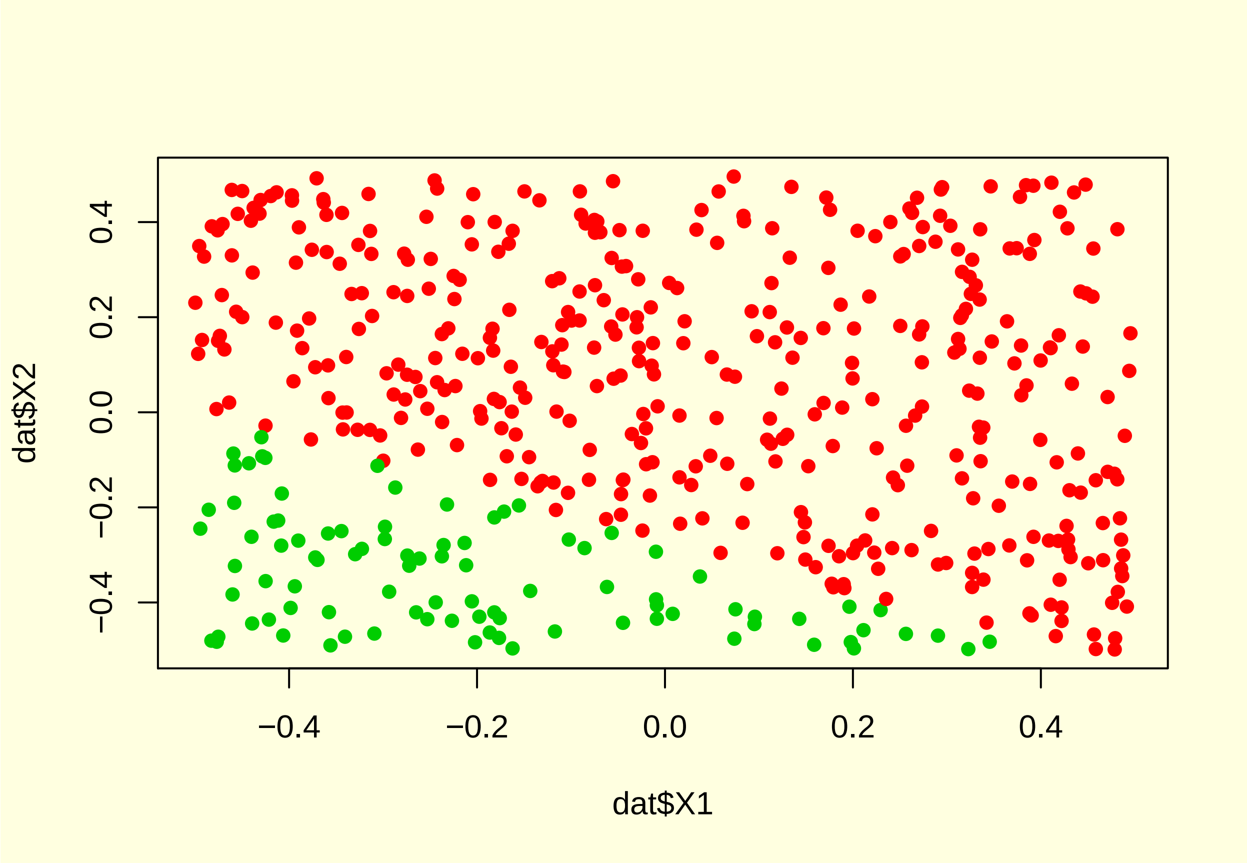
```
##
## Call:
## glm(formula = Y ~ X1 + X2, family = "binomial")
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.200  -1.161  -1.131   1.190   1.223
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.03150    0.08949  -0.352    0.725
## X1          -0.06176    0.30506  -0.202    0.840
## X2          -0.11509    0.31086  -0.370    0.711
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 693.02  on 499  degrees of freedom
```

```
## Residual deviance: 692.85  on 497  degrees of freedom
## AIC: 698.85
##
## Number of Fisher Scoring iterations: 3
```

(d) Apply this model to the training data in order to obtain a predicted class label for each training observation. Plot the observations, colored according to the predicted class labels. The decision boundary should be linear.

Answer

```r
par(mfrow=c(1,1),bg="lightyellow")

dat<-data.frame(X1,X2,Y)

logid<-glm(Y ~ ., data = dat, family = 'binomial')

lpm<-predict(logid,dat$Y,type="response")

lp<-ifelse(lpm>0.50,1,0)

plot(dat$X1, dat$X2, col = lp+ 2,cex=1,pch=16)
```



(e) Now fit a logistic regression model to the data using non-linear functions of X1 and X2 as predictors (e.g. X2 1, X1×X2, log(X2),and so forth).

Answer

```r
logip<-glm(Y~poly(X1,2)+poly(X2,2), data = dat, family = 'binomial')
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

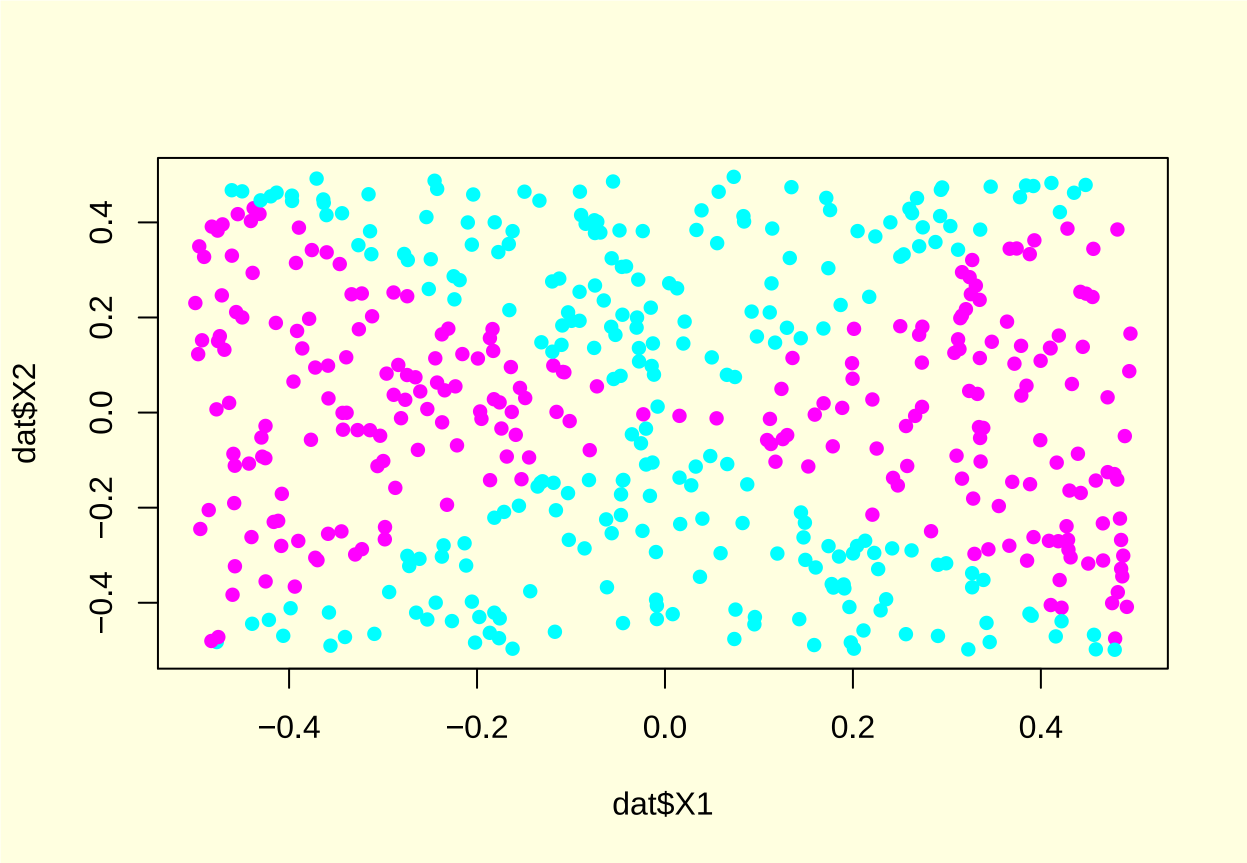
```r
summary(logip)
```

```
##
## Call:
## glm(formula = Y ~ poly(X1, 2) + poly(X2, 2), family = "binomial",
##     data = dat)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -0.003739   0.000000   0.000000   0.000000   0.003504
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)     252.0     3439.0   0.073    0.942
## poly(X1, 2)1   2792.7    40700.9   0.069    0.945
## poly(X1, 2)2 100456.5  1343613.6   0.075    0.940
## poly(X2, 2)1   -792.9    16782.0  -0.047    0.962
## poly(X2, 2)2 -99176.6  1325929.2  -0.075    0.940
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6.9302e+02  on 499  degrees of freedom
## Residual deviance: 2.8367e-05  on 495  degrees of freedom
## AIC: 10
##
## Number of Fisher Scoring iterations: 25
```

(f) Apply this model to the training data in order to obtain a predicted class label for each training observation. Plot the observations, colored according to the predicted class labels. The decision boundary should be obviously non-linear. If it is not, then repeat (a)-(e) until you come up with an example in which the predicted class labels are obviously non-linear.

Answer

```r
par(mfrow=c(1,1),bg="lightyellow")

lpm1<-predict(logip,dat$Y,type="response")

lp1<-ifelse(lpm1>0.50,1,0)

plot(dat$X1, dat$X2, col = lp1 + 5,cex=1,pch=16)
```

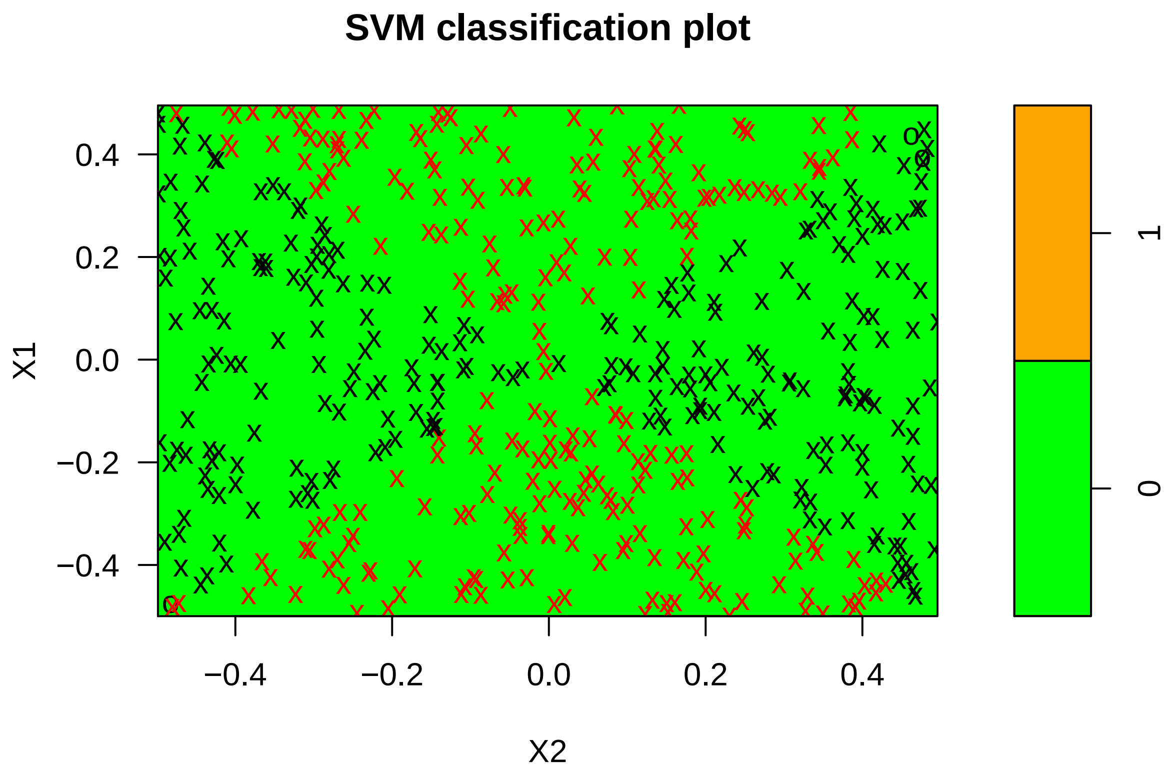


(g)Fit a support vector classifier to the data with X1 and X2 as predictors. Obtain a class prediction for each training observation. Plot the observations, colored according to the predicted class labels.

```
require(e1071)
```

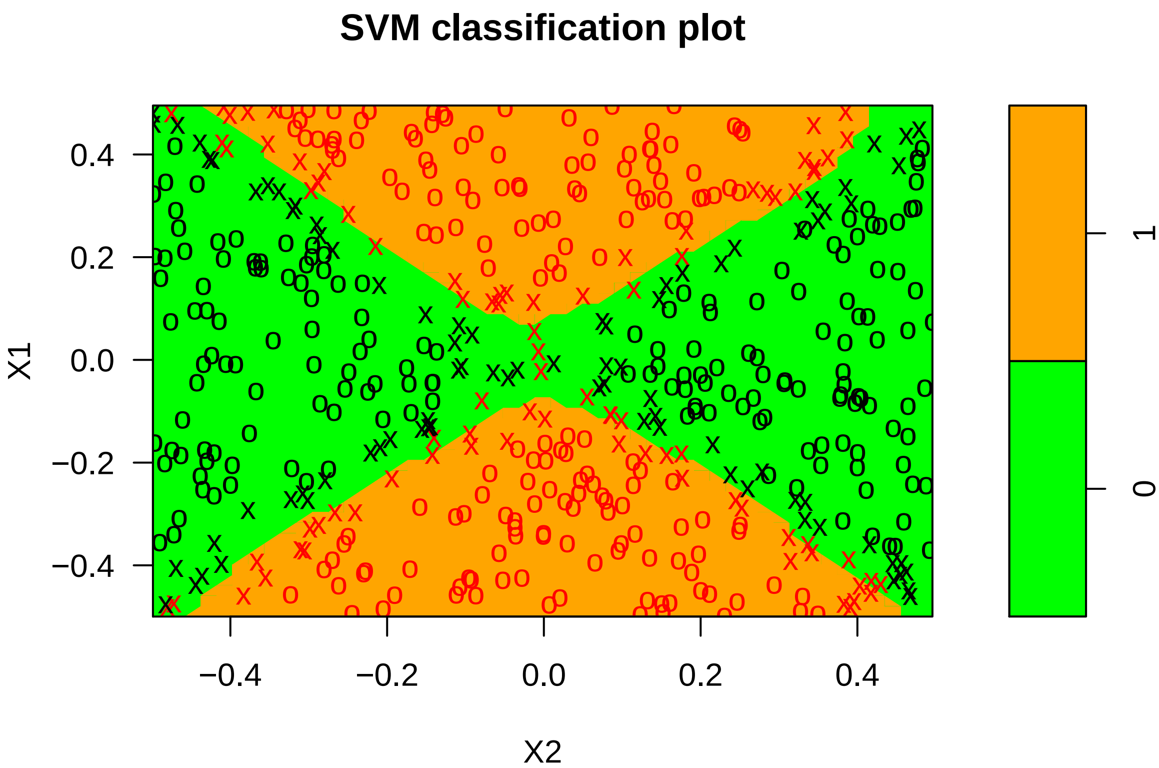
```
## Loading required package: e1071
```

```
library(e1071)

svm_linear<-svm(Y~.,data = dat, kernel="linear",cost=1)

plot(svm_linear,dat,col=c("green","orange"))
```



(h) Fit a SVM using a non-linear kernel to the data. Obtain a class prediction for each training observation. Plot the observations, colored according to the predicted class labels.

```
svm_radial<-svm(Y~.,data = dat,kernel="radial",gamma=1)

plot(svm_radial,dat,col=c("green","orange"))
```



(i) Comment on your results.

Logistic regression with polynomial predictors and svm radial yields better results juxtapose with any linear model.