SVMPractice

Mathias Bourel

23/11/2018

```
rm(list=ls())
library(e1071)

?svm
```

1- Datos Spam

```
library(kernlab)
data(spam)
datos=spam
summary(datos)
```

```
##       make           address             all             num3d
##  Min.   :0.0000   Min.   : 0.000   Min.   :0.0000   Min.   : 0.00000
##  1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.: 0.00000
##  Median :0.0000   Median : 0.000   Median :0.0000   Median : 0.00000
##  Mean   :0.1046   Mean   : 0.213   Mean   :0.2807   Mean   : 0.06542
##  3rd Qu.:0.0000   3rd Qu.: 0.000   3rd Qu.:0.4200   3rd Qu.: 0.00000
##  Max.   :4.5400   Max.   :14.280   Max.   :5.1000   Max.   :42.81000
##       our               over            remove           internet
##  Min.   : 0.0000   Min.   :0.0000   Min.   :0.0000   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median :0.0000   Median :0.0000   Median : 0.0000
##  Mean   : 0.3122   Mean   :0.0959   Mean   :0.1142   Mean   : 0.1053
##  3rd Qu.: 0.3800   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.: 0.0000
##  Max.   :10.0000   Max.   :5.8800   Max.   :7.2700   Max.   :11.1100
##      order              mail             receive              will
##  Min.   :0.00000   Min.   : 0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median : 0.0000   Median :0.00000   Median :0.1000
##  Mean   :0.09007   Mean   : 0.2394   Mean   :0.05982   Mean   :0.5417
##  3rd Qu.:0.00000   3rd Qu.: 0.1600   3rd Qu.:0.00000   3rd Qu.:0.8000
##  Max.   :5.26000   Max.   :18.1800   Max.   :2.61000   Max.   :9.6700
##      people            report            addresses           free
##  Min.   :0.00000   Min.   : 0.00000   Min.   :0.0000   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.: 0.00000   1st Qu.:0.0000   1st Qu.: 0.0000
##  Median :0.00000   Median : 0.00000   Median :0.0000   Median : 0.0000
##  Mean   :0.09393   Mean   : 0.05863   Mean   :0.0492   Mean   : 0.2488
##  3rd Qu.:0.00000   3rd Qu.: 0.00000   3rd Qu.:0.0000   3rd Qu.: 0.1000
##  Max.   :5.55000   Max.   :10.00000   Max.   :4.4100   Max.   :20.0000
##     business          email              you             credit
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   : 0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.: 0.00000
##  Median :0.0000   Median :0.0000   Median : 1.310   Median : 0.00000
##  Mean   :0.1426   Mean   :0.1847   Mean   : 1.662   Mean   : 0.08558
```

```
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.: 2.640   3rd Qu.: 0.00000
##  Max.   :7.1400   Max.   :9.0900   Max.   :18.750   Max.   :18.18000
##       your             font             num000          money
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :0.0000   Min.   : 0.00000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.: 0.00000
##  Median : 0.2200   Median : 0.0000   Median :0.0000   Median : 0.00000
##  Mean   : 0.8098   Mean   : 0.1212   Mean   :0.1016   Mean   : 0.09427
##  3rd Qu.: 1.2700   3rd Qu.: 0.0000   3rd Qu.:0.0000   3rd Qu.: 0.00000
##  Max.   :11.1100   Max.   :17.1000   Max.   :5.4500   Max.   :12.50000
##        hp              hpl              george            num650
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   :0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median :0.0000
##  Mean   : 0.5495   Mean   : 0.2654   Mean   : 0.7673   Mean   :0.1248
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:0.0000
##  Max.   :20.8300   Max.   :16.6600   Max.   :33.3300   Max.   :9.0900
##       lab                labs            telnet             num857
##  Min.   : 0.00000   Min.   :0.0000   Min.   : 0.00000   Min.   :0.00000
##  1st Qu.: 0.00000   1st Qu.:0.0000   1st Qu.: 0.00000   1st Qu.:0.00000
##  Median : 0.00000   Median :0.0000   Median : 0.00000   Median :0.00000
##  Mean   : 0.09892   Mean   :0.1029   Mean   : 0.06475   Mean   :0.04705
##  3rd Qu.: 0.00000   3rd Qu.:0.0000   3rd Qu.: 0.00000   3rd Qu.:0.00000
##  Max.   :14.28000   Max.   :5.8800   Max.   :12.50000   Max.   :4.76000
##       data              num415             num85          technology
##  Min.   : 0.00000   Min.   :0.00000   Min.   : 0.0000   Min.   :0.00000
##  1st Qu.: 0.00000   1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.00000
##  Median : 0.00000   Median :0.00000   Median : 0.0000   Median :0.00000
##  Mean   : 0.09723   Mean   :0.04784   Mean   : 0.1054   Mean   :0.09748
##  3rd Qu.: 0.00000   3rd Qu.:0.00000   3rd Qu.: 0.0000   3rd Qu.:0.00000
##  Max.   :18.18000   Max.   :4.76000   Max.   :20.0000   Max.   :7.69000
##     num1999         parts              pm               direct
##  Min.   :0.000   Min.   :0.0000   Min.   : 0.00000   Min.   :0.00000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.: 0.00000   1st Qu.:0.00000
##  Median :0.000   Median :0.0000   Median : 0.00000   Median :0.00000
##  Mean   :0.137   Mean   :0.0132   Mean   : 0.07863   Mean   :0.06483
##  3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.: 0.00000   3rd Qu.:0.00000
##  Max.   :6.890   Max.   :8.3300   Max.   :11.11000   Max.   :4.76000
##        cs             meeting            original        project
##  Min.   :0.00000   Min.   : 0.0000   Min.   :0.0000   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.: 0.0000
##  Median :0.00000   Median : 0.0000   Median :0.0000   Median : 0.0000
##  Mean   :0.04367   Mean   : 0.1323   Mean   :0.0461   Mean   : 0.0792
##  3rd Qu.:0.00000   3rd Qu.: 0.0000   3rd Qu.:0.0000   3rd Qu.: 0.0000
##  Max.   :7.14000   Max.   :14.2800   Max.   :3.5700   Max.   :20.0000
##        re               edu              table             conference
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :0.000000   Min.   : 0.00000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.000000   1st Qu.: 0.00000
##  Median : 0.0000   Median : 0.0000   Median :0.000000   Median : 0.00000
##  Mean   : 0.3012   Mean   : 0.1798   Mean   :0.005444   Mean   : 0.03187
##  3rd Qu.: 0.1100   3rd Qu.: 0.0000   3rd Qu.:0.000000   3rd Qu.: 0.00000
##  Max.   :21.4200   Max.   :22.0500   Max.   :2.170000   Max.   :10.00000
##  charSemicolon     charRoundbracket charSquarebracket charExclamation
##  Min.   :0.00000   Min.   :0.000    Min.   :0.00000   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.:0.000    1st Qu.:0.00000   1st Qu.: 0.0000
```

```
##  Median :0.00000   Median :0.065    Median :0.00000   Median : 0.0000
##  Mean   :0.03857   Mean   :0.139    Mean   :0.01698   Mean   : 0.2691
##  3rd Qu.:0.00000   3rd Qu.:0.188    3rd Qu.:0.00000   3rd Qu.: 0.3150
##  Max.   :4.38500   Max.   :9.752    Max.   :4.08100   Max.   :32.4780
##    charDollar          charHash          capitalAve        capitalLong
##  Min.   :0.00000   Min.   : 0.00000   Min.   :   1.000   Min.   :   1.00
##  1st Qu.:0.00000   1st Qu.: 0.00000   1st Qu.:   1.588   1st Qu.:   6.00
##  Median :0.00000   Median : 0.00000   Median :   2.276   Median :  15.00
##  Mean   :0.07581   Mean   : 0.04424   Mean   :   5.191   Mean   :  52.17
##  3rd Qu.:0.05200   3rd Qu.: 0.00000   3rd Qu.:   3.706   3rd Qu.:  43.00
##  Max.   :6.00300   Max.   :19.82900   Max.   :1102.500   Max.   :9989.00
##   capitalTotal          type
##  Min.   :    1.0   nonspam:2788
##  1st Qu.:   35.0   spam   :1813
##  Median :   95.0
##  Mean   :  283.3
##  3rd Qu.:  266.0
##  Max.   :15841.0
```

```r
set.seed(9146301)
ytable <- table(spam$type)
m=nrow(datos)
s=sample(m,m/3)
spam.learn <- spam[-s, ]
table(spam.learn$type)
```

```
##
## nonspam    spam
##    1888    1180
```

```r
n <- nrow(spam.learn)
p <- ncol(spam.learn) - 1
spam.test <- spam[s, ]
table(spam.test$type)
```

```
##
## nonspam    spam
##     900     633
```

```r
modelo=svm(type~.,data=spam.learn)
summary(modelo)
```

```
##
## Call:
## svm(formula = type ~ ., data = spam.learn)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.01754386
##
## Number of Support Vectors:  908
##
##  ( 427 481 )
```

```
##
##
## Number of Classes:  2
##
## Levels:
##  nonspam spam
```

```
predicciones <- predict(modelo, newdata=spam.test, type="class")
error=sum(predicciones!=spam.test$type)/nrow(spam.test)
error
```

```
## [1] 0.07632094
```

```
modelo1=svm(type~.,data=spam.learn,cross=10)
summary(modelo1)
```

```
##
## Call:
## svm(formula = type ~ ., data = spam.learn, cross = 10)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.01754386
##
## Number of Support Vectors:  908
##
##  ( 427 481 )
##
##
## Number of Classes:  2
##
## Levels:
##  nonspam spam
##
## 10-fold cross-validation on training data:
##
## Total Accuracy: 93.22034
## Single Accuracies:
##  94.77124 91.85668 95.11401 93.81107 95.11401 92.81046 89.90228 92.83388 92.50814 93.48534
```

Para tunear el costo:

2- Datos Breast

```
br=read.table("breast.txt",sep=",",header=F)
newdata=data.frame(br[,7],br[,10],as.factor(br[,2]))
colnames(newdata)=c("x.smoothness","x.concavepoints","clase")
head(newdata)
```

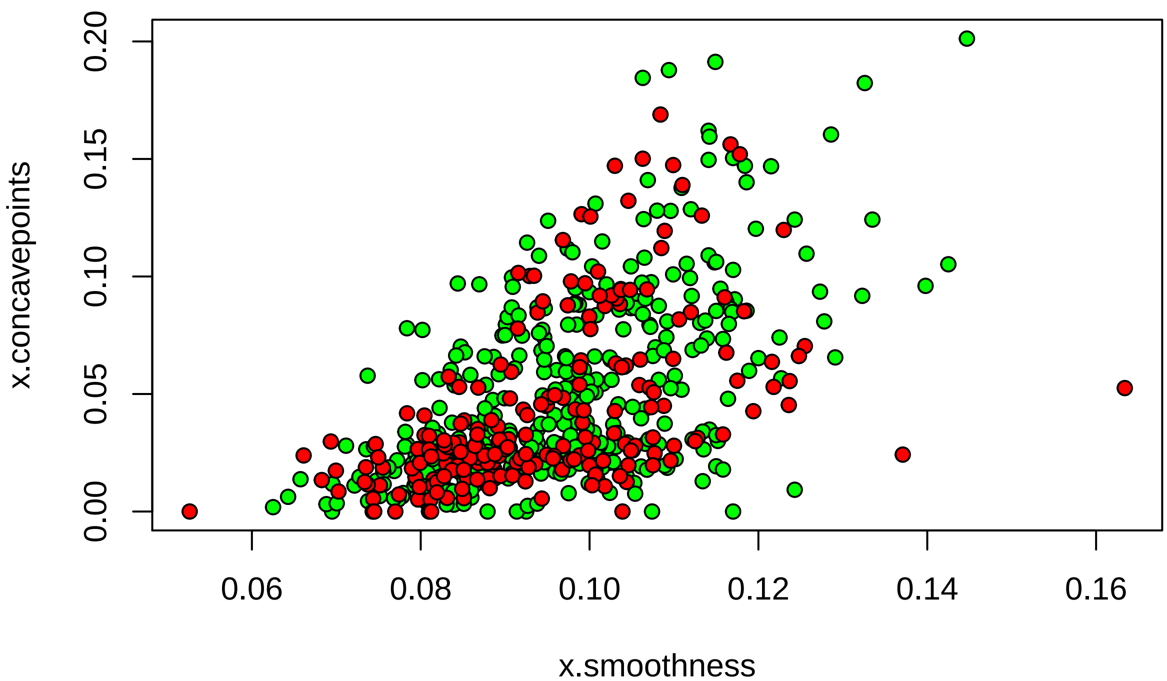
```
##   x.smoothness x.concavepoints clase
## 1      0.11840         0.14710     M
```

```
## 2      0.08474          0.07017     M
## 3      0.10960          0.12790     M
## 4      0.14250          0.10520     M
## 5      0.10030          0.10430     M
## 6      0.12780          0.08089     M
```

```r
summary(newdata)
```

```
##   x.smoothness      x.concavepoints    clase
##  Min.   :0.05263   Min.   :0.00000   B:357
##  1st Qu.:0.08637   1st Qu.:0.02031   M:212
##  Median :0.09587   Median :0.03350
##  Mean   :0.09636   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.20120
```

```r
plot(newdata[,1:2],pch=21,bg=c(rep('green',sum(newdata$clase=='B')),rep('red',sum(newdata$clase=='M')))
```




```r
svm.lineal=svm(newdata$clase~.,data=newdata,kernel='linear',cost=10,cross=2,scale=F)
summary(svm.lineal)
```

```
##
## Call:
## svm(formula = newdata$clase ~ ., data = newdata, kernel = "linear",
##     cost = 10, cross = 2, scale = F)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##       gamma:  0.5
##
## Number of Support Vectors:  258
```

```
##
##  ( 129 129 )
##
##
## Number of Classes:  2
##
## Levels:
##  B M
##
## 2-fold cross-validation on training data:
##
## Total Accuracy: 88.57645
## Single Accuracies:
##  88.73239 88.42105
```

```r
svm.lineal=svm(newdata$clase~.,data=newdata,kernel='linear',cost=10^3,cross=2,scale=F)
summary(svm.lineal)
```

```
##
## Call:
## svm(formula = newdata$clase ~ ., data = newdata, kernel = "linear",
##     cost = 10^3, cross = 2, scale = F)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1000
##       gamma:  0.5
##
## Number of Support Vectors:  128
##
##  ( 64 64 )
##
##
## Number of Classes:  2
##
## Levels:
##  B M
##
## 2-fold cross-validation on training data:
##
## Total Accuracy: 91.3884
## Single Accuracies:
##  92.60563 90.17544
```

Indice de los vectores soporte

```r
svm.lineal$index
```

```
##   [1]   8  11  12  14  16  17  32  36  37  39  40  41  42  44  45  48  55
##  [18]  65  74  87 100 101 120 127 128 133 136 142 172 183 185 187 191 194
##  [35] 197 200 206 208 214 215 216 224 230 256 262 264 265 275 278 298 338
##  [52] 354 380 386 415 436 442 445 461 490 502 515 537 567  20  77  82  89
##  [69]  90 112 113 124 125 129 134 148 149 153 201 209 222 226 228 239 248
##  [86] 276 285 287 289 291 292 319 341 356 357 364 376 377 381 383 396 397
```

```
## [103] 398 407 414 422 424 446 454 467 470 483 485 486 492 496 497 501 506
## [120] 509 514 519 529 531 542 559 560 561
```

Coeficientes por los que se multiplican las observaciones para obtener el vector perpendicular al hiperplano; termino independiente; cálculo del hiperplano

```
svm.lineal$coefs;
```

```
##              [,1]
##   [1,] 1000.0000
##   [2,] 1000.0000
##   [3,]  246.4754
##   [4,] 1000.0000
##   [5,] 1000.0000
##   [6,] 1000.0000
##   [7,] 1000.0000
##   [8,] 1000.0000
##   [9,] 1000.0000
##  [10,] 1000.0000
##  [11,] 1000.0000
##  [12,] 1000.0000
##  [13,] 1000.0000
##  [14,] 1000.0000
##  [15,] 1000.0000
##  [16,] 1000.0000
##  [17,] 1000.0000
##  [18,] 1000.0000
##  [19,] 1000.0000
##  [20,] 1000.0000
##  [21,] 1000.0000
##  [22,] 1000.0000
##  [23,] 1000.0000
##  [24,] 1000.0000
##  [25,] 1000.0000
##  [26,] 1000.0000
##  [27,] 1000.0000
##  [28,] 1000.0000
##  [29,] 1000.0000
##  [30,] 1000.0000
##  [31,] 1000.0000
##  [32,] 1000.0000
##  [33,] 1000.0000
##  [34,] 1000.0000
##  [35,] 1000.0000
##  [36,] 1000.0000
##  [37,] 1000.0000
##  [38,] 1000.0000
##  [39,] 1000.0000
##  [40,] 1000.0000
##  [41,] 1000.0000
##  [42,] 1000.0000
##  [43,] 1000.0000
##  [44,] 1000.0000
##  [45,] 1000.0000
##  [46,] 1000.0000
```

```
##  [47,]  1000.0000
##  [48,]  1000.0000
##  [49,]  1000.0000
##  [50,]  1000.0000
##  [51,]  1000.0000
##  [52,]  1000.0000
##  [53,]  1000.0000
##  [54,]  1000.0000
##  [55,]  1000.0000
##  [56,]  1000.0000
##  [57,]  1000.0000
##  [58,]  1000.0000
##  [59,]  1000.0000
##  [60,]  1000.0000
##  [61,]  1000.0000
##  [62,]  1000.0000
##  [63,]  1000.0000
##  [64,]  1000.0000
##  [65,] -1000.0000
##  [66,] -1000.0000
##  [67,] -1000.0000
##  [68,] -1000.0000
##  [69,] -1000.0000
##  [70,] -1000.0000
##  [71,] -1000.0000
##  [72,] -1000.0000
##  [73,]  -622.9104
##  [74,] -1000.0000
##  [75,] -1000.0000
##  [76,] -1000.0000
##  [77,] -1000.0000
##  [78,] -1000.0000
##  [79,] -1000.0000
##  [80,] -1000.0000
##  [81,] -1000.0000
##  [82,] -1000.0000
##  [83,] -1000.0000
##  [84,] -1000.0000
##  [85,] -1000.0000
##  [86,] -1000.0000
##  [87,] -1000.0000
##  [88,] -1000.0000
##  [89,] -1000.0000
##  [90,] -1000.0000
##  [91,] -1000.0000
##  [92,] -1000.0000
##  [93,] -1000.0000
##  [94,] -1000.0000
##  [95,] -1000.0000
##  [96,] -1000.0000
##  [97,] -1000.0000
##  [98,] -1000.0000
##  [99,] -1000.0000
## [100,] -1000.0000
```

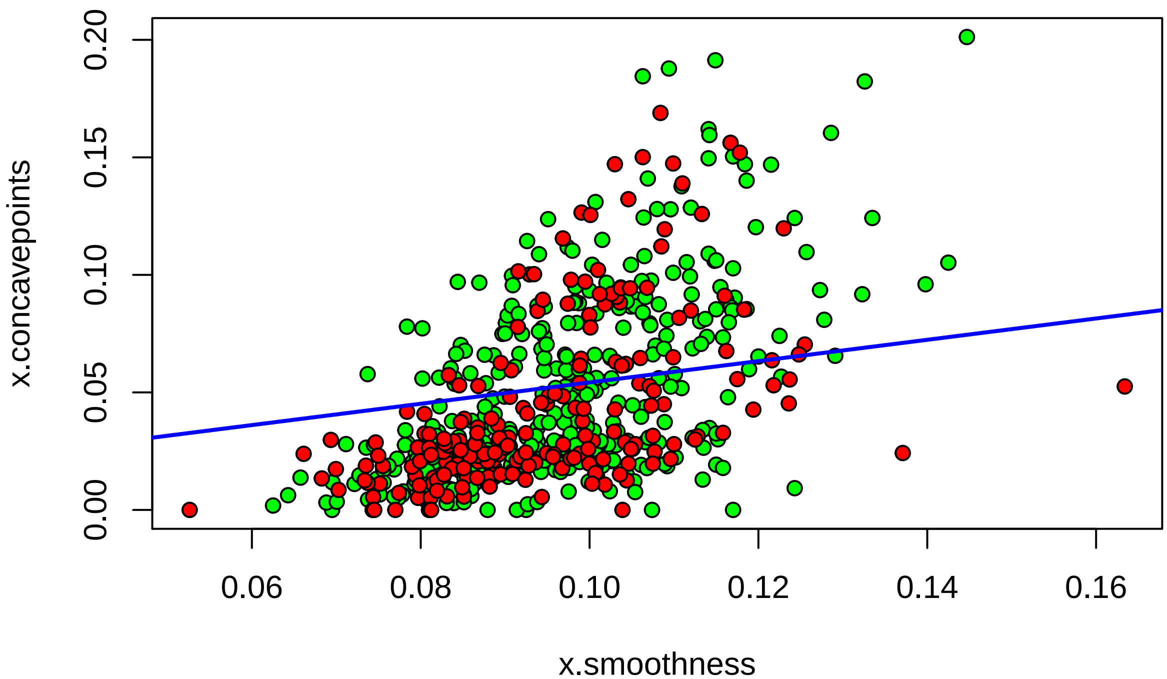
```
## [101,] -1000.0000
## [102,] -1000.0000
## [103,] -1000.0000
## [104,] -1000.0000
## [105,] -1000.0000
## [106,] -1000.0000
## [107,] -1000.0000
## [108,] -1000.0000
## [109,] -1000.0000
## [110,] -1000.0000
## [111,] -1000.0000
## [112,] -1000.0000
## [113,] -1000.0000
## [114,] -1000.0000
## [115,] -1000.0000
## [116,] -1000.0000
## [117,] -1000.0000
## [118,] -1000.0000
## [119,] -1000.0000
## [120,] -1000.0000
## [121,] -1000.0000
## [122,] -1000.0000
## [123,] -1000.0000
## [124,]  -623.5650
## [125,] -1000.0000
## [126,] -1000.0000
## [127,] -1000.0000
## [128,] -1000.0000
```

```r
svm.lineal$rho
```

```
## [1] 0.6746077
```

```r
newdata.svm=newdata[svm.lineal$index,1:2]
w=crossprod(as.matrix(newdata.svm),svm.lineal$coefs)
w0=svm.lineal$rho
```

```r
plot(newdata[,1:2],pch=21,bg=c(rep('green',sum(newdata$clase=='B')),rep('red',sum(newdata$clase=='M')))
abline(w0/w[2],-w[1]/w[2],lwd=2,col='blue')
```



3- la función tune.svm y probabilidades a posteriori.

```
data(iris)
n=nrow(iris)
s=sample(n,n/3)
iris.train=iris[-s,]
iris.test=iris[s,]
best <- tune.svm(Species~., data = iris.train,
                cost = 2^(2:8),
                kernel = "linear")

best2 <- tune.svm(Species~., data = iris.train,
                cost = 2^(2:8), gamma=c(1,0.5,1.25,4),
                kernel = "radial")

modelo=svm(Species~.,data=iris.train,cost=best2$best.model$cost,gamma=best2$best.model$gamma, probabili

prevsvmproba1=predict(modelo,newdata=iris.test,probability=T)
prevsvmproba1
```

```
##         16          57          94         129          96          78
##     setosa versicolor versicolor   virginica versicolor   virginica
##        101          99         115         121         102         149
##  virginica versicolor   virginica   virginica   virginica   virginica
##         34           3          41          68          52          48
##     setosa      setosa      setosa versicolor versicolor      setosa
##         54         104          38         141         140         131
## versicolor   virginica      setosa   virginica   virginica   virginica
##        117         111          83         130          62          65
##  virginica   virginica versicolor   virginica versicolor versicolor
```

```
##        108         61        134         70        112         91
##  virginica versicolor versicolor versicolor  virginica versicolor
##         66         50         63        148         19        145
## versicolor     setosa versicolor  virginica     setosa  virginica
##        146        107        147         28        110          9
##  virginica versicolor  virginica     setosa  virginica     setosa
##        137         46
##  virginica     setosa
## attr(,"probabilities")
##         setosa  versicolor   virginica
## 16  0.82568815 0.086520142 0.087791712
## 57  0.01643445 0.916227579 0.067337967
## 94  0.04372581 0.889644092 0.066630095
## 129 0.01446048 0.003341569 0.982197951
## 96  0.02413403 0.966196044 0.009669930
## 78  0.01990286 0.440222458 0.539874679
## 101 0.04946805 0.043735993 0.906795954
## 99  0.03766847 0.930354998 0.031976531
## 115 0.03064029 0.008795168 0.960564545
## 121 0.02043939 0.020759848 0.958800766
## 102 0.01458301 0.025317982 0.960099004
## 149 0.04889714 0.154586957 0.796515907
## 34  0.93976889 0.031182123 0.029048984
## 3   0.96887675 0.016619236 0.014504012
## 41  0.97230121 0.014920747 0.012778042
## 68  0.01953681 0.973460394 0.007002800
## 52  0.01327829 0.959290386 0.027431322
## 48  0.96639252 0.018054667 0.015552816
## 54  0.01133852 0.928868700 0.059792781
## 104 0.01490344 0.015035887 0.970060677
## 38  0.96879075 0.016718162 0.014491093
## 141 0.02146271 0.009693585 0.968843702
## 140 0.01657557 0.026774683 0.956649746
## 131 0.02906894 0.071970721 0.898960340
## 117 0.01596354 0.040948062 0.943088397
## 111 0.01972443 0.089660975 0.890614592
## 83  0.01368323 0.980063797 0.006252975
## 130 0.03693838 0.326736841 0.636324779
## 62  0.01449859 0.950836797 0.034664613
## 65  0.02744393 0.961069459 0.011486610
## 108 0.02794680 0.061942302 0.910110896
## 61  0.10497330 0.703420436 0.191606263
## 134 0.01684443 0.604840953 0.378314621
## 70  0.01106481 0.976079613 0.012855579
## 112 0.01358972 0.016813345 0.969596937
## 91  0.01108057 0.958303723 0.030615707
## 66  0.01537917 0.973148726 0.011472107
## 50  0.96982880 0.016072793 0.014098411
## 63  0.02598130 0.944523733 0.029494969
## 148 0.01443629 0.014604648 0.970959058
## 19  0.95756972 0.023572985 0.018857291
## 145 0.03664179 0.037523333 0.925834878
## 146 0.01866258 0.007928889 0.973408536
## 107 0.04480676 0.635810763 0.319382478
```

```
## 147 0.01678429 0.113482880 0.869732832
## 28  0.97051412 0.016022262 0.013463613
## 110 0.05864274 0.102152603 0.839204654
## 9   0.96037481 0.022362866 0.017262324
## 137 0.05146309 0.107311760 0.841225147
## 46  0.96342727 0.020888435 0.015684292
## Levels: setosa versicolor virginica
```

```r
prevsvm=as.numeric(prevsvmproba1) #clases
prevsvm
```

```
##  [1] 1 2 2 3 2 3 3 2 3 3 3 3 1 1 1 2 2 1 2 3 1 3 3 3 3 3 2 3 2 2 3 2 2 2 3
## [36] 2 2 1 2 3 1 3 3 2 3 1 3 1 3 1
```

4- Svm cuadrático

```r
svm.cuadratico=svm(newdata$clase~.,data=newdata,kernel='polynomial',degree=2, gamma=1,
                   cost=10,cross=2,scale=F)
summary(svm.cuadratico)
```

```
##
## Call:
## svm(formula = newdata$clase ~ ., data = newdata, kernel = "polynomial",
##     degree = 2, gamma = 1, cost = 10, cross = 2, scale = F)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  2
##       gamma:  1
##      coef.0:  0
##
## Number of Support Vectors:  424
##
##  ( 212 212 )
##
##
## Number of Classes:  2
##
## Levels:
##  B M
##
## 2-fold cross-validation on training data:
##
## Total Accuracy: 62.74165
## Single Accuracies:
##  60.91549 64.5614
```