Máquinas de vectores soporte con R

Datos

Cargamos la librería en la que se encuentran las funciones que vamos a utilizar y el fichero con los datos necesarios:

```
library(MASS)
library(e1071)
load(url('http://www.uam.es/joser.berrendero/datos/practica-svm-io.RData'))
```

Los datos proceden de un estudio sobre diagnóstico del cáncer de mama por imagen. Mediante una punción con aguja fina se extrae una muestra del tejido sospechoso de la paciente. La muestra se tiñe para resaltar los núcleos de las células y se determinan los límites exactos de los núcleos. Las variables consideradas corresponden a distintos aspectos de la forma del núcleo. El fichero contiene un `data.frame`, llamado `breast.cancer2`, con 2 variables explicativas medidas en pacientes cuyos tumores fueron diagnosticados posteriormente como benignos o malignos y el factor `y` que toma los valores 0 o 1 en función de si la correspondiente fila corresponde a un tumor benigno o maligno respectivamente. Más información sobre los datos se puede encontrar en esta dirección. En el fichero original se considera un total de 30 variables explicativas. Hay 569 observaciones, de las que 357 corresponden a tumores benignos y 212 a tumores malignos.

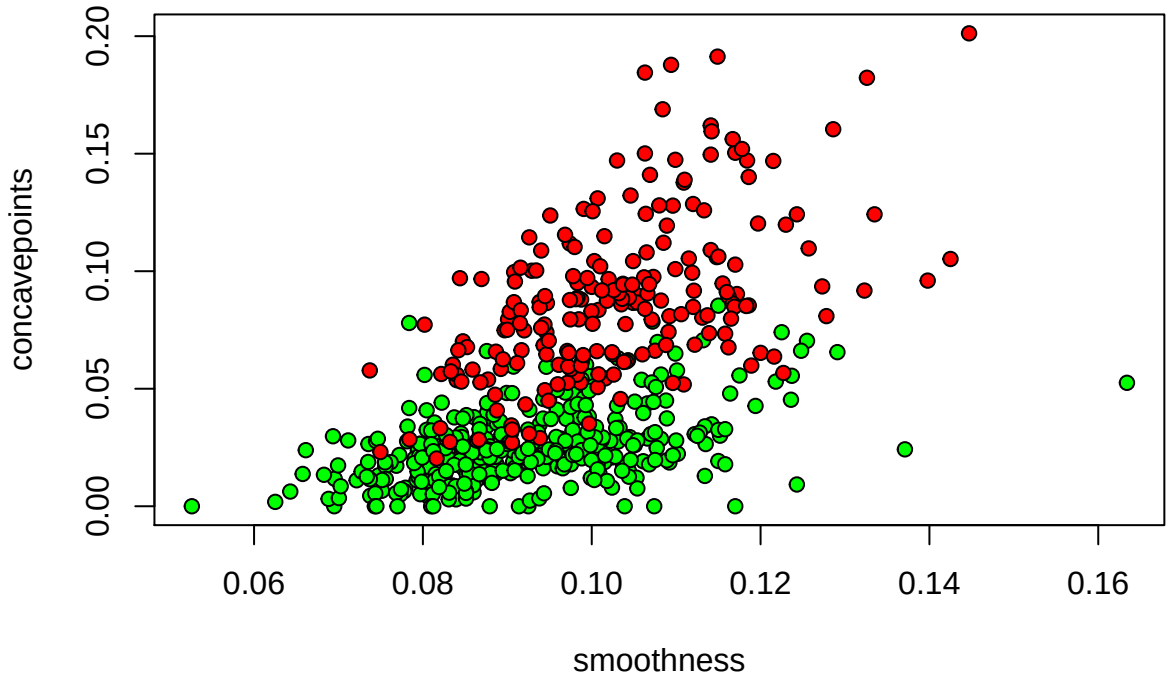
```
head(breast.cancer2)
```

```
##    x.smoothness x.concavepoints y
## 20      0.09779        0.047810 0
## 21      0.10750        0.031100 0
## 22      0.10240        0.020760 0
## 38      0.08983        0.029230 0
## 47      0.08600        0.005917 0
## 49      0.10310        0.027490 0
```

Representamos gráficamente las dos variables predictivas:

```
# Prepara los datos
x <- cbind(breast.cancer2$x.smoothness, breast.cancer2$x.concavepoints)
y <- breast.cancer2$y
n0 <- sum(y==0)
n1 <- sum(y==1)
# Para que los graficos queden mas bonitos (rojo = maligno, verde = benigno)
colores <- c(rep('green',n0),rep('red',n1))
pchn <- 21

# Diagrama de dispersion
plot(x, pch = pchn, bg = colores, xlab='smoothness', ylab='concavepoints')
```



SVM lineal

Para calcular la regla de clasificación SVM lineal con $C = 10$, se usa la función **svm** del paquete **e1071**. El primer argumento `y~.` indica que la variable `y` (que debe ser necesariamente un factor) se desea predecir en términos del resto de variables del fichero. La sintaxis es similar a la que utilizaríamos para ajustar un modelo lineal o un modelo de regresión logística. El segundo argumento indica el fichero en el que están las variables que vamos a usar. El argumento `kernel` corresponde al núcleo que representa el producto escalar que queremos utilizar. La opción `linear` corresponde a $k(x, y) = x'y$. El argumento `cost` determina la penalización que ponemos a los errores de clasificación. Con el fin de estimar la probabilidad de clasificar erróneamente una observación se puede utilizar validación cruzada, dividiendo la muestra en, por ejemplo, dos partes. Ello se consigue fijando `cross=2`. Finalmente, `scale=FALSE` se usa para usar los datos no estandarizados (por defecto, sí se estandarizan).

```
C <- 10
svm.lineal <- svm(y~., data=breast.cancer2, kernel='linear', cost=C, cross=2, scale=FALSE)
summary(svm.lineal)
```

```
##
## Call:
## svm(formula = y ~ ., data = breast.cancer2, kernel = "linear",
##     cost = C, cross = 2, scale = FALSE)
##
##
## 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:
##  0 1
##
## 2-fold cross-validation on training data:
##
## Total Accuracy: 88.04921
## Single Accuracies:
##  89.43662 86.66667
```

Hay 258 vectores soporte. Usando validación cruzada con la muestra dividida en dos partes se estima una probabilidad de acierto en la clasificación de aproximadamente el 88%. Podemos cambiar el parámetro de penalización para ver si estos valores aumentan o disminuyen.

```
C <- 10^3
svm.lineal <- svm(y~., data=breast.cancer2, kernel='linear', cost=C, cross=2, scale=FALSE)
summary(svm.lineal)
```

```
##
## Call:
## svm(formula = y ~ ., data = breast.cancer2, kernel = "linear",
##     cost = C, cross = 2, scale = FALSE)
##
##
## 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:
##  0 1
##
## 2-fold cross-validation on training data:
##
## Total Accuracy: 91.73989
```

```
## Single Accuracies:
##  91.5493 91.92982
```

Cuestiones

- ¿Es preferible $C = 10$ o $C = 10^3$?
- ¿Cambia mucho el porcentaje de acierto estimado si en lugar de dividir la muestra en dos partes, lo hacemos en 10?

El objeto `svm.lineal`que hemos generado es una lista que incluye información más detallada sobre los resultados:

```
# Indices de los vectores soporte
svm.lineal$index
```

```
##   [1]   1  23  26  28  29  44  45  51  52  54  56  65  66  70  97 100 106
##  [18] 109 111 117 124 135 140 142 144 146 147 170 184 195 196 203 208 209
##  [35] 212 214 223 224 225 233 239 245 247 263 268 279 281 293 295 296 300
##  [52] 303 304 306 309 312 315 317 326 328 336 353 354 355 365 368 369 371
##  [69] 373 374 386 390 391 392 393 394 395 397 398 400 402 406 410 417 422
##  [86] 423 428 431 432 435 437 439 447 452 453 454 455 456 458 461 465 466
## [103] 469 470 471 474 475 486 492 494 495 498 499 503 513 518 526 527 533
## [120] 538 539 540 544 549 553 557 563 568
```

```
# Coeficientes por los que se multiplican las observaciones para obtener
# el vector perpendicular al hiperplano que resuelve el problema
svm.lineal$coefs
```

```
##              [,1]
##   [1,]  1000.0000
##   [2,]  1000.0000
##   [3,]  1000.0000
##   [4,]  1000.0000
##   [5,]  1000.0000
##   [6,]  1000.0000
##   [7,]  1000.0000
##   [8,]  1000.0000
##   [9,]   623.3655
##  [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,]   620.7901
##  [61,]  1000.0000
##  [62,]  1000.0000
##  [63,]  1000.0000
##  [64,]  1000.0000
##  [65,] -1000.0000
##  [66,] -1000.0000
##  [67,]  -244.1556
##  [68,] -1000.0000
##  [69,] -1000.0000
##  [70,] -1000.0000
##  [71,] -1000.0000
##  [72,] -1000.0000
##  [73,] -1000.0000
##  [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,] -1000.0000
## [125,] -1000.0000
## [126,] -1000.0000
## [127,] -1000.0000
## [128,] -1000.0000
```

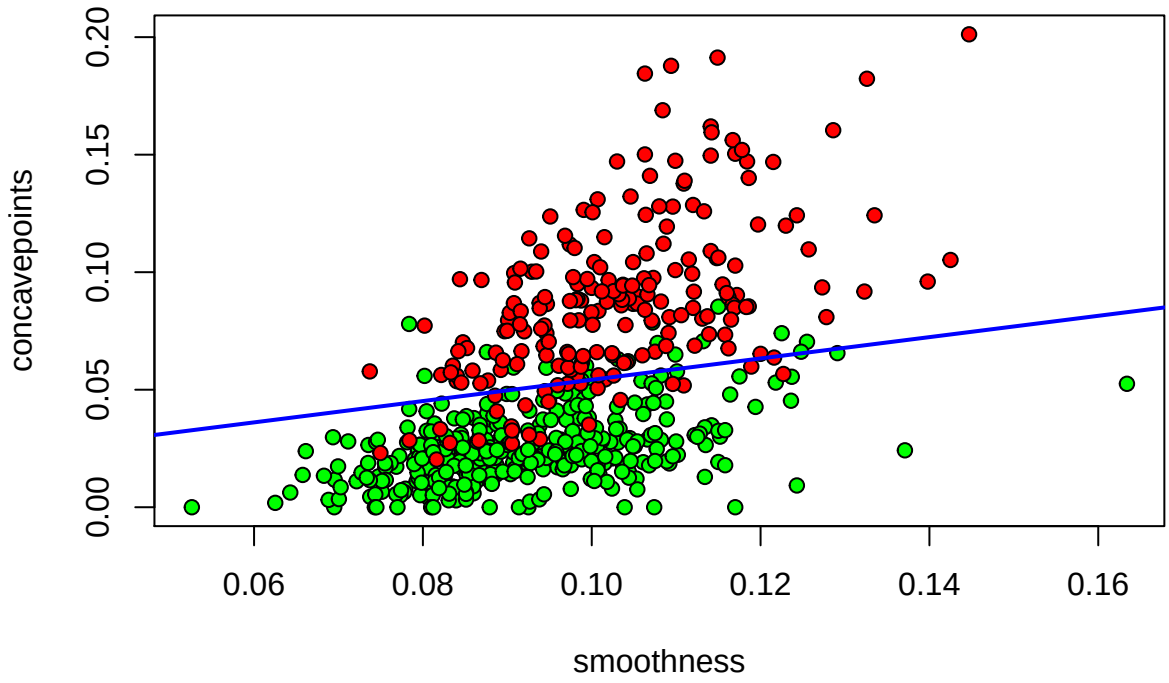
```r
# Termino independiente
svm.lineal$rho
```

```
## [1] -0.6763177
```

Con toda esta información es posible calcular el hiperplano que da la la regla de clasificación $w_0 + w'x = 0$, donde w_0 es el valor contenido en `svm.lineal$rho` cambiado de signo, y $w = \sum_{i \in S} \alpha_i x_i$ (con S el conjunto de índices de los vectores soporte) y representar gráficamente la regla resultante:

```
x.svm <- x[svm.lineal$index,]
w <- crossprod(x.svm, svm.lineal$coefs)
w0 <- svm.lineal$rho

plot(x, pch = pchn, bg = colores, xlab='smoothness', ylab='concavepoints')
abline(w0/w[2], -w[1]/w[2], lwd=2, col='blue')
```



SVM cuadrático

Si queremos obtener una regla de clasificación cuadrática en lugar de una lineal basta cambiar el núcleo. Por ejemplo, para usar el núcleo polinómico de segundo grado, $k(x, y) = (x'y + 1)^2$, podemos usar el comando:

```
svm.cuadratico <- svm(y~., data=breast.cancer2, kernel='polynomial', degree=2, gamma=1, coef0=1, cost=C
summary(svm.cuadratico)
```

```
##
## Call:
## svm(formula = y ~ ., data = breast.cancer2, kernel = "polynomial",
```

```
##     degree = 2, gamma = 1, coef0 = 1, cost = C, cross = 10, scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1000
##      degree:  2
##       gamma:  1
##      coef.0:  1
##
## Number of Support Vectors:  124
##
##  ( 62 62 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
##
## 10-fold cross-validation on training data:
##
## Total Accuracy: 91.56415
## Single Accuracies:
##  85.71429 91.22807 89.47368 89.47368 98.24561 87.7193 96.49123 96.49123 87.7193 92.98246
```

Se puede consultar la ayuda para ver la lista de todos los núcleos disponibles.

Ampliar información

Una referencia útil para ampliar información sobre cómo aplicar SVM con R es Karatzoglou et al. (2006). En este artículo se comparan otras implementaciones de las SVM con la que hemos discutido en esta práctica.