Clasificadores y Máquinas de Vectores soporte
Aprendizaje Estadístico, 2024/2025

UNIVERSIDAD DE
MURCIA

Semana del 26 de Noviembre, 2024

Aprendizaje Estadístico

Máster en Tecnología de Análisis de Datos Masivos - Big Data.

Juan A. Botía (juanbot@um.es)

Máquinas de vectores soporte

- Abordamos el problema de **clasificación binaria** de una manera directa: *Intentamos encontrar un hiperplano que separe las clases en el espacio de características.*
- Si no podemos hacerlo, nos volvemos creativos de dos maneras:
  - Suavizamos lo que queremos decir con "separar las clases", y
  - Enriquecemos y ampliamos el espacio de características para que la separación sea posible.

¿Qué es un hiperplano?

Un hiperplano en p dimensiones es un subespacio afín plano de dimensión $p - 1$.

En general, la ecuación de un hiperplano tiene la forma:

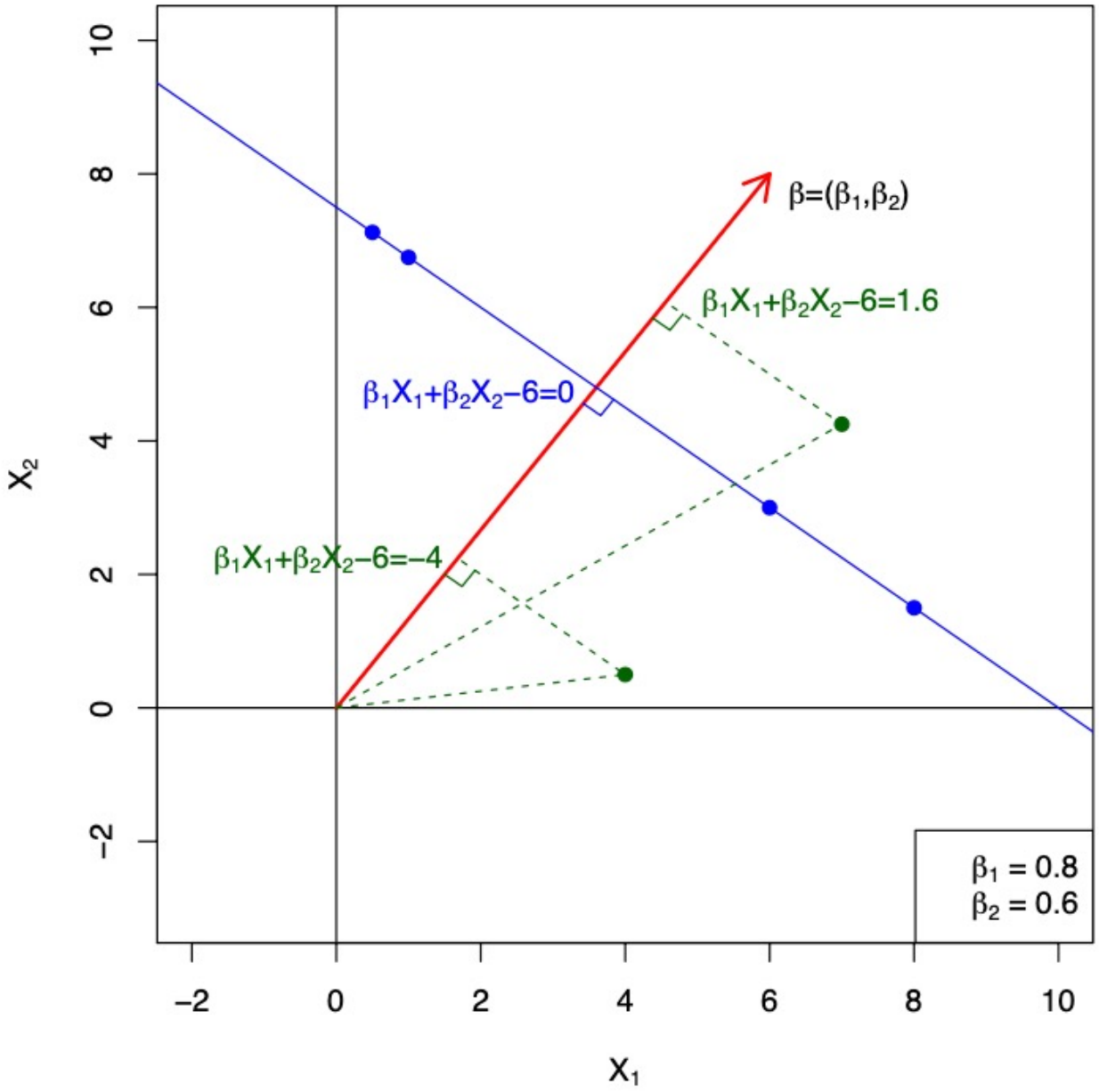
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

En $p = 2$ dimensiones, un hiperplano es una línea.

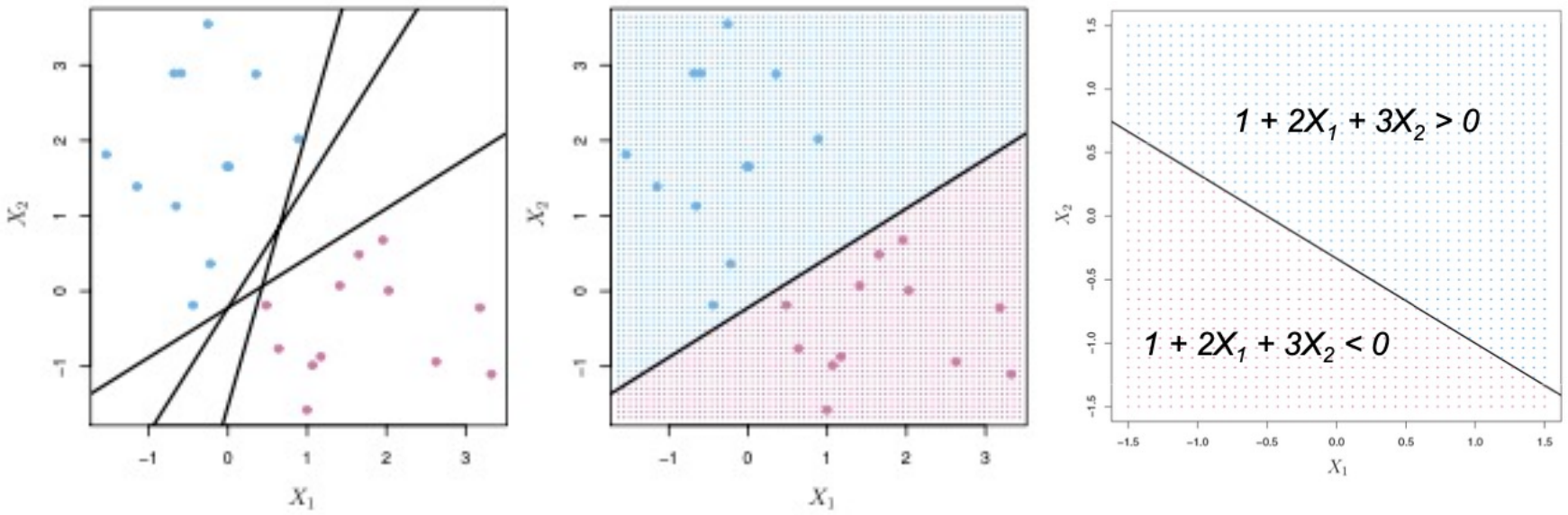
Si $\beta_0 = 0$, el hiperplano pasa por el origen, de lo contrario, no.

El vector $\beta = (\beta_1, \beta_2, \ldots, \beta_p)$ se llama vector normal: apunta en una dirección ortogonal a la superficie de un hiperplano.

Hiperplano en 2 dimensiones,



Hiperplanos separadores

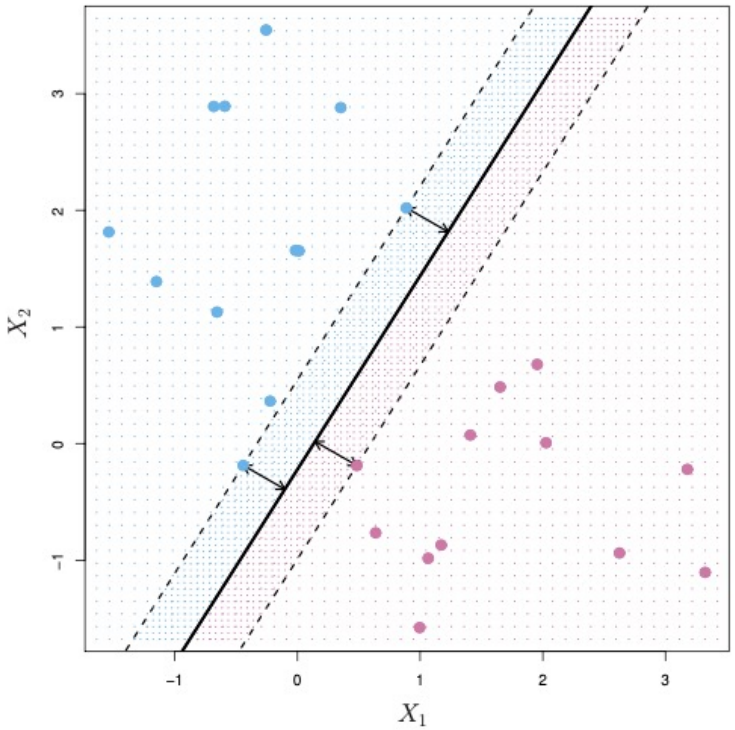


Si $f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$, entonces $f(X) > 0$ para puntos en un lado del hiperplano, y $f(X) < 0$ para puntos en el otro lado.

- Si codificamos los puntos coloreados como $Y_i = +1$ para azul, por ejemplo, y $Y_i = -1$ para malva, entonces si $Y_i \cdot f(X_i) > 0$ para todos i, $f(X) = 0$ define un hiperplano separador.

Clasificador de margen máximo

- Buscamos el hiperplano separador que maximiza el espacio (margen) entre las dos clases



Constrained optimization problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M$$

$$\text{for all} \quad i = 1, \ldots, N.$$

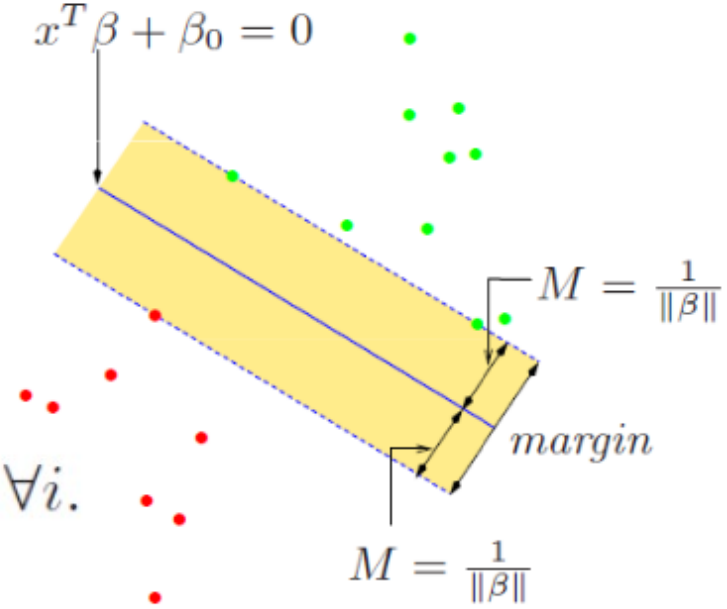
Este problema es un problema cuadrático convexo (ver paquete e1017, función svm()).
La primera condición regulariza y añade significado a la segunda.
La segunda asegura que todo punto está en el lado en el que tiene que estar.
El clasificador es, entonces $G(x)=\text{sign}[x^T b + b_0]$

Resolución del problema

- Reformulamos el problema según

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2$$

$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1$$



- Y esto nos lleva a la solución

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i, \;\; \alpha_i[y_i(x_i^T\beta + \beta_0) - 1] = 0 \;\forall i.$$

- Resulta que solo nos interesan los x_i según

$$\text{if } \alpha_i > 0, \text{ then } y_i(x_i^T\beta + \beta_0) = 1$$

- Es decir, los que están sobre el hiperplano!!
- Para resolverlo, nos basta con hacerlo para cualquier vector soporte (normalmente promediamos todos)

Sobre el problema anterior

- Solo tiene solución si todos los puntos caen a un lado o bien al otro, se ha de cumplir para todos los puntos que

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0 \text{ if } y_i = 1$$
$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0 \text{ if } y_i = -1$$
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0$$

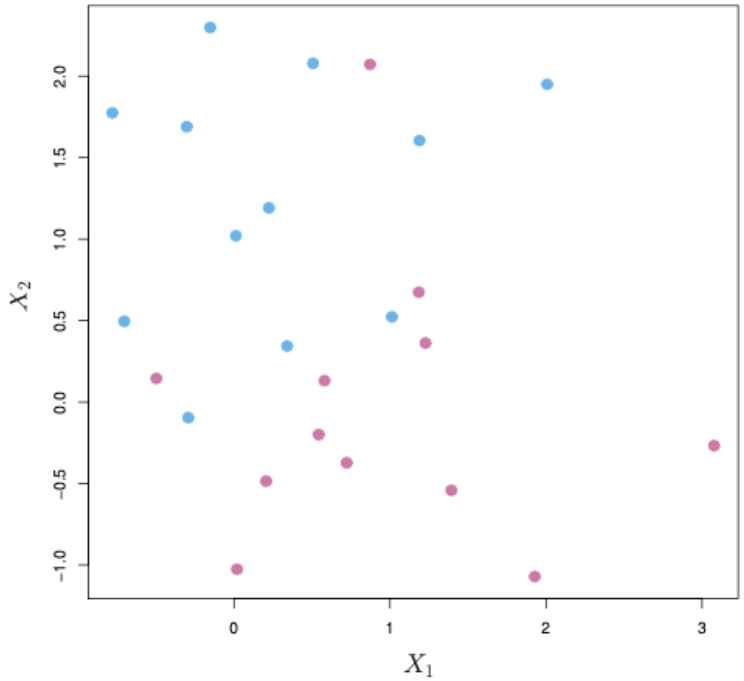
- Si $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_p x_p^*$

es muy grande (positivo o negativo) está muy alejado del hiperplano, y estaremos más seguros de su correcta clasificación

- Cuando p es grande, este clasificador puede generar overfitting
- Los vectores soporte son los puntos que soportan en hiperplano: *Si se movieran, el hiperplano también se movería*

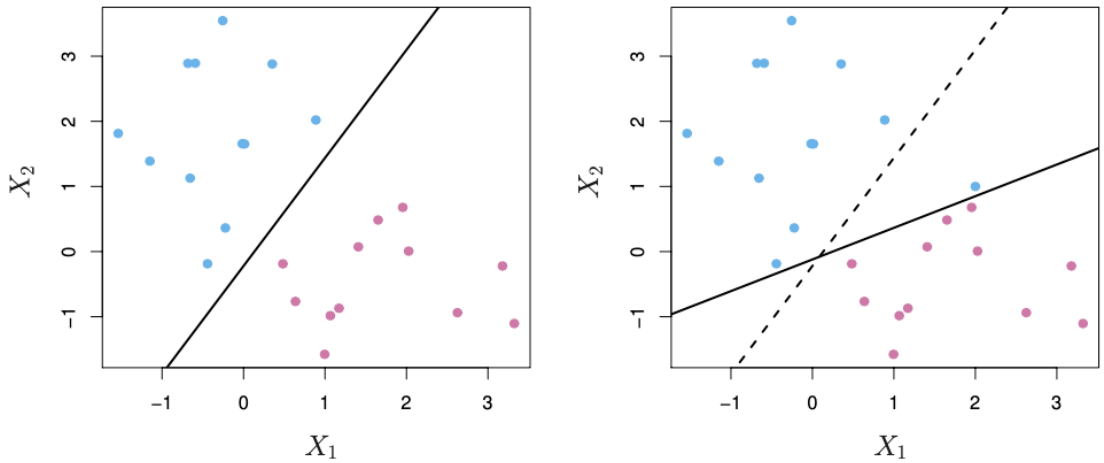
Nos vamos a encontrar varios problemas

- Datos no separables linealmente, frecuente al menos que N < p

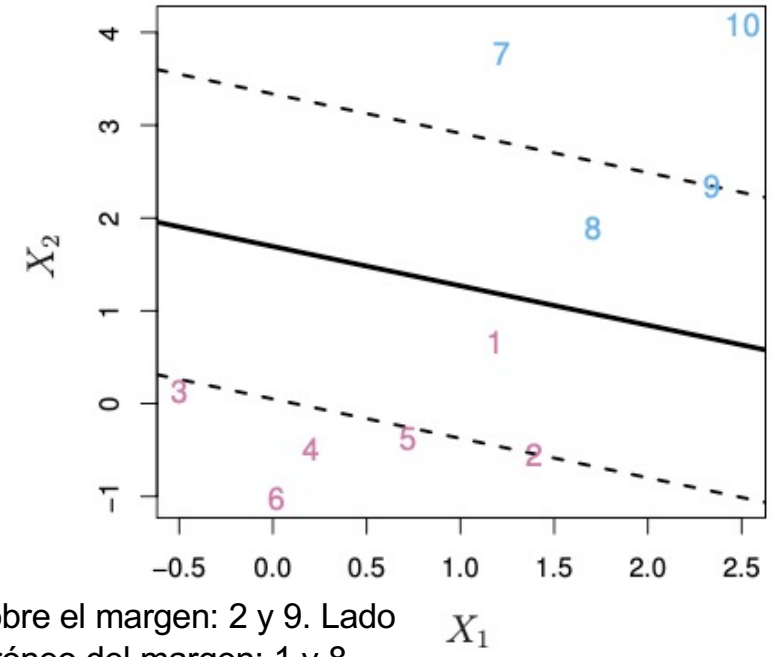


Datos ruidosos

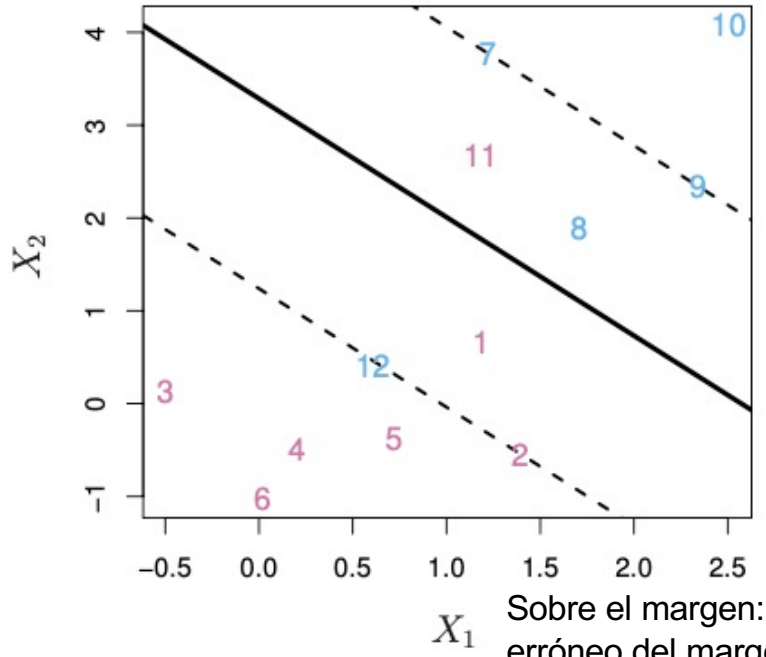
- En este caso tenemos datos separables (izq) pero con ruido (der)
- al añadir un nuevo punto, el clasificador de margen máximo se distorsiona enormemente)



Clasificadores de vector soporte



Sobre el margen: 2 y 9. Lado erróneo del margen: 1 y 8



Sobre el margen: 2, 7 y 9. Lado erróneo del margen: 1 y 8. Lado erróneo del hiperplano: 11, 12

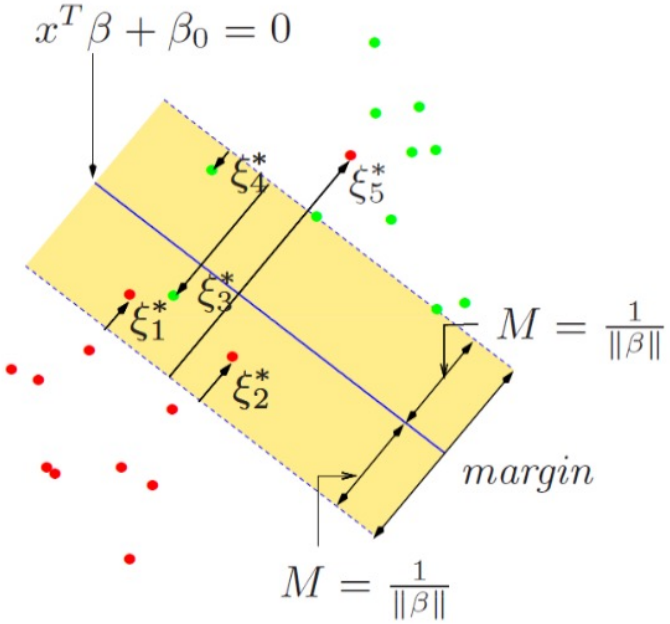
$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \; M \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \; \sum_{i=1}^{n} \epsilon_i \leq C,$$

Notas sobre el problema

- También denominado clasificador de margen blando (soft margin) porque
  - Permite que algunos ejemplos estén en el lado incorrecto del margen
  - Permite que algunos ejemplos estén en el lado incorrecto del hiperplano
  - Aumenta la robustez a ejemplos individuales ruidosos y mejora la clasificación en ejemplos de test
- Los epsilon$_i$ indican en dónde se ubica el ejemplo con respect al hiperplano y margen
  - Si epsilon$_i$ > 0 el ejemplo i está en el lado incorrecto del margen
  - Si epsilon$_i$ > 1, entonces además está en el lado incorrecto del hiperplano

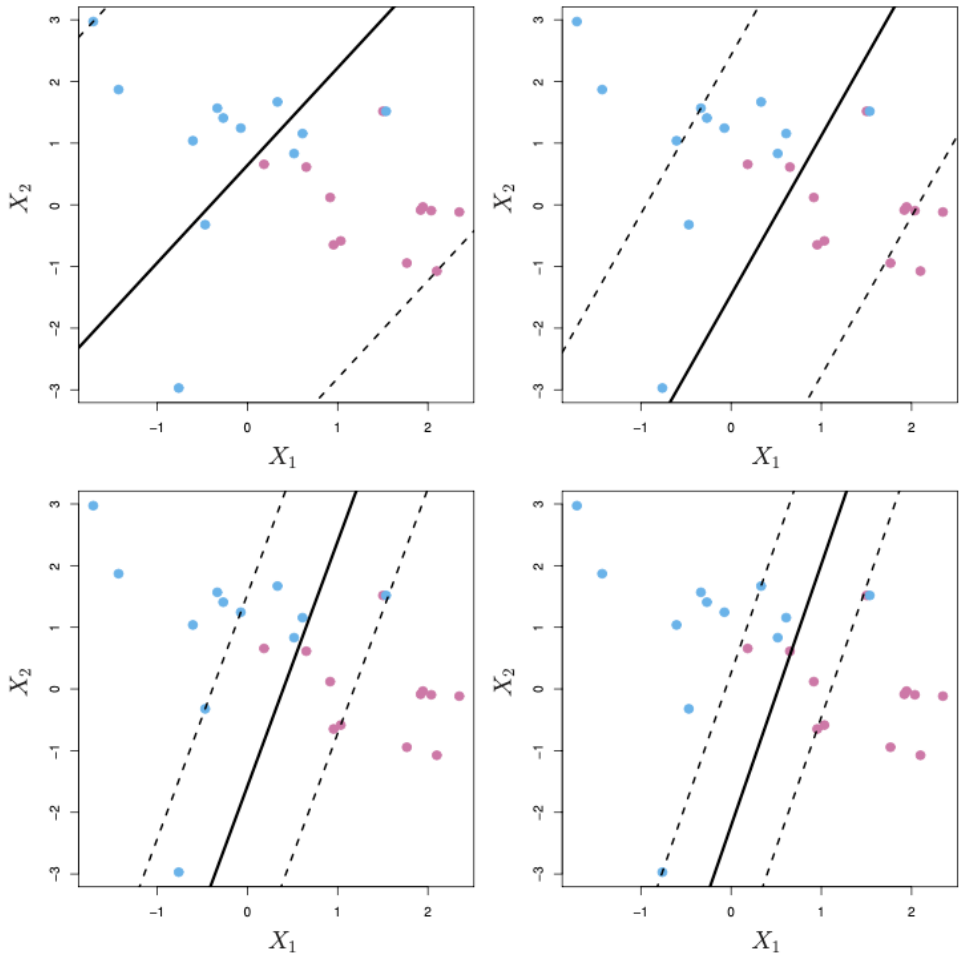


Notas sobre el problema (y II)

C puede verse como la tolerancia a incumplimientos del margen que estamos dispuestos a permitir

- C=0 reduce el problema al de margen máximo
- Si C > 0 no permitiremos más de C observaciones más allá del hiperplano (cada observación al otro lado implica un epsilon > 1)

En el ejemplo vemos de izquierda a derecha y arriba a abajo mismo dataset y valores cada vez más pequeños de C

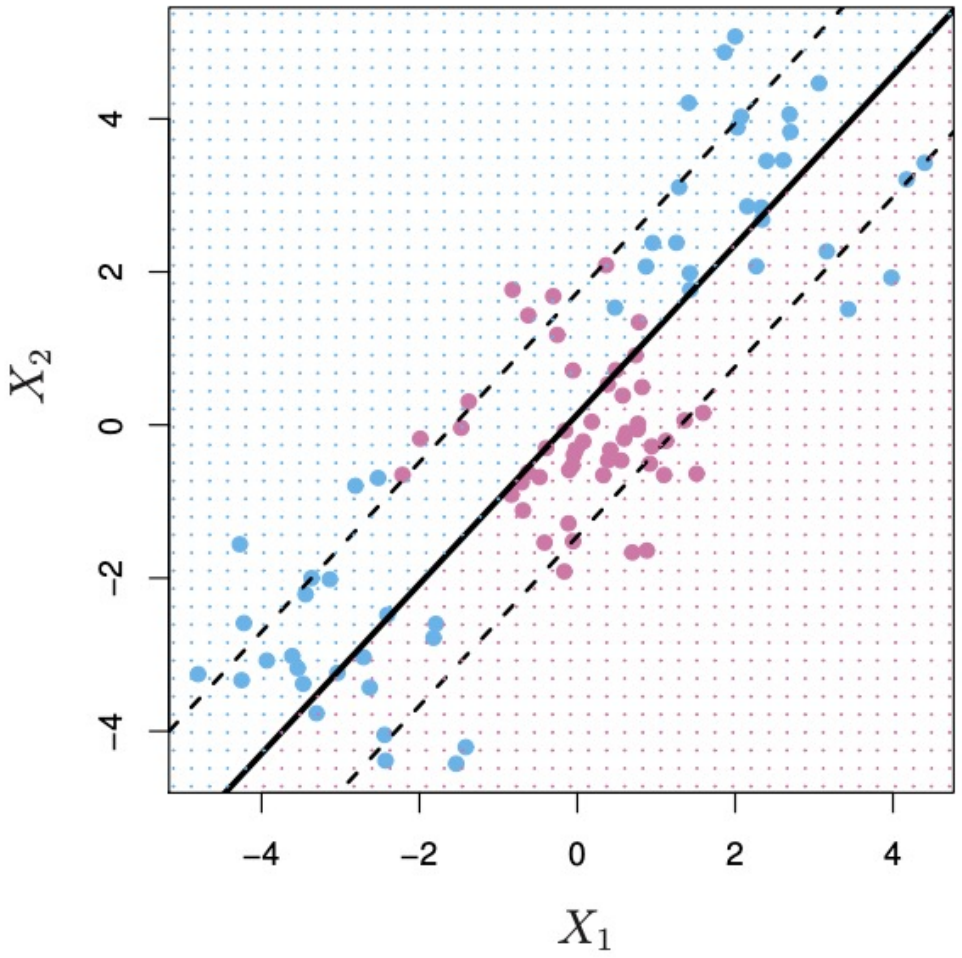


Notas sobre el problema (y III)

- El hiperparámetro C se optimiza con validación cruzada
- C controla la relación entre sesgo y varianza
  - Mayor C implica mayor sesgo, menos varianza
  - Menor C implica sesgo pequeño y variación alta de modelo a modelo
- La resolución del problema se basa en promediar las soluciones para cada vector soporte

¿Y si el problema es no lineal?

- El ejemplo muestra que no es possible aplicar un clasificador lineal, independientemente del valor de C
- Solución: buscar aproximaciones no lineales

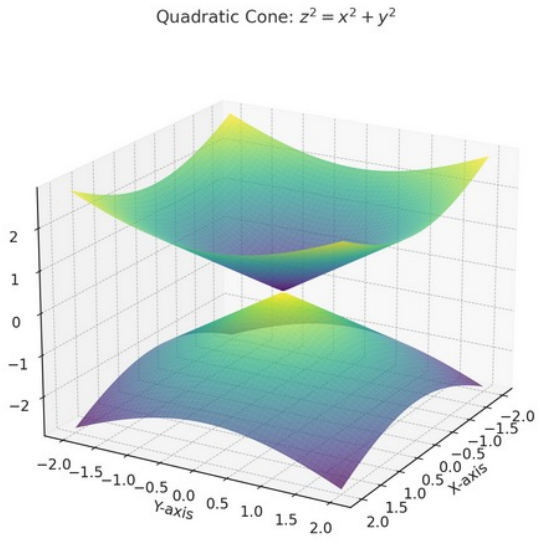


Posible solución: expandir variables

- Se trata de pasar de X_1, X_2, …, X_p a,
  - Por ejemplo X_1, X_1^2, X_2, X_2^2, …, X_p,X_p^2, a 2p variables
- El límite de decision sería ahora, para dos variables

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 = 0$$

- El espacio original sería de fronteras de decision de un cono cuadrático, como el de la figura



Con 2p variables la reformulación queda

$$\underset{\beta_0,\beta_{11},\beta_{12},\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n,\,M}{\text{maximize}} \quad M$$

$$\text{subject to } y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2 \right) \geq M(1-\epsilon_i),$$

$$\sum_{i=1}^{n} \epsilon_i \leq C, \quad \epsilon_i \geq 0, \quad \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1.$$

Si pasamos de 2 a 9 variables

$$\beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_1^2+\beta_4 X_2^2+\beta_5 X_1 X_2+\beta_6 X_1^3+\beta_7 X_2^3+\beta_8 X_1 X_2^2+\beta_9 X_1^2 X_2 = 0$$

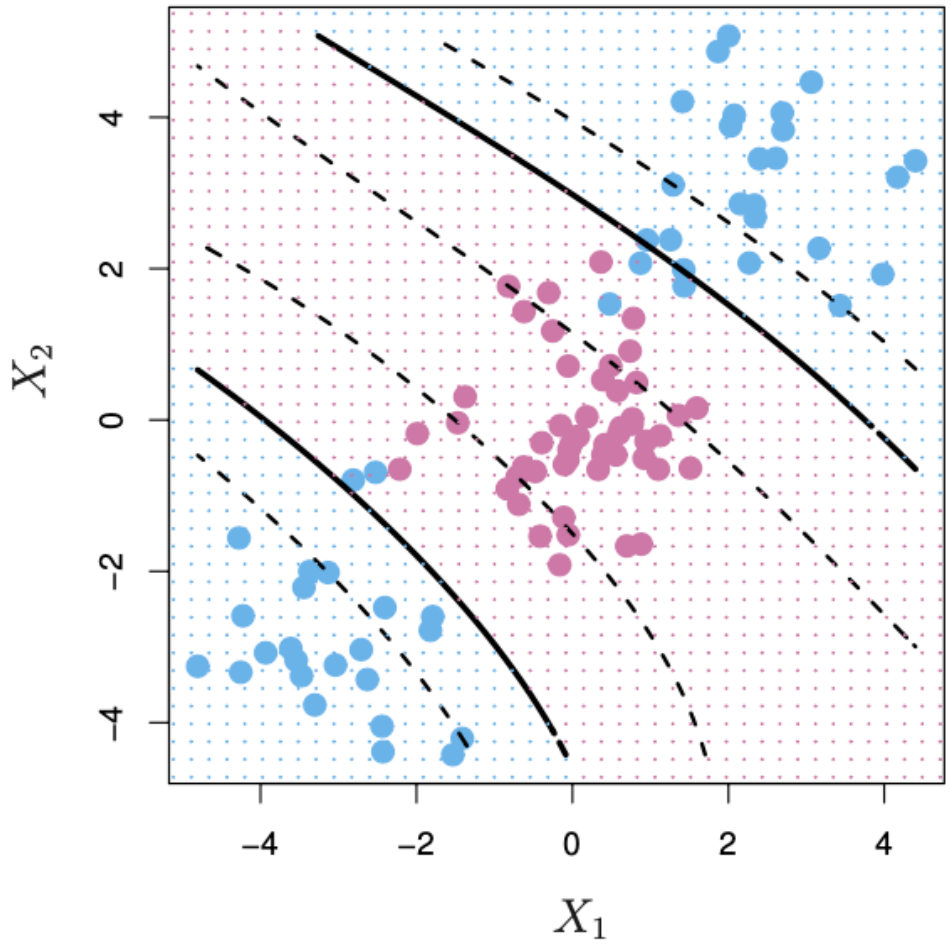
El clasificador de vector soporte en el espacio expandido se encarga entonces de resolver el problema en el espacio de menos dimensiones

Si no tenemos cuidado, el número de variables puede crecer rápidamente

- Si el problema ya tiene un p alto, aun peor!!

Necesitamos otra solución

Figura de 9 a 2 dimensiones



Kernels

- El producto interno entre vectors se define como $\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$
- El clasificador lineal de vectores de soporte se puede representar como:

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle \quad \text{(con n parámetros)}$$

- Para estimar los parámetros $\alpha_1, \ldots, \alpha_n$ y β_0, todo lo que necesitamos son los $\binom{n}{2}$ productos internos $\langle x_i, x_{i'} \rangle$ entre todos los pares de observaciones de entrenamiento.

Resulta que la mayoría de los $\hat{\alpha}_i$ pueden ser cero:

$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i \langle x, x_i \rangle$$

Donde S es el conjunto de índices de soporte i tal que $\hat{\alpha}_i > 0$. [ver diapositiva 8]

Kernels y máquinas de vectores soporte (SVMs)

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \; M \quad \text{subject to} \; \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \; \sum_{i=1}^{n} \epsilon_i \leq C,$$

- Definimos los SVM como clasificadores de vectores soporte que expanden el espacio de variables mediante kernels
- El uso de kernels es simplemente un enfoque simple y eficiente para proporcionar fronteras de decision no lineales
- Un kernel es la generalización de la operación de product interno y cuantifica la similitude entre dos ejemplos
- Para un SVM, la frontera de separación es $f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \hat{\alpha}_i K(x, x_i).$
- Ejemplos

Kernel polynomial (calcula productos internos para polinomios d-dimensionales)

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d$$

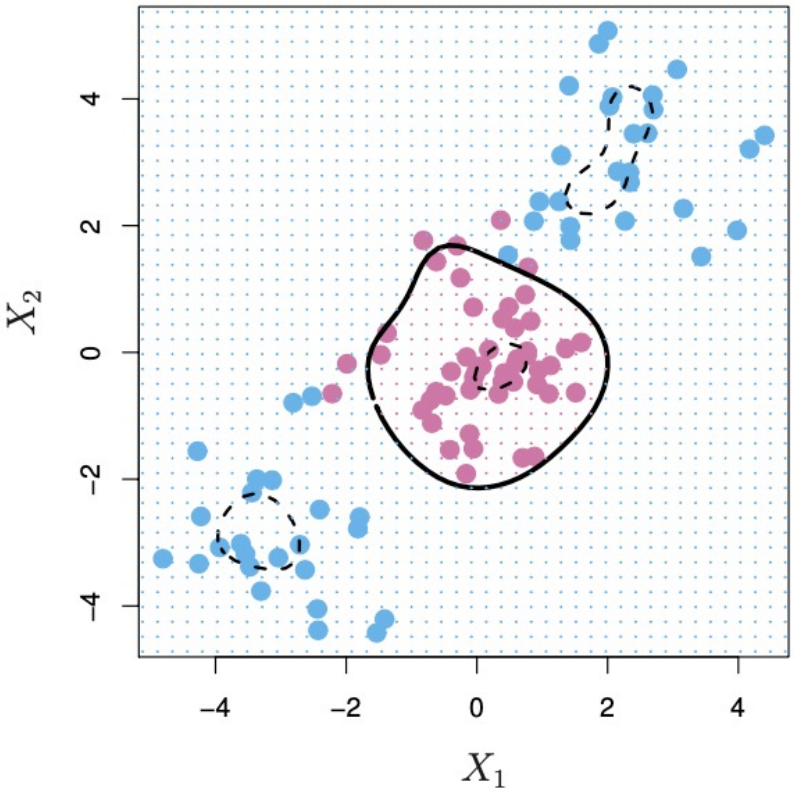
Kernel radial

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$

Kernel radial

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$

- Gamma es una constante positiva
  - Si la distancia euclidea entre xi y xi' es grande, la similitud radial va a ser diminuta: xi no cuenta en la predicción hecha por f(x)
    - Solo observaciones muy cercanas son tenidas en cuenta (comportamiento muy local)

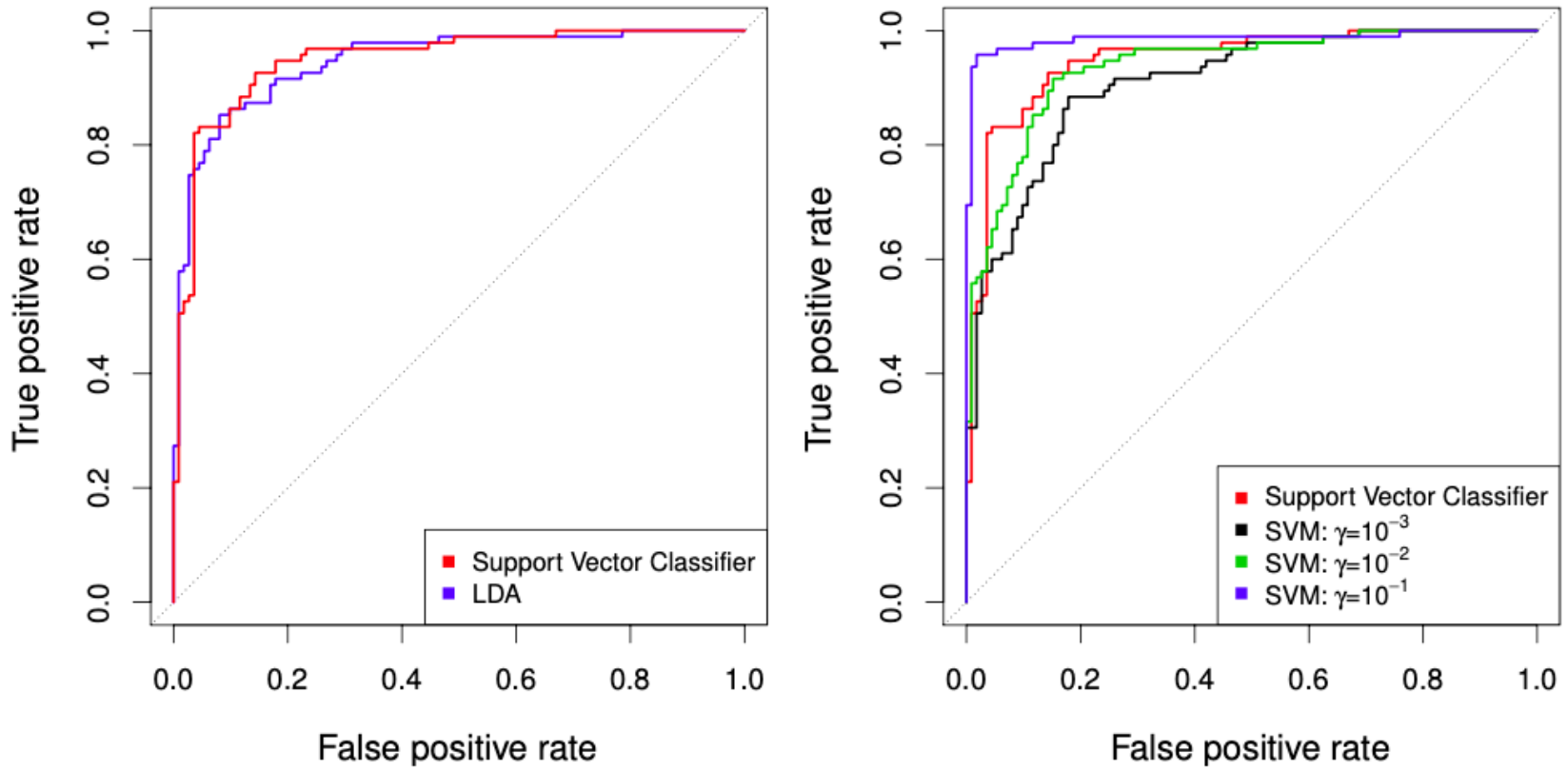


SVM para K > 2

El **SVM** tal como está definido funciona para $K = 2$ clases. ¿Qué hacemos si tenemos $K > 2$ clases?

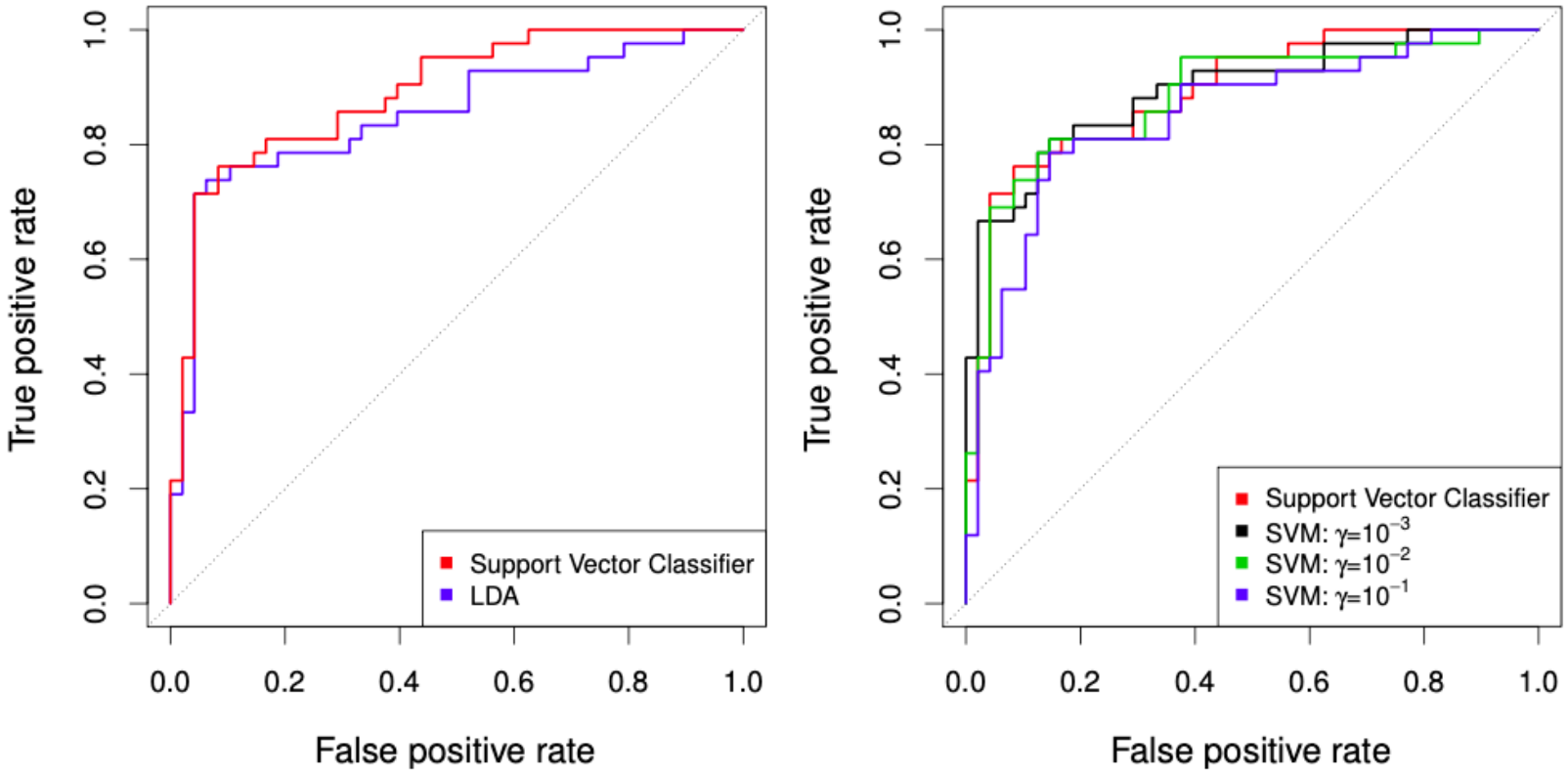
- **OVA (One versus All - Uno contra Todos)**:
  Ajustamos K clasificadores SVM diferentes de 2 clases $\hat{f}_k(x)$, $k = 1, \ldots, K$, cada clase contra el resto.
  Clasificamos x^* en la clase para la cual $\hat{f}_k(x^*)$ es la mayor.
- **OVO (One versus One - Uno contra Uno)**:
  Ajustamos todos los clasificadores por pares $\hat{f}_{k\ell}(x)$.
  Clasificamos x^* en la clase que gana la mayor cantidad de competiciones por pares.
- **¿Cuál elegir?**
  Si K no es demasiado grande, usa **OVO**.

SVM aplicado a heart



Curvas ROC en los datos de training. LDA equivale a regression logística.
SVC equivale a SVM polynomial con d=1
Gamma más alto equivale a comportamiento más no-lineal

SVM aplicado a heart (cont.)



Curvas ROC en test. Ligero overfitting de SVM con gamma alto
El mejor kernel depende del problema

References

UNIVERSIDAD DE MURCIA

- G. James, D. Witten, T. Hastie, y R. Tibshirani, An Introduction to Statistical Learning with Applications in R. Springer, 2013.
  - Chapter 9