



# MÁQUINAS DE SOPORTE VECTORIAL

- **SVM (*Support Vector Machines*)**
  - Técnica de clasificación
  - Se derivan de la teoría de aprendizaje estadístico postulada por Vapnik y Chervonenkis
  - Fueron presentadas en 1992, y recibieron mucho interés en los años recientes por:
    - Muy alta precisión en la clasificación
    - Base matemática muy firme
    - En muchos campos han demostrado superar en resultados a los sistemas de más éxito (RR.NN.AA.)
  - Clasificador en dos clases
    - Generalizable a más clases
    - Ampliable para problemas de regresión



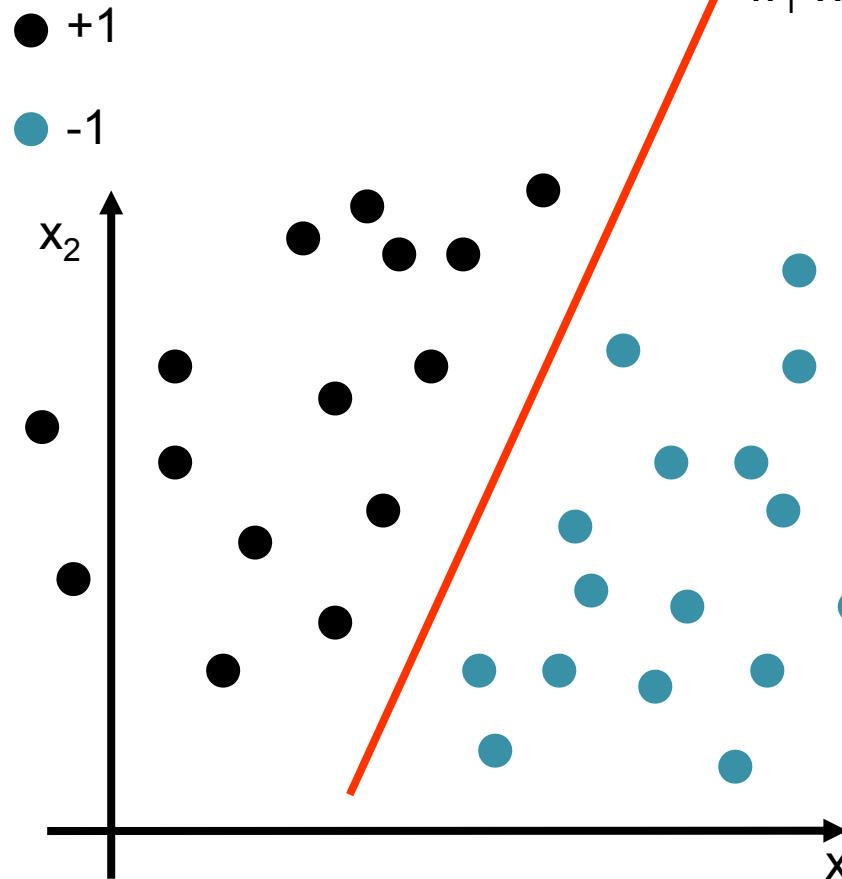
# MÁQUINAS DE SOPORTE VECTORIAL

- Se tiene una serie de observaciones, cada una consiste en un par de datos:
  - Un vector  $x_i \in R^n, i = 1, \dots, l$
  - Una etiqueta  $y_i \in \{+1, -1\}$
- Supóngase que se tiene un hiperplano que separa las muestras positivas (+1) de las negativas (-1).
  - Problemas linealmente separables
  - Ejemplo, en  $R^2$ :

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Ejemplo, en  $\mathbb{R}^2$ :



(en esta imagen los subíndices de  $x_1$  y  $x_2$  denotan la dimensión, no que sean los patrones 1 y 2)

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$w^T x + b = 0$$



Todos los patrones en el hiperplano cumplen:  
 $w \cdot x + b = 0$

Todos los patrones ● cumplen:  
 $w \cdot x + b > 0$

Todos los patrones ● cumplen:  
 $w \cdot x + b < 0$

# MÁQUINAS DE SOPORTE VECTORIAL

- Se tiene una serie de observaciones, cada una consiste en un par de datos:
  - Un vector  $x_i \in R^n, i = 1, \dots, l$
  - Una etiqueta  $y_i \in \{+1, -1\}$
- Supóngase que se tiene un hiperplano que separa las muestras positivas (+1) de las negativas (-1).
  - **Problemas linealmente separables**
  - Hiperplano, caso general:  $w^T x + b = 0$ 
    - Los puntos  $x_i$  que están en el hiperplano satisfacen  $w^T x + b = 0$
    - Los puntos  $x_i$  que están «a un lado» del hiperplano satisfacen  $w^T x + b > 0$
    - Los puntos  $x_i$  que están «al otro lado» del hiperplano satisfacen  $w^T x + b < 0$



# MÁQUINAS DE SOPORTE VECTORIAL

- En el mundo real, la gran mayoría de los problemas no son linealmente separables
- Estudio de los SVM en tres casos distintos:
  - Problemas linealmente separables
    - SVM lineales
    - Región de separación: hiperplano
  - Problemas no linealmente separables
    - SVM lineales
    - Región de separación: hiperplano
    - Se permite cierto error en la clasificación
  - SVM no lineales
    - Aún admitiendo cierto error, no se puede separar los datos
    - Regiones de separación más complejas

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Se desea encontrar el hiperplano

$$w \cdot x + b = 0$$

definido por el par  $(w, b)$ , tal que se pueda separar el punto  $x_i$  de acuerdo a la función

$$f(x_i) = \begin{cases} 1 & w \cdot x_i + b > 0 \\ -1 & w \cdot x_i + b < 0 \end{cases} \quad \begin{matrix} y_i=1 \\ y_i=-1 \end{matrix}$$

es decir

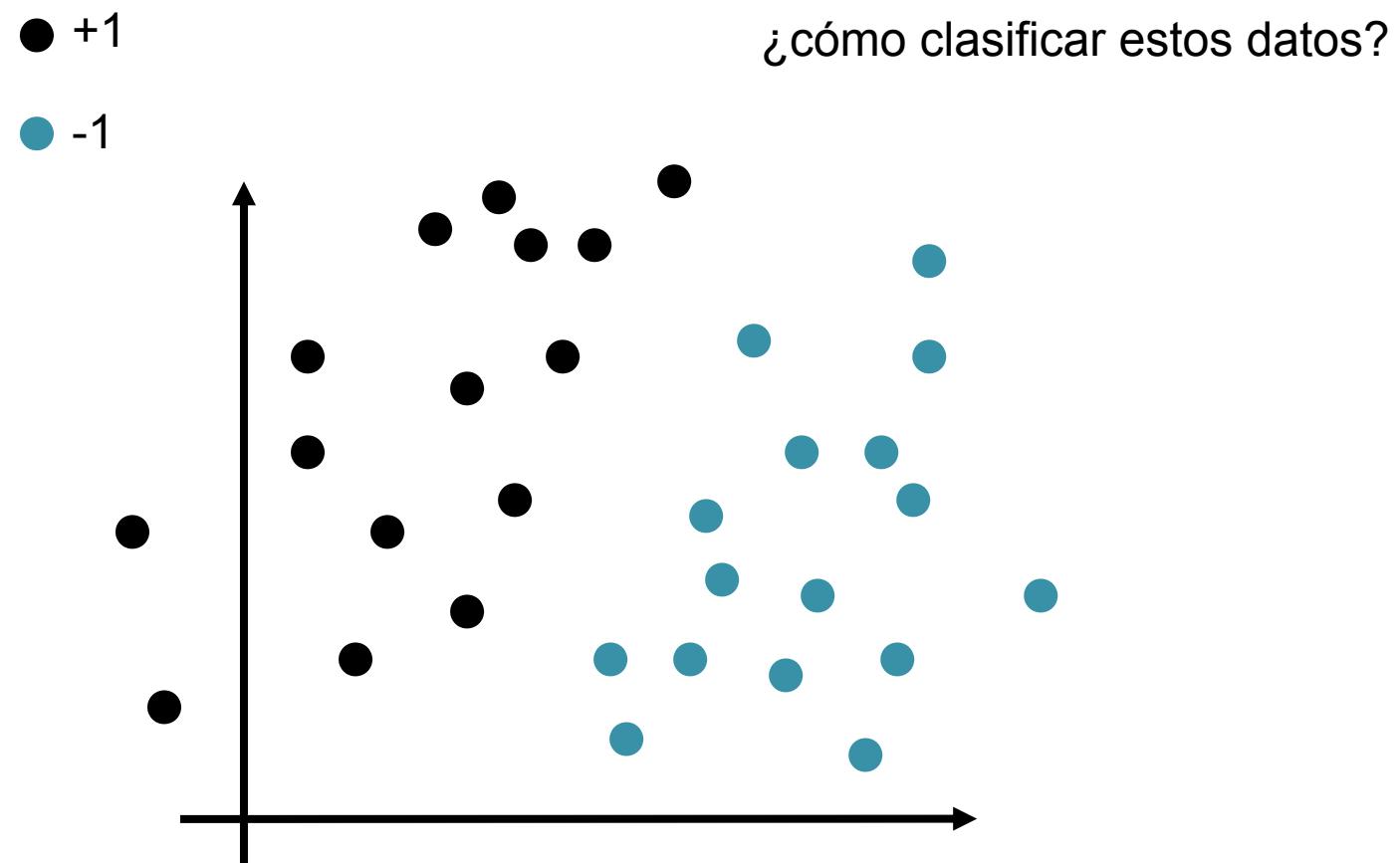
$$f(x_i) = sign(w \cdot x_i + b) = \begin{cases} 1 & y_i=1 \\ -1 & y_i=-1 \end{cases}$$

donde  $x \in \Re^N$  y  $b \in \Re$

- Es decir, la función  $f$  clasifica cada punto en  $+1$  o  $-1$

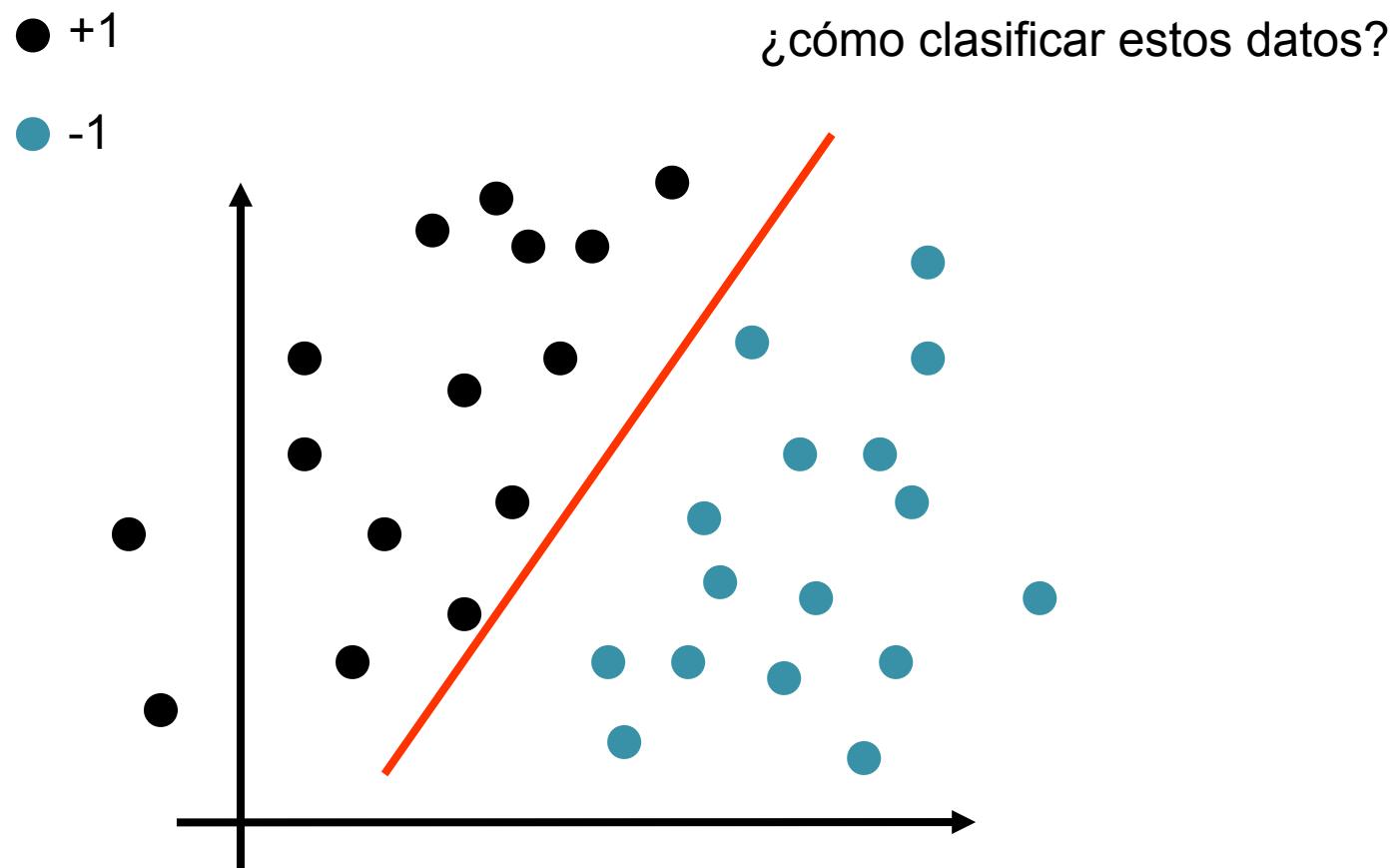
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



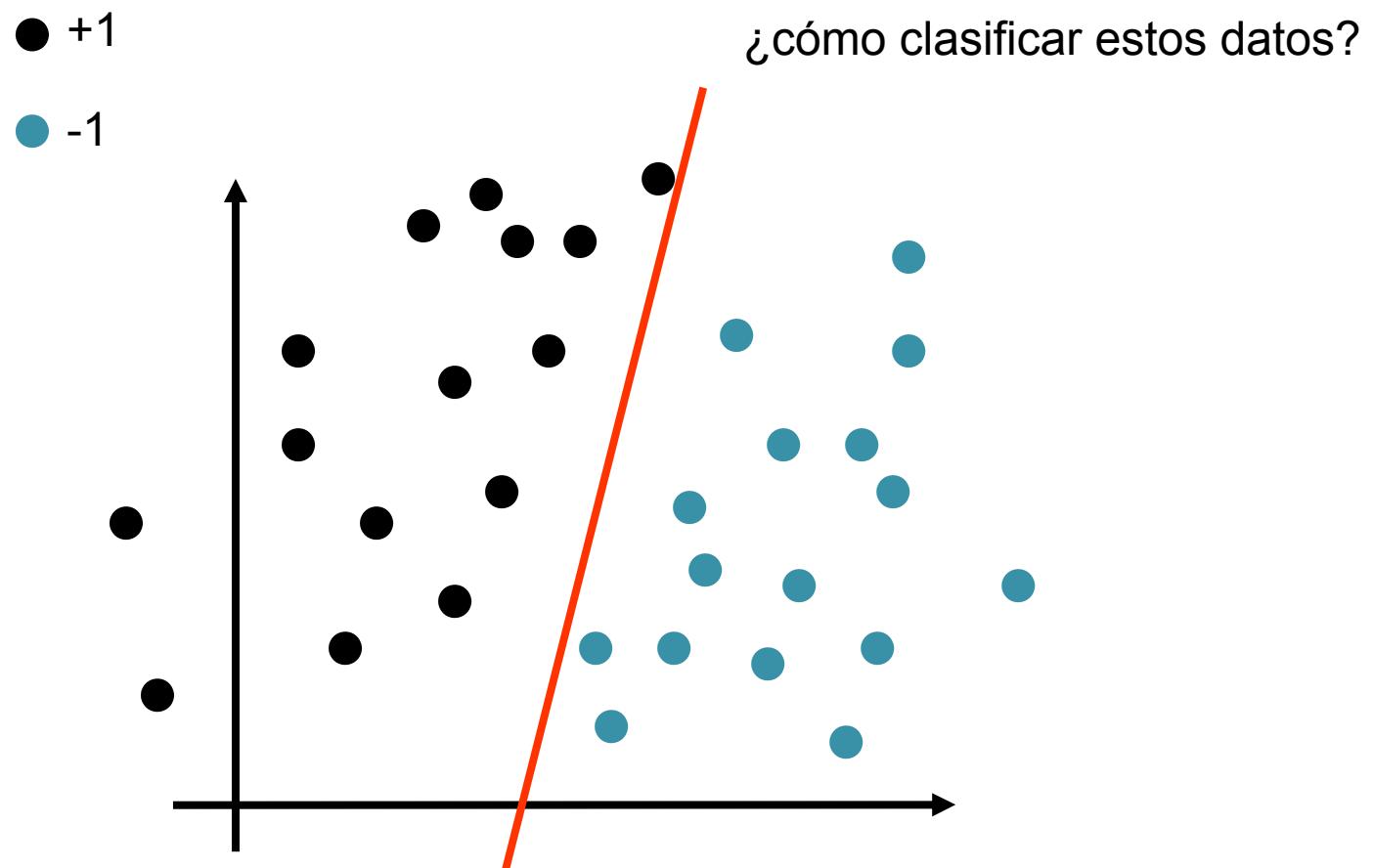
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



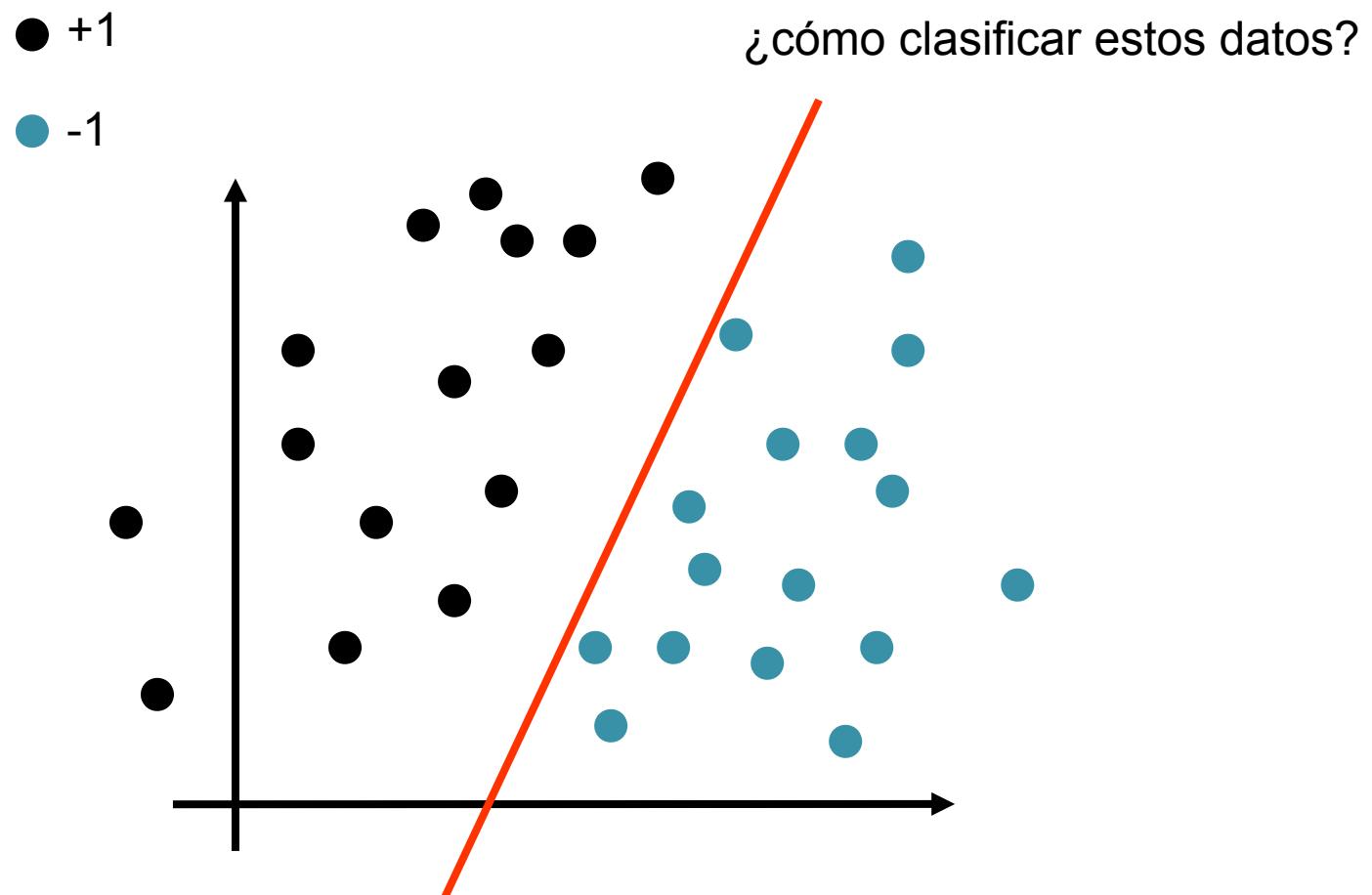
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



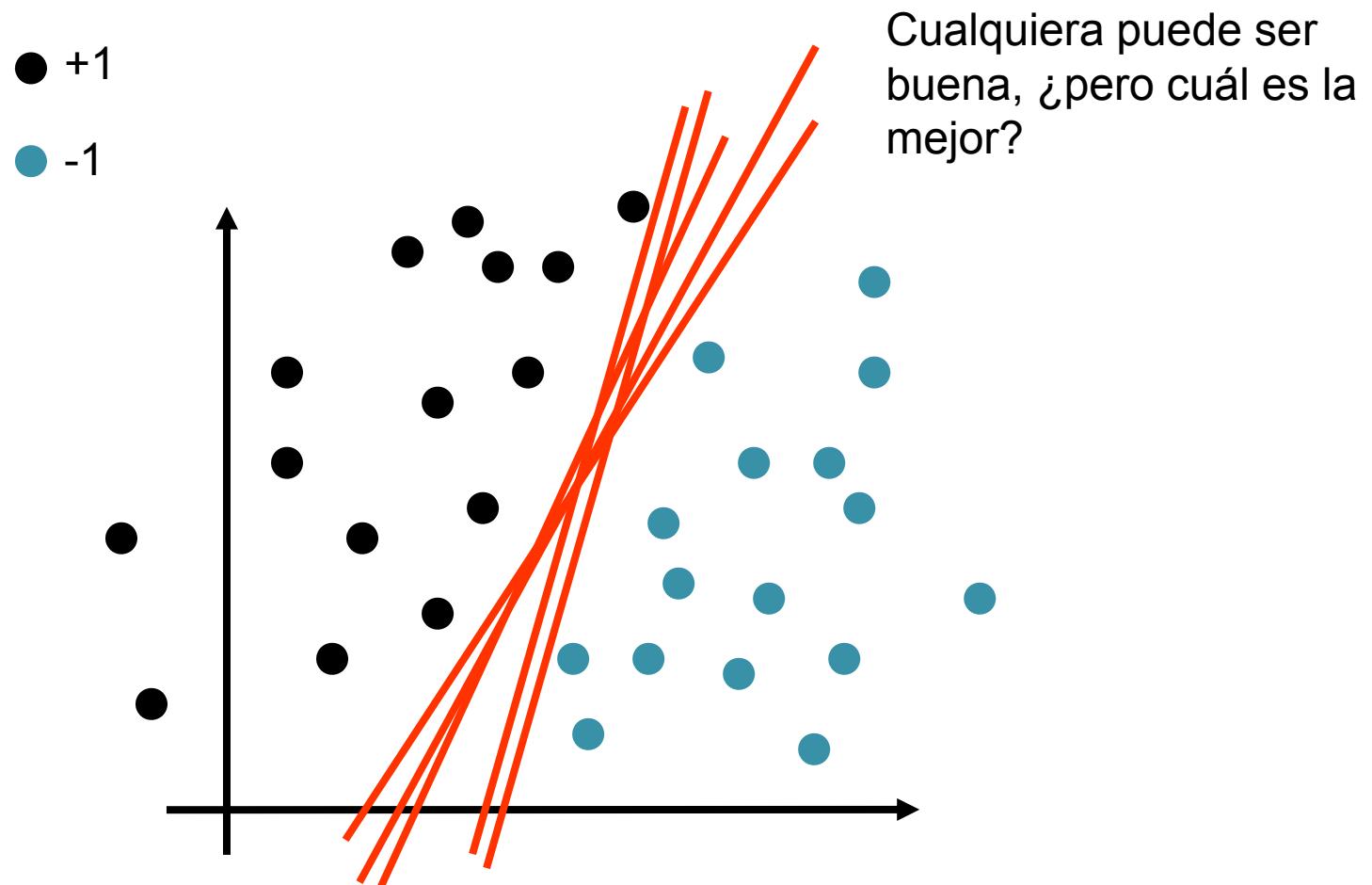
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



# MÁQUINAS DE SOPORTE VECTORIAL

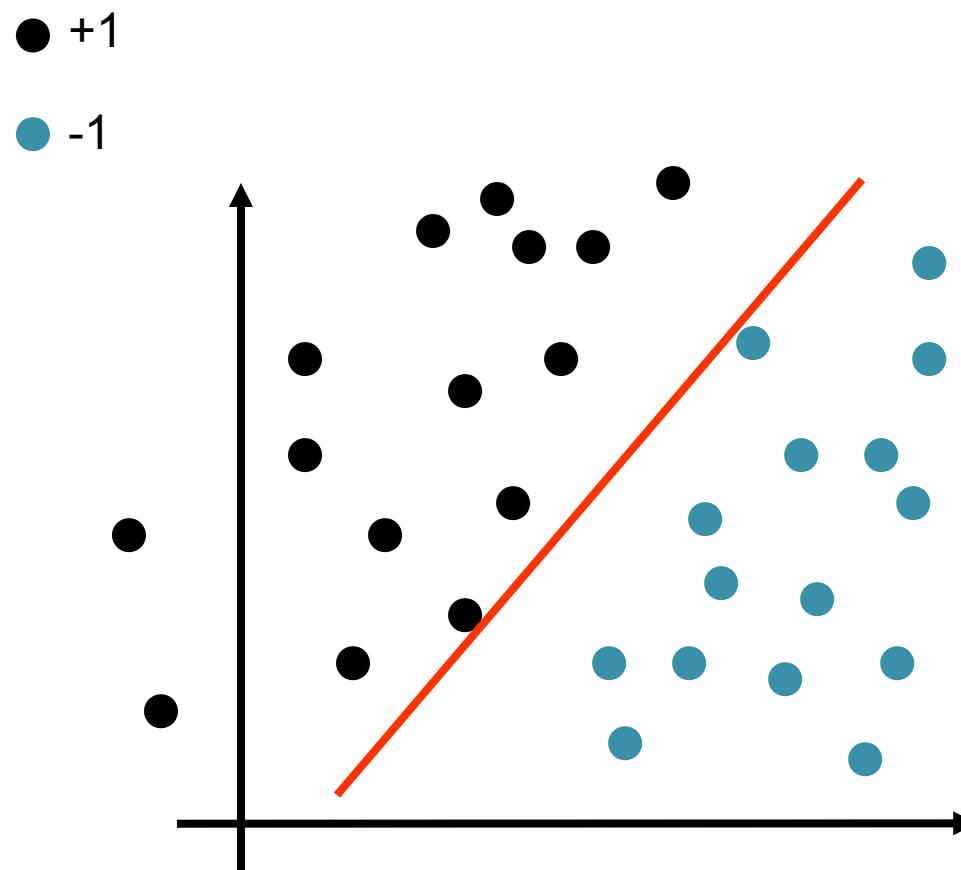
- Problemas linealmente separables



# MÁQUINAS DE SOPORTE VECTORIAL

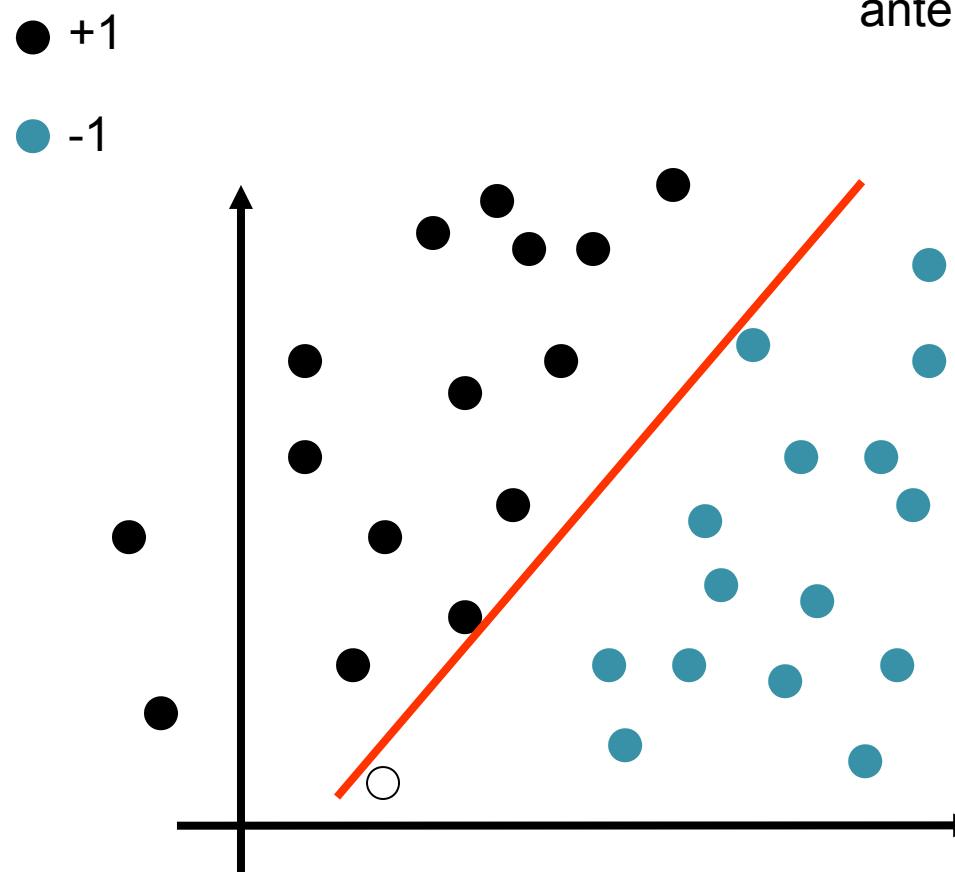
- Problemas linealmente separables

Si se escoge este hiperplano



# MÁQUINAS DE SOPORTE VECTORIAL

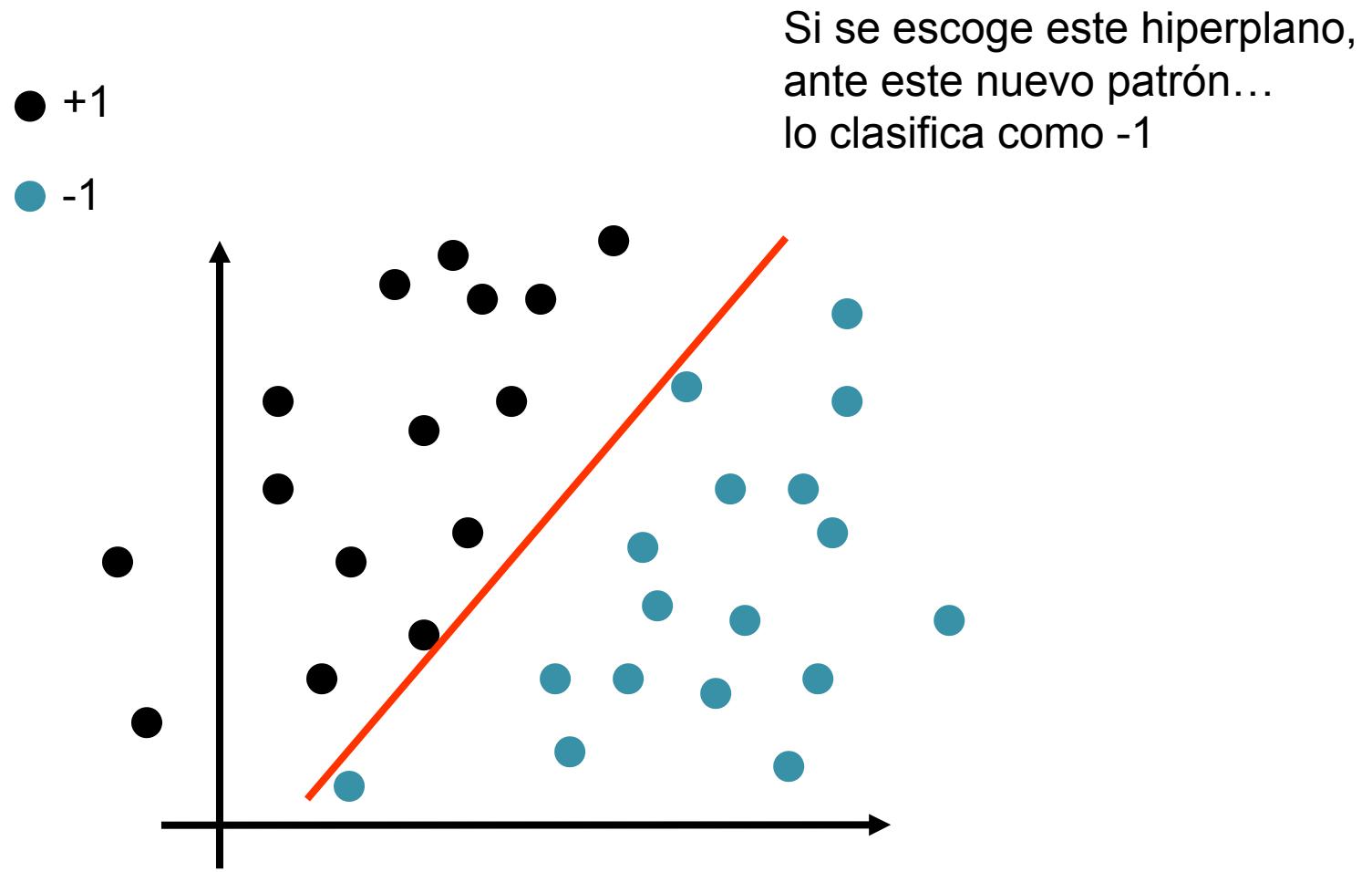
- Problemas linealmente separables



Si se escoge este hiperplano,  
ante este nuevo patrón...

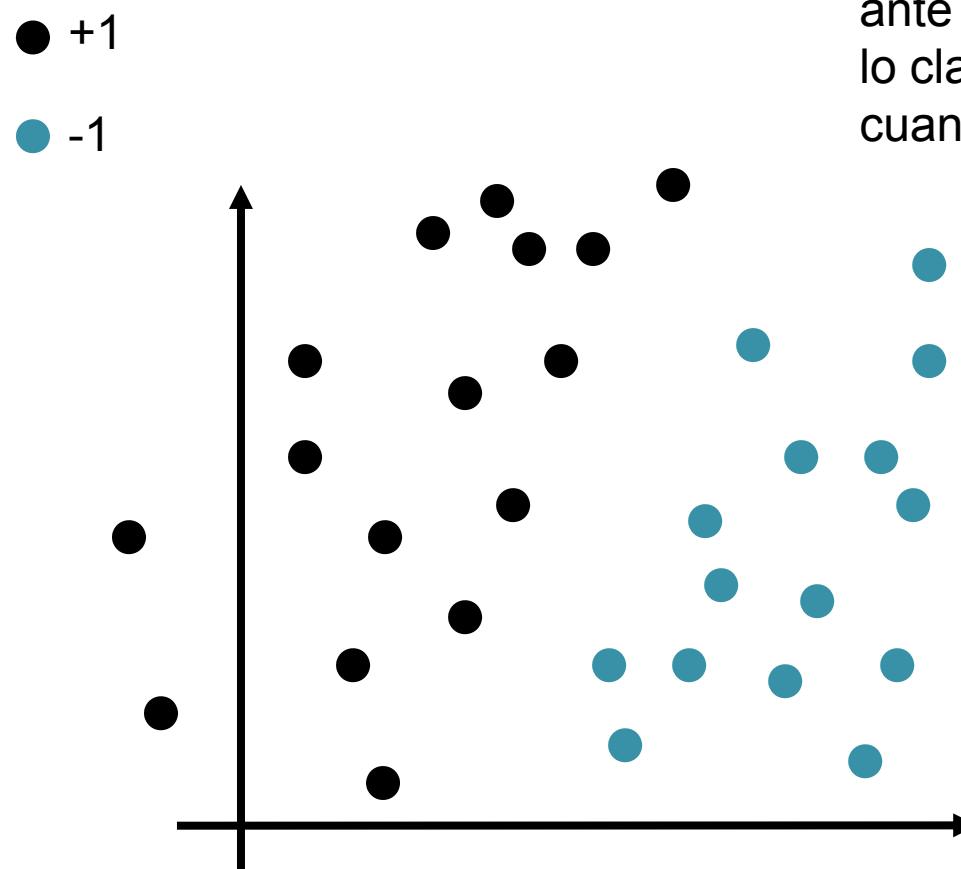
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



# MÁQUINAS DE SOPORTE VECTORIAL

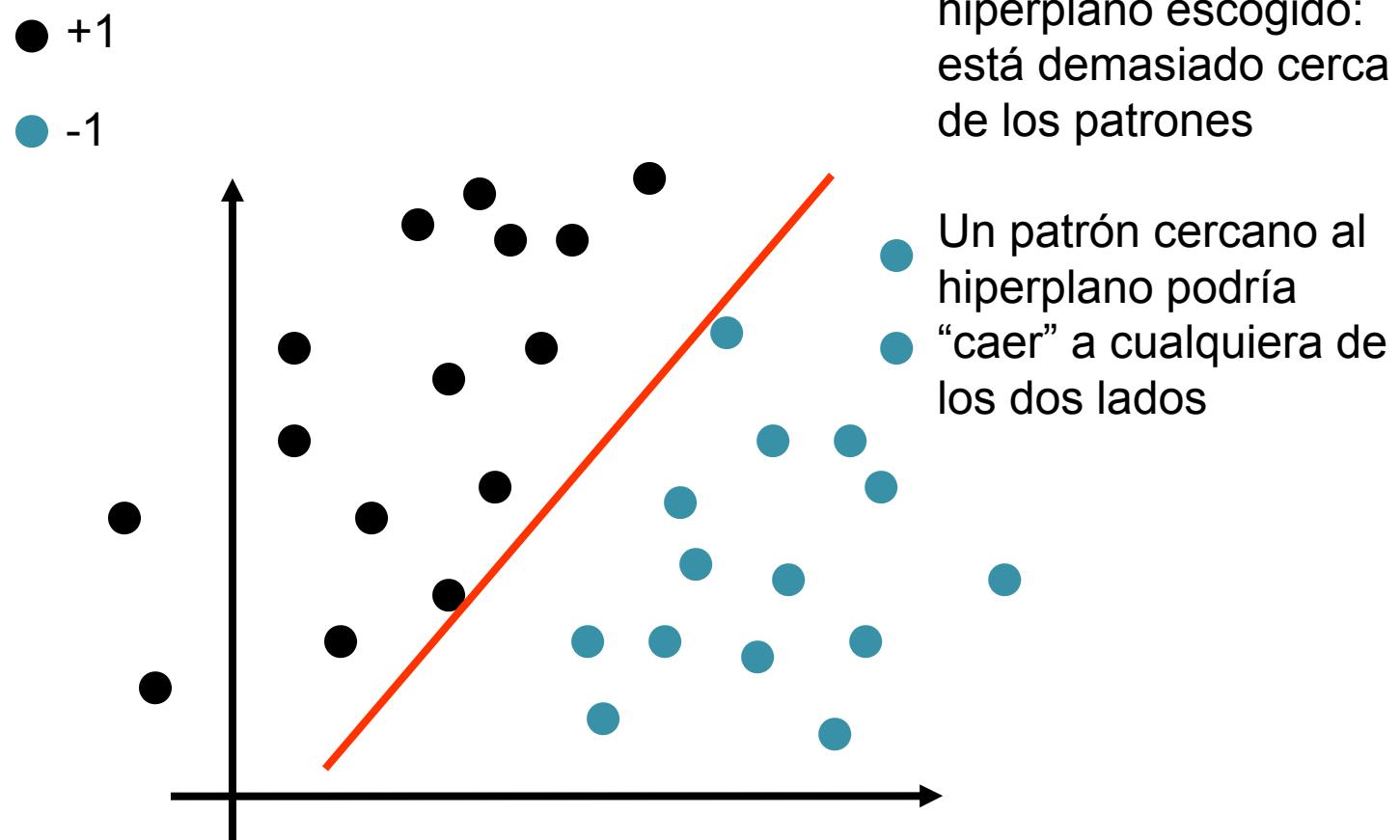
- Problemas linealmente separables



Si se escoge este hiperplano,  
ante este nuevo patrón...  
lo clasifica como -1  
cuando debería ser +1

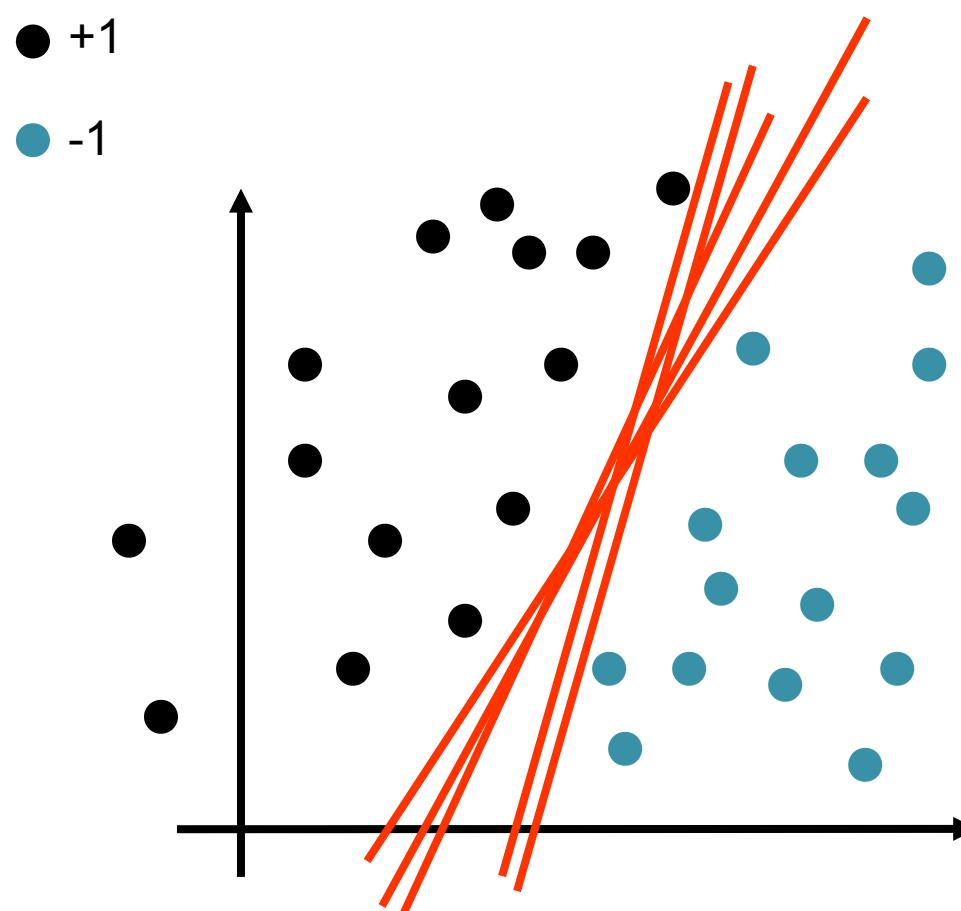
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



# MÁQUINAS DE SOPORTE VECTORIAL

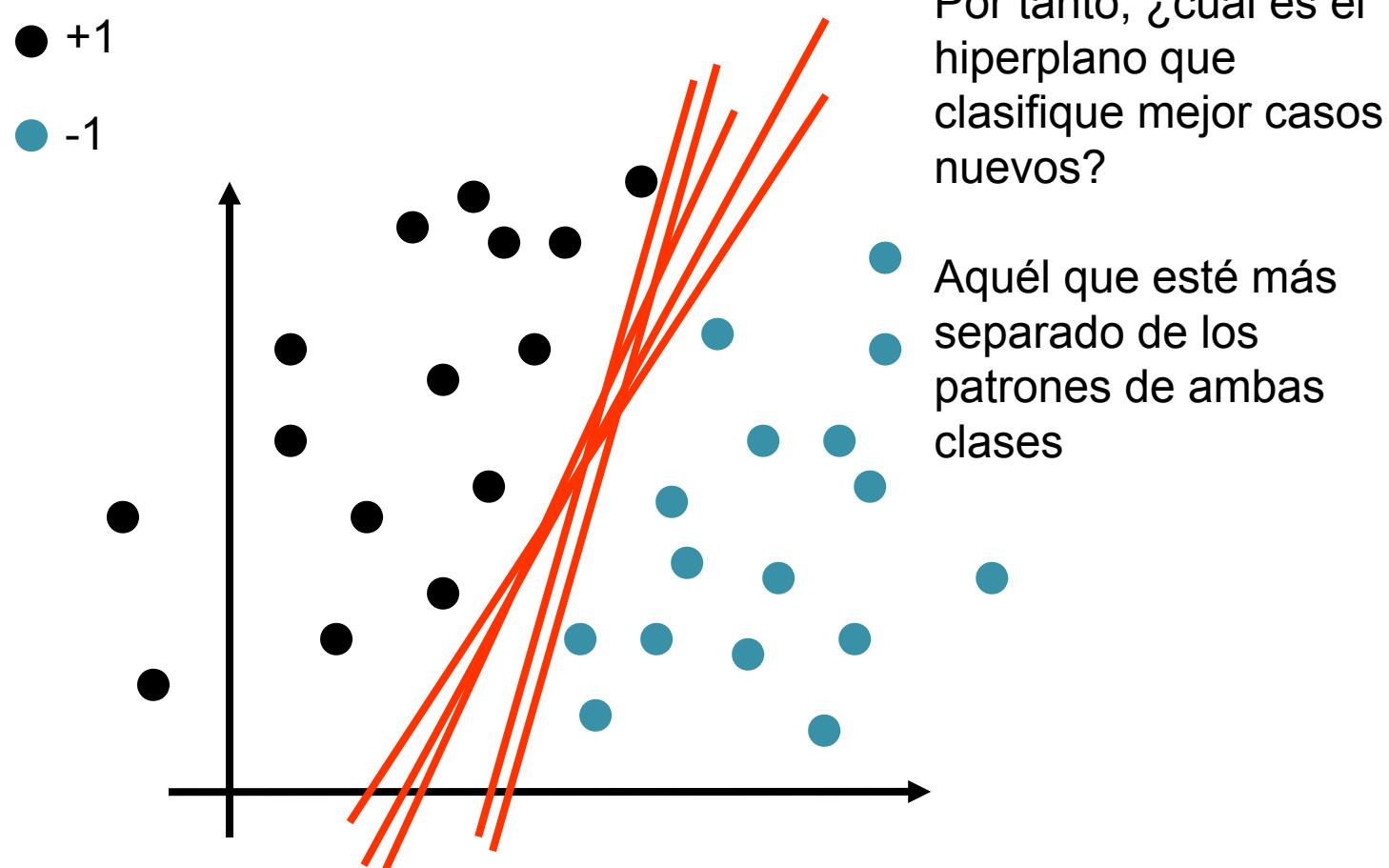
- Problemas linealmente separables



Por tanto, ¿cuál es el hiperplano que clasifique mejor casos nuevos?

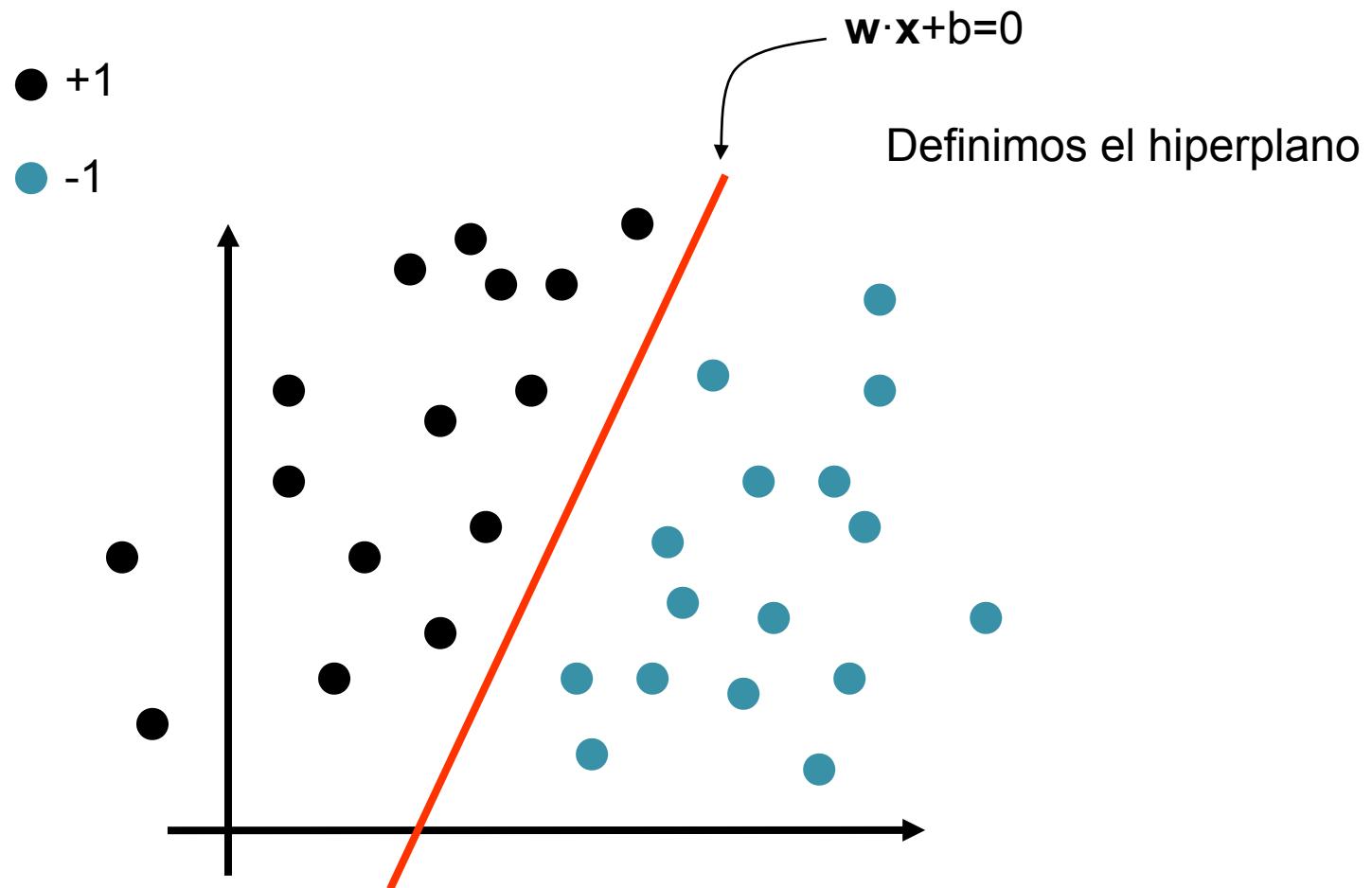
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



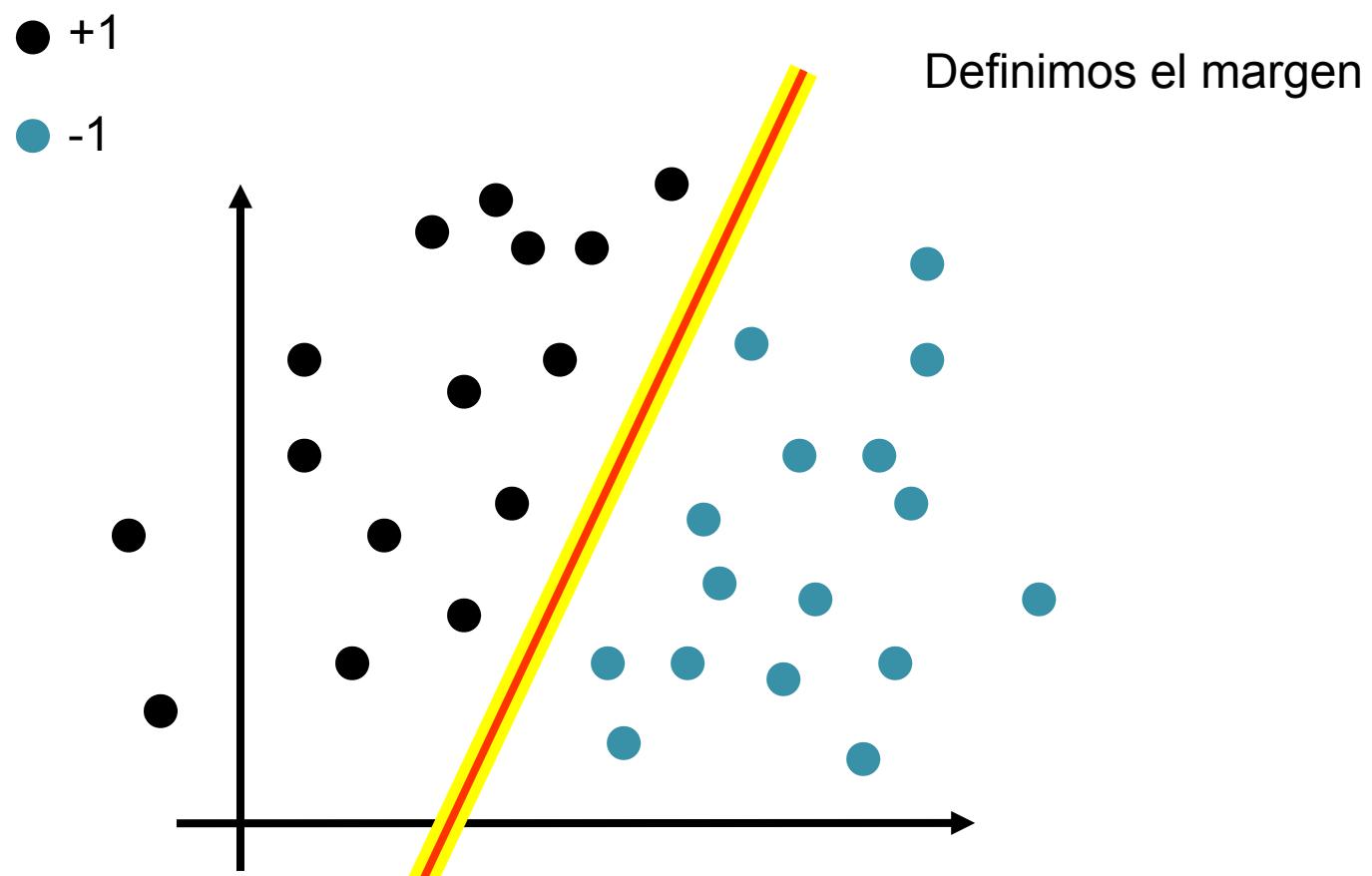
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



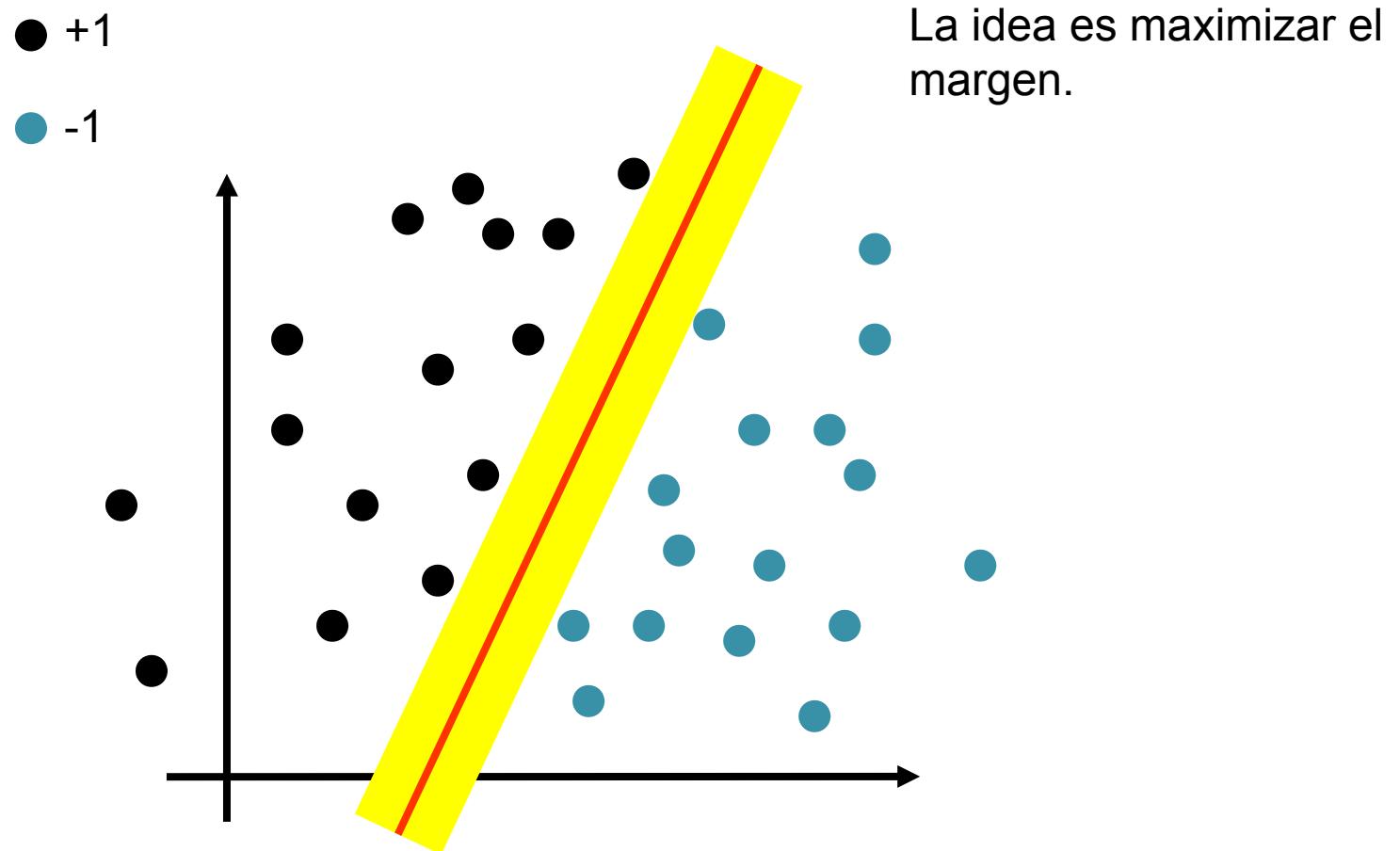
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



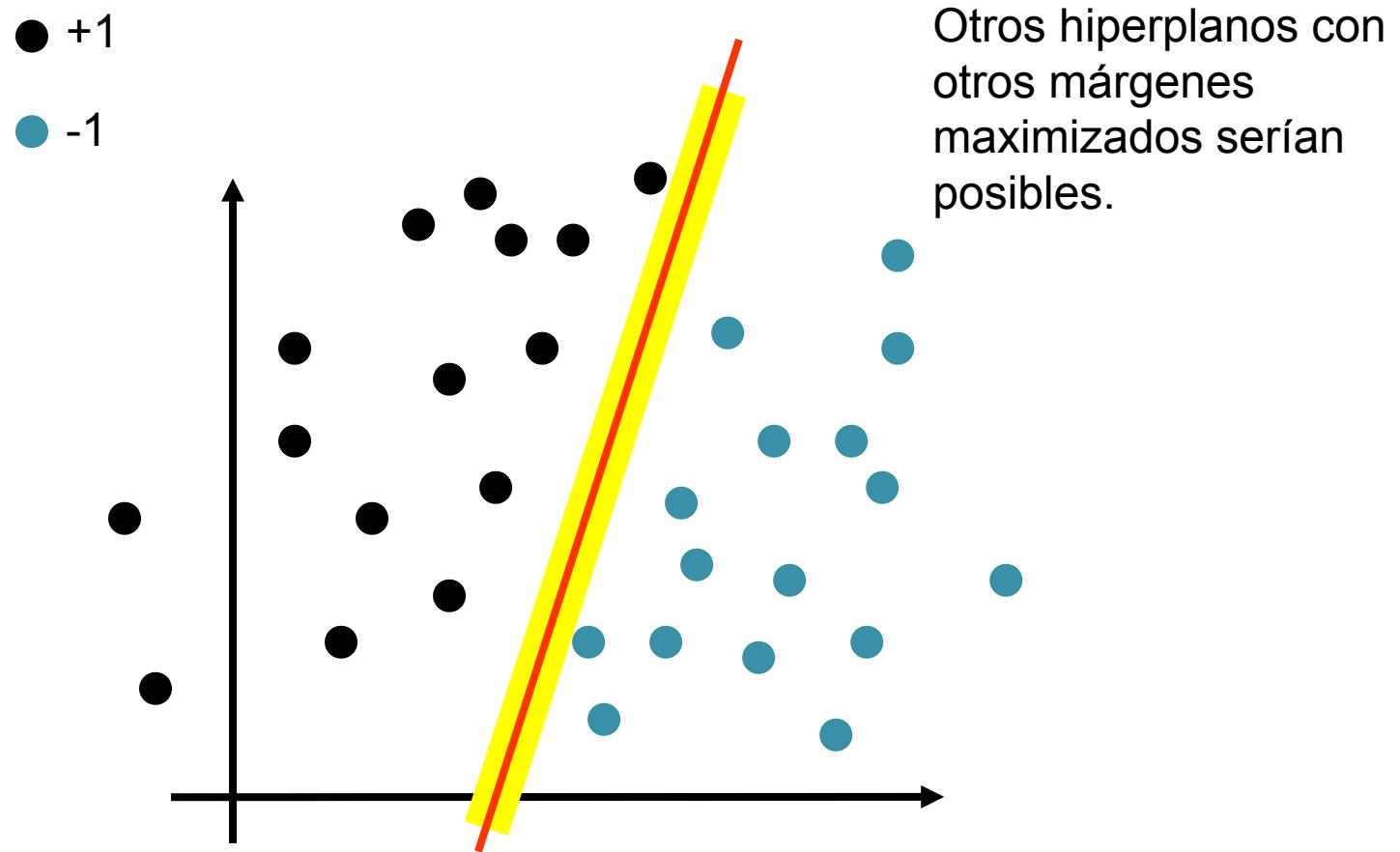
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



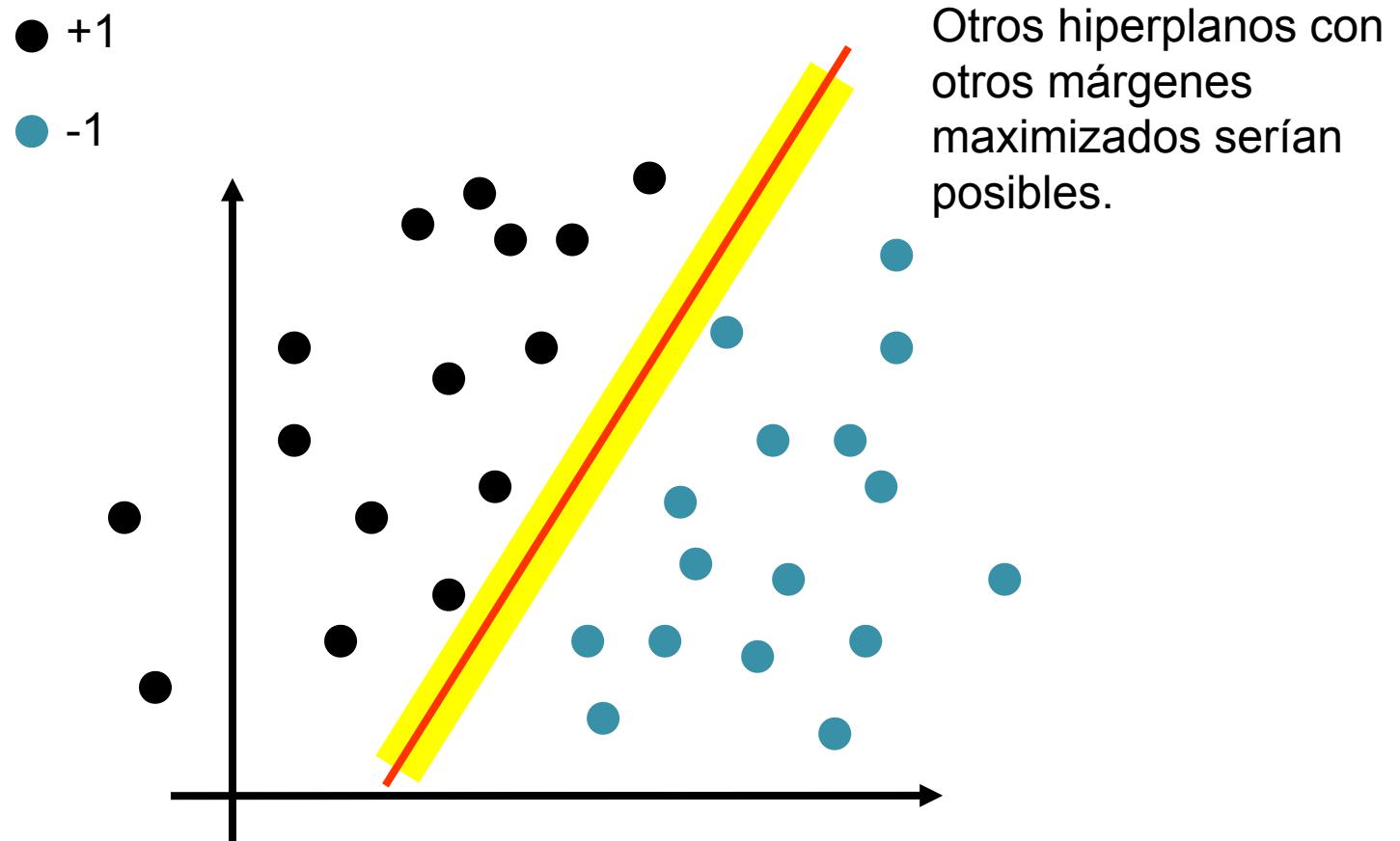
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



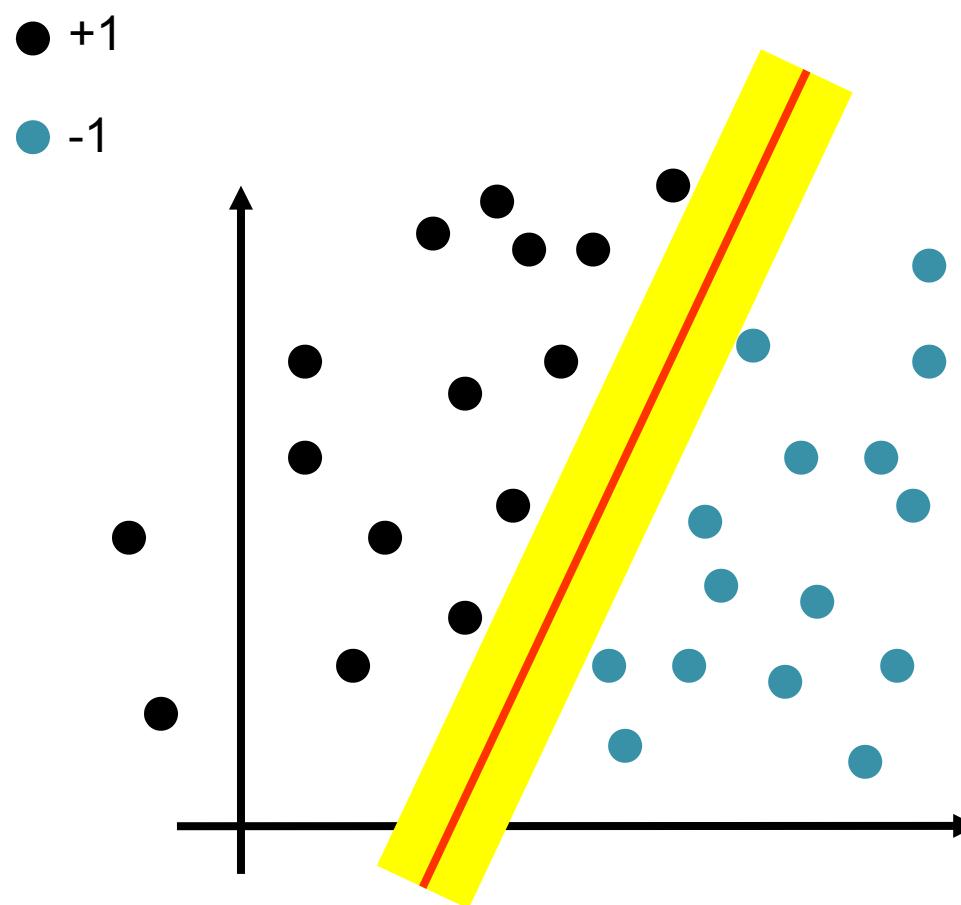
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

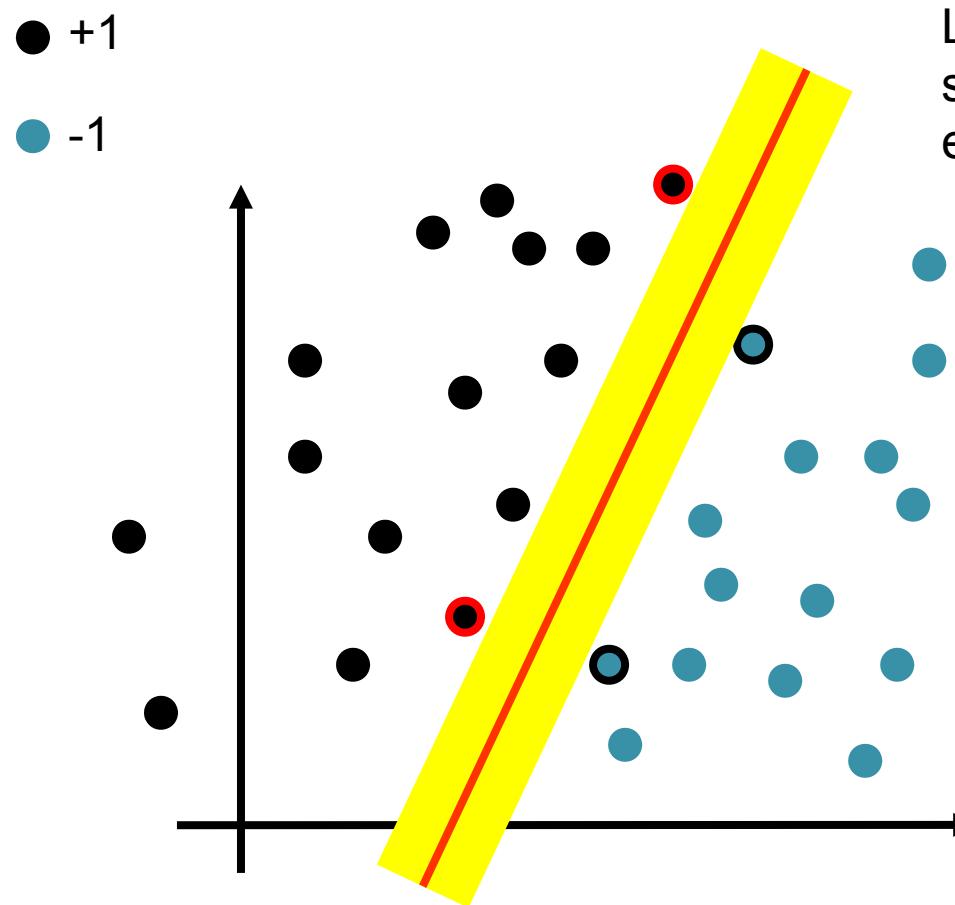


El hiperplano que tenga el mayor margen es el mejor clasificador de los datos.

Esta es la clase más simple de SVM, la LSVM.

# MÁQUINAS DE SOPORTE VECTORIAL

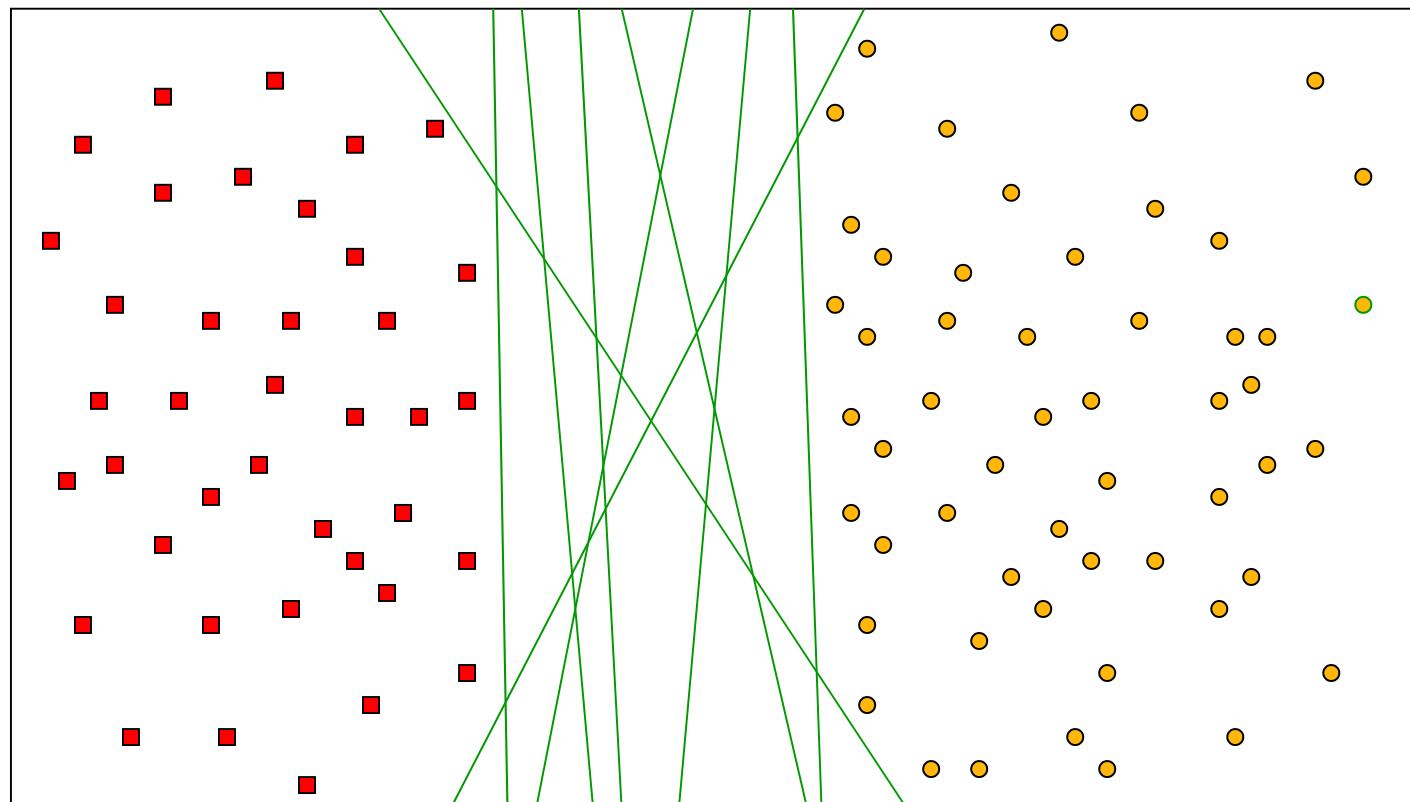
- Problemas linealmente separables



Los vectores de soporte son los puntos que tocan el límite del margen.

# MÁQUINAS DE SOPORTE VECTORIAL

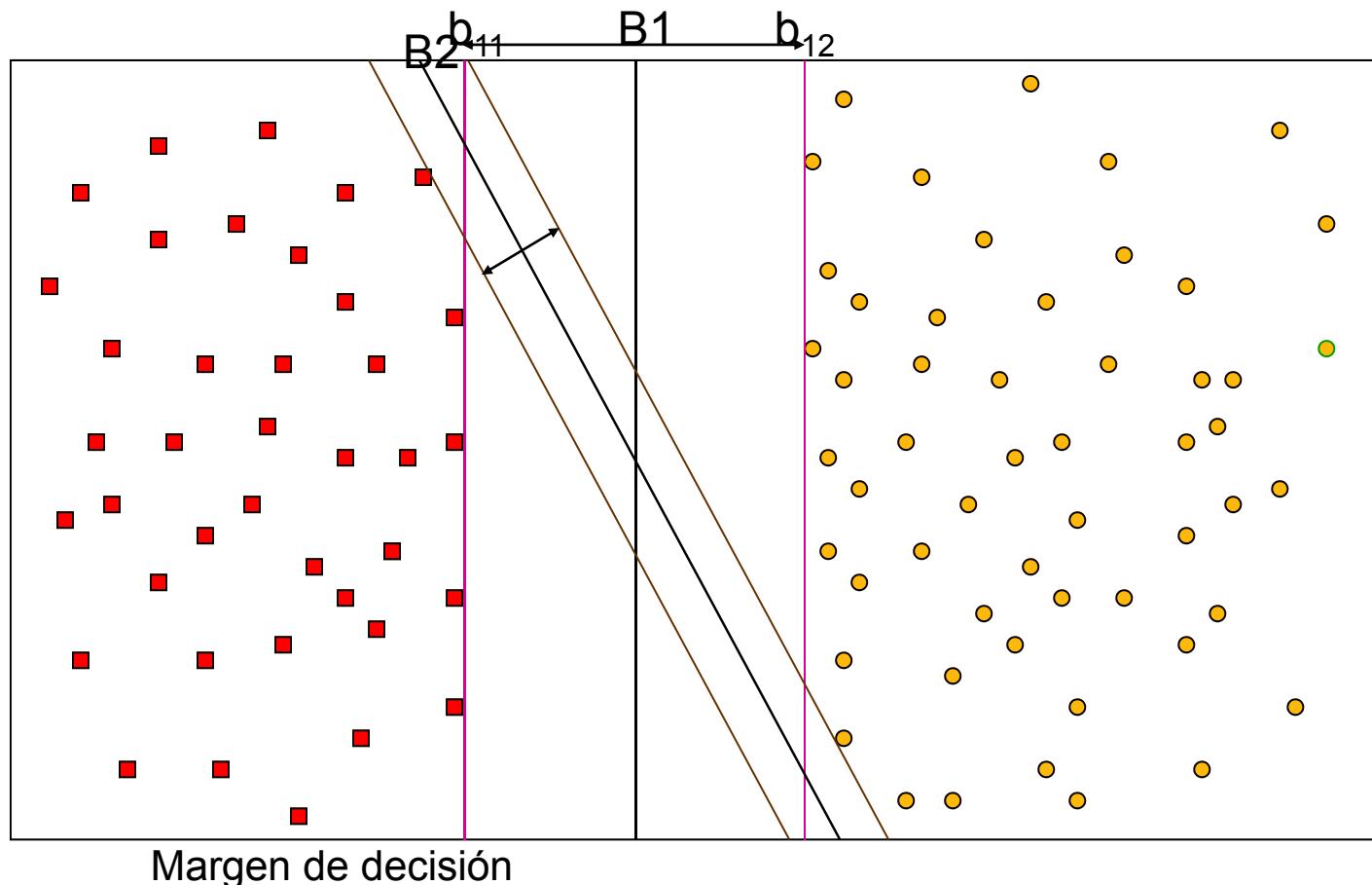
- Problemas linealmente separables



Posibles fronteras de decisión para datos linealmente separables

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



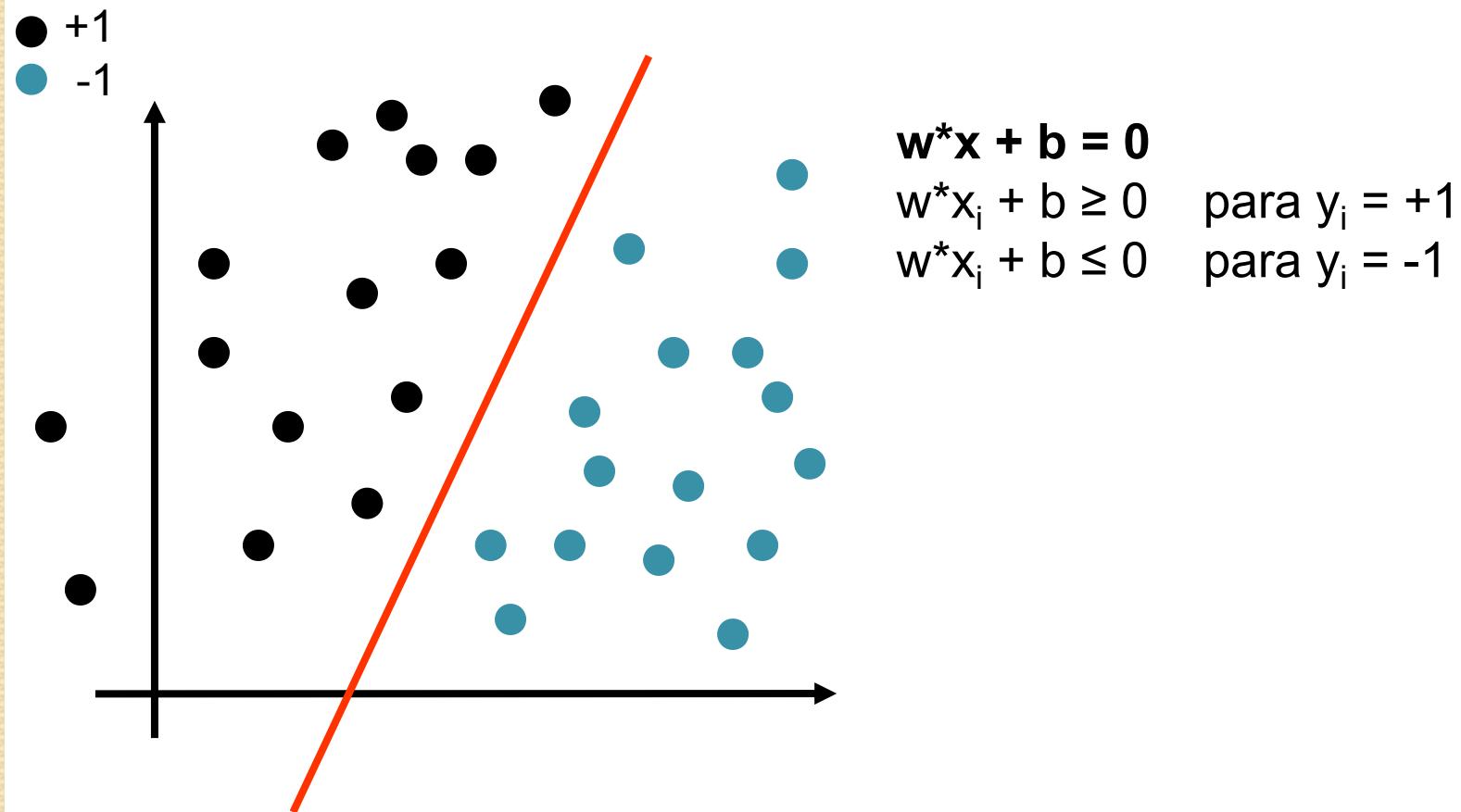


# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Dado un conjunto de ejemplos de entrenamiento,
  - Construir un hiperplano “ $w^*x + b$ ” como superficie de decisión de tal forma que la **separación de las dos clases sea máxima** (principio de generalización)
    - Margen de decisión lo más amplio posible
    - El entrenamiento busca encontrar  $w$  y  $b$

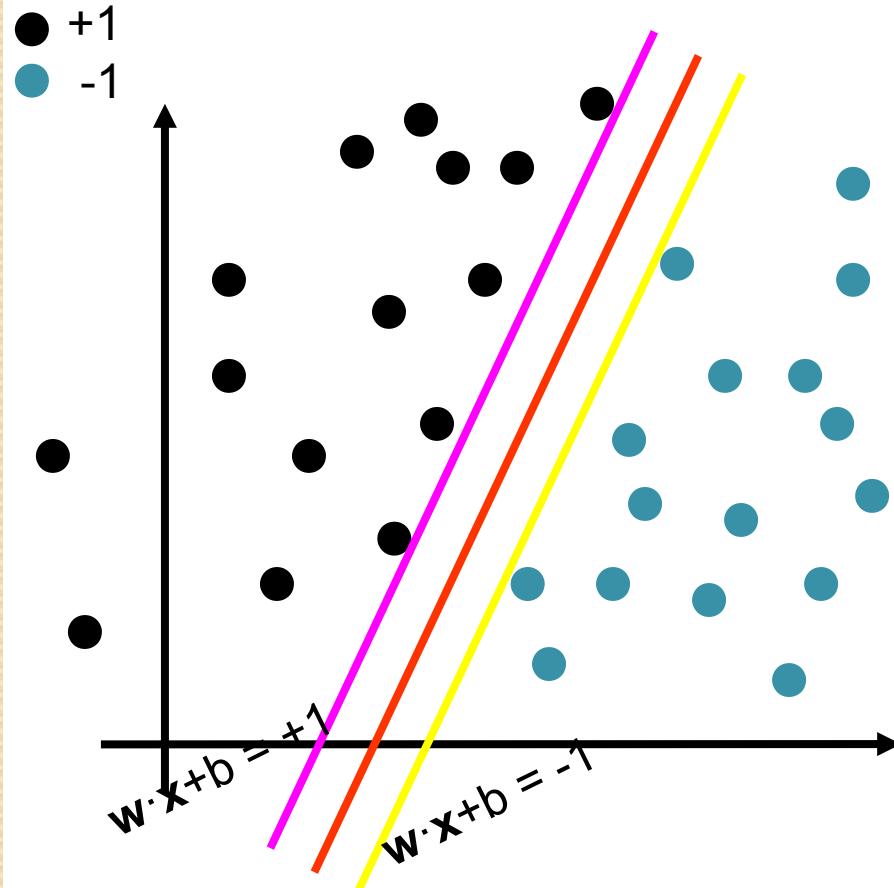
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Se definen los hiperplanos de los márgenes:



hiperplano “positivo”:  $w \cdot x + b = +1$

hiperplano “negativo”:  $w \cdot x + b = -1$

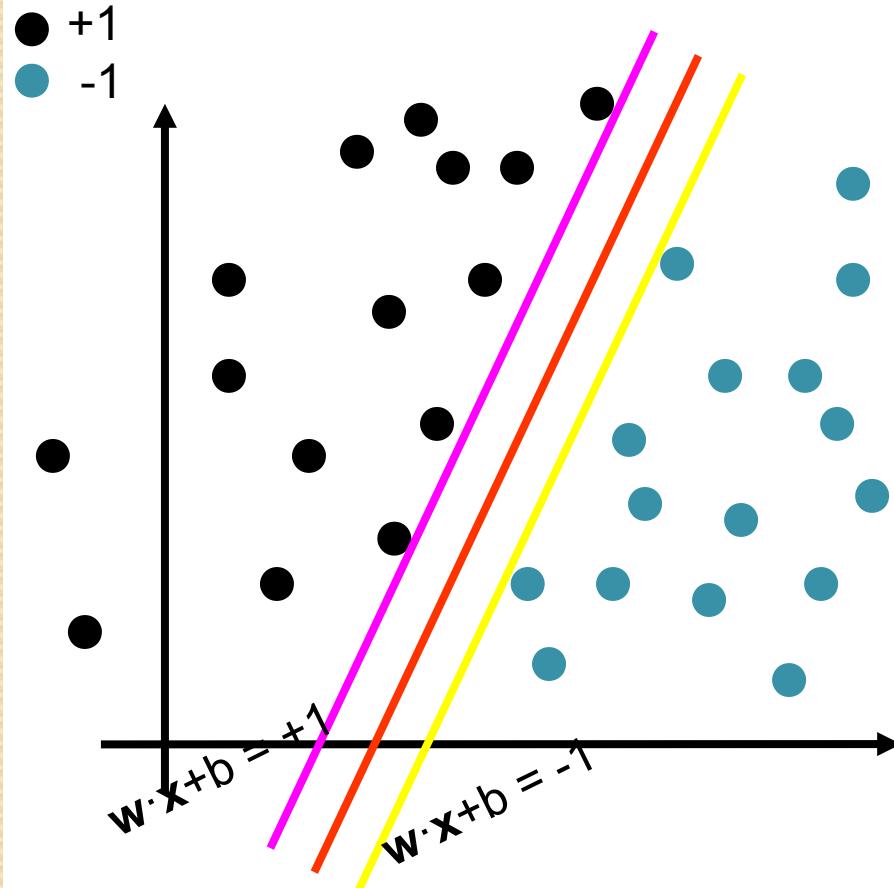
$$w^*x + b = 0$$

$w^*x_i + b \geq 0$  para  $y_i = +1$

$w^*x_i + b \leq 0$  para  $y_i = -1$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Se definen los hiperplanos de los márgenes:



hiperplano “positivo”:  $w \cdot x + b = +1$

hiperplano “negativo”:  $w \cdot x + b = -1$

$$w^*x + b = 0$$

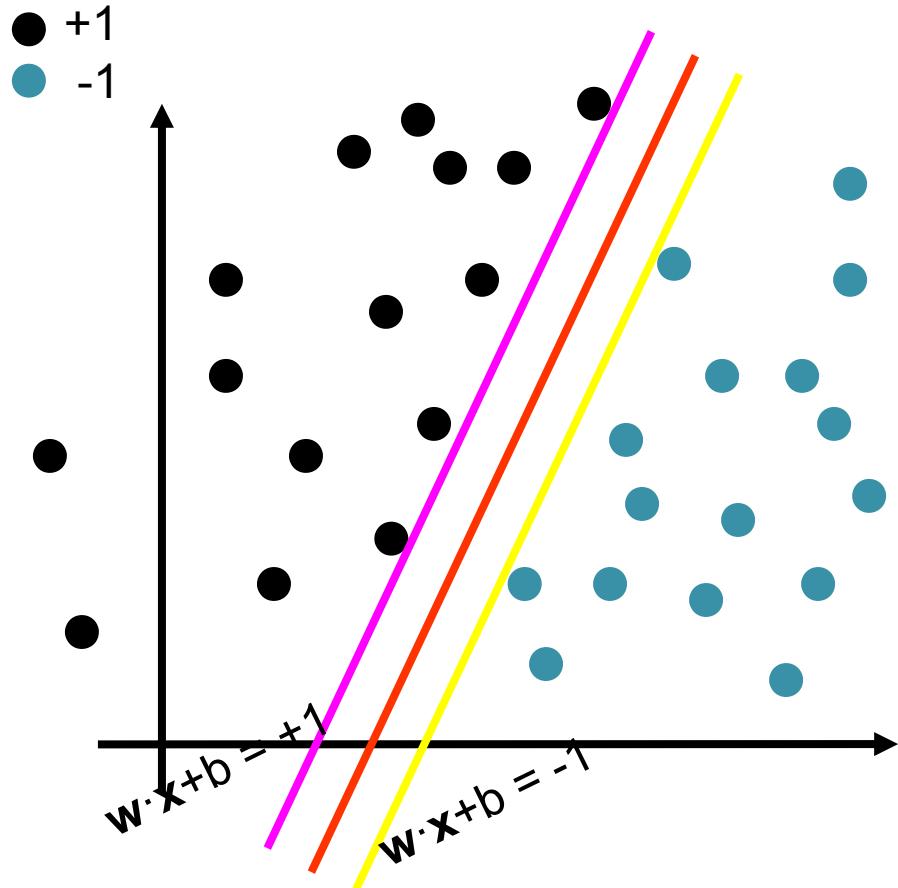
$w^*x_i + b \geq 0$  para  $y_i = +1$

$w^*x_i + b \leq 0$  para  $y_i = -1$

Se desea que dentro del margen no haya patrones:

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Se definen los hiperplanos de los márgenes:



hiperplano “positivo”:  $w \cdot x + b = +1$

hiperplano “negativo”:  $w \cdot x + b = -1$

$$w^*x + b = 0$$

$w^*x_i + b \geq 0$  para  $y_i = +1$

$w^*x_i + b \leq 0$  para  $y_i = -1$

Se desea que dentro del margen no haya patrones:

Todos los patrones ● cumplen:  
 $w \cdot x + b \geq +1$

Todos los patrones ● cumplen:  
 $w \cdot x + b \leq -1$



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Si se trabaja con un conjunto de vectores (patrones)  $S$ , se dice que es linealmente separable si existe  $(w,b)$  tal que las inecuaciones
$$\begin{cases} (w \cdot x_i + b) \geq 1 & y_i = 1 \text{ (casos positivos)} \\ (w \cdot x_i + b) \leq -1 & y_i = -1 \text{ (casos negativos)} \end{cases} \quad i=1, \dots, L$$
  - sean válidas para todos los elementos del conjunto  $S$ 
    - Para el caso linealmente separable de  $S$ , se puede encontrar un **único hiperplano óptimo**, para el cual el margen entre los puntos de entrenamiento de dos diferentes clases es maximizado



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Estas ecuaciones

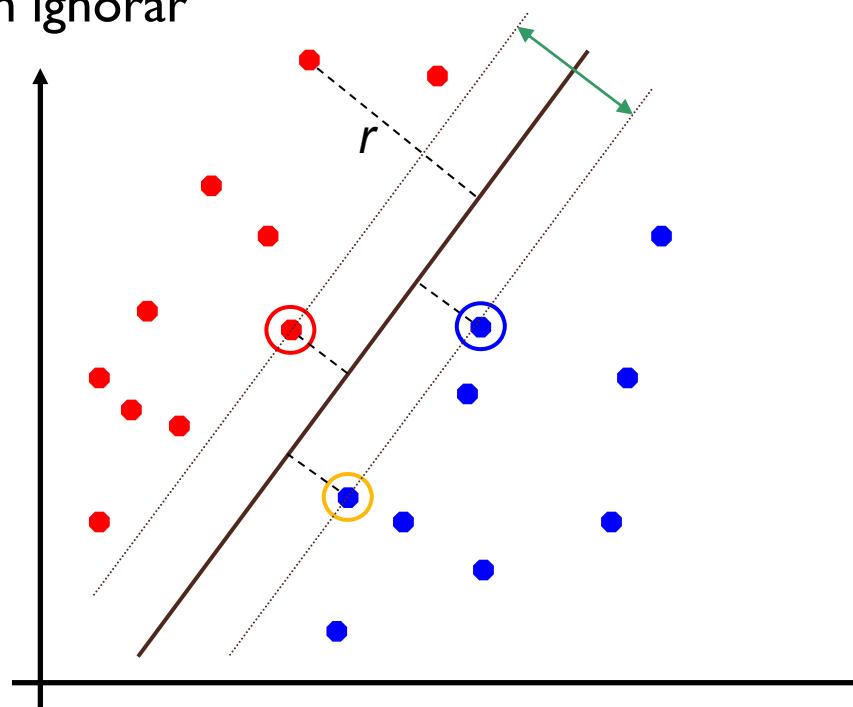
$$\begin{cases} (w \cdot x_i + b) \geq 1 & y_i = 1 \\ (w \cdot x_i + b) \leq -1 & y_i = -1 \end{cases} \quad i=1, \dots, L$$

- se pueden reformular a

$$y_i(w \cdot x_i + b) \geq 1 \quad i=1, \dots, L$$

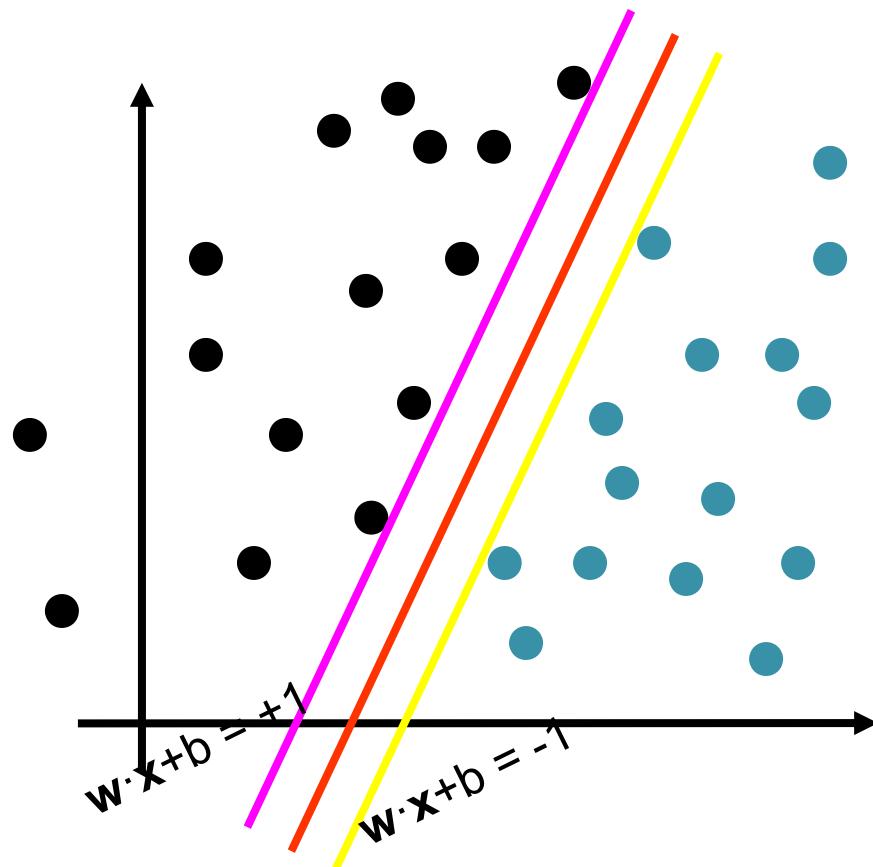
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Los ejemplos más cercanos al hiperplano son los llamados vectores de soporte
    - Solo estos importan, el resto de ejemplos de entrenamiento se pueden ignorar



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Función de decisión



$$w^*x + b = 0$$

$$w^*x_i + b \geq 0 \quad \text{para } y_i = +1$$
$$w^*x_i + b \leq 0 \quad \text{para } y_i = -1$$

hiperplano “positivo”:  $w \cdot x + b = +1$   
hiperplano “negativo”:  $w \cdot x + b = -1$

Todos los patrones ● cumplen:  
 $w \cdot x + b \geq +1$

Todos los patrones ● cumplen:  
 $w \cdot x + b \leq -1$

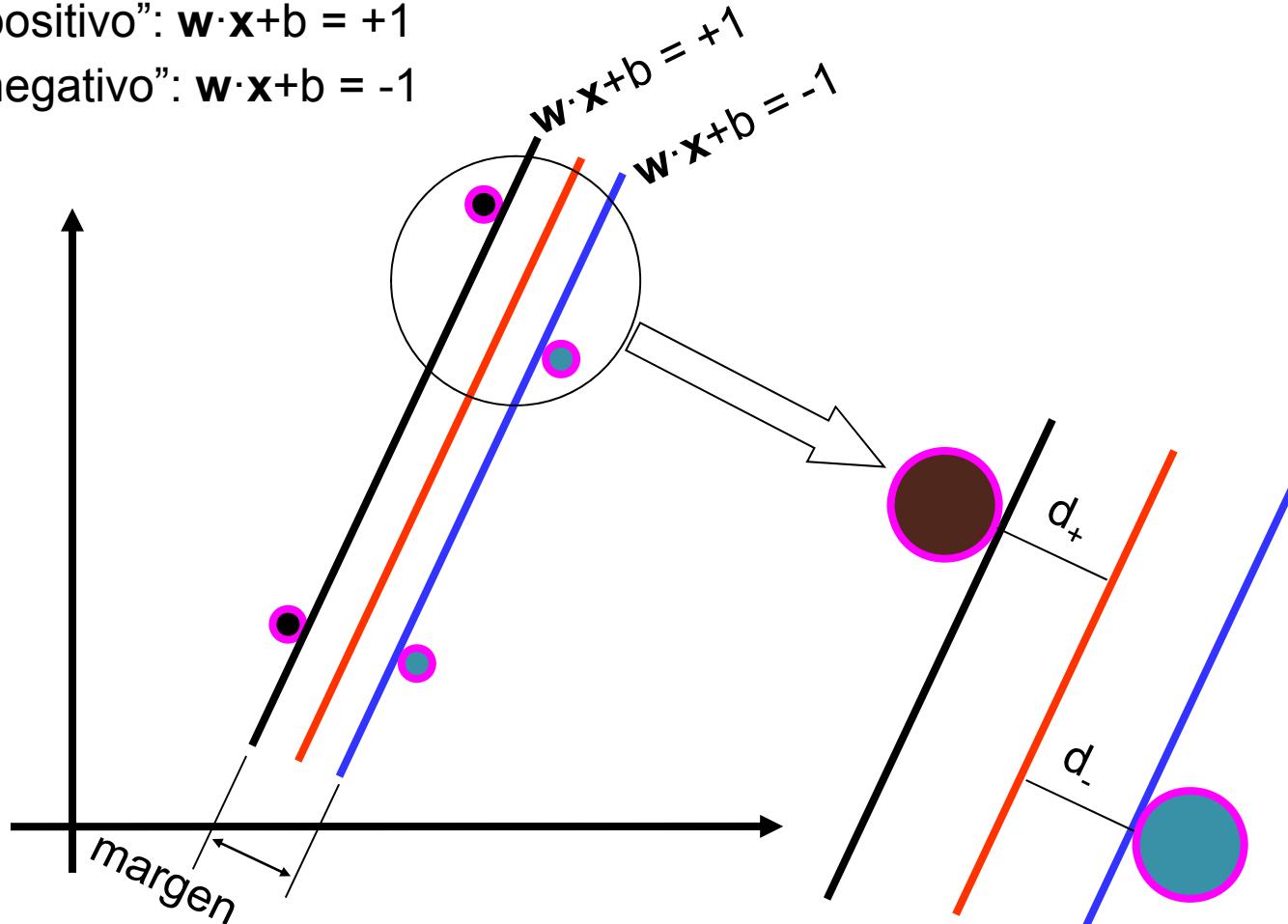
Los vectores de soporte cumplen:  
 $w \cdot x + b = +1$   
 $w \cdot x + b = -1$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

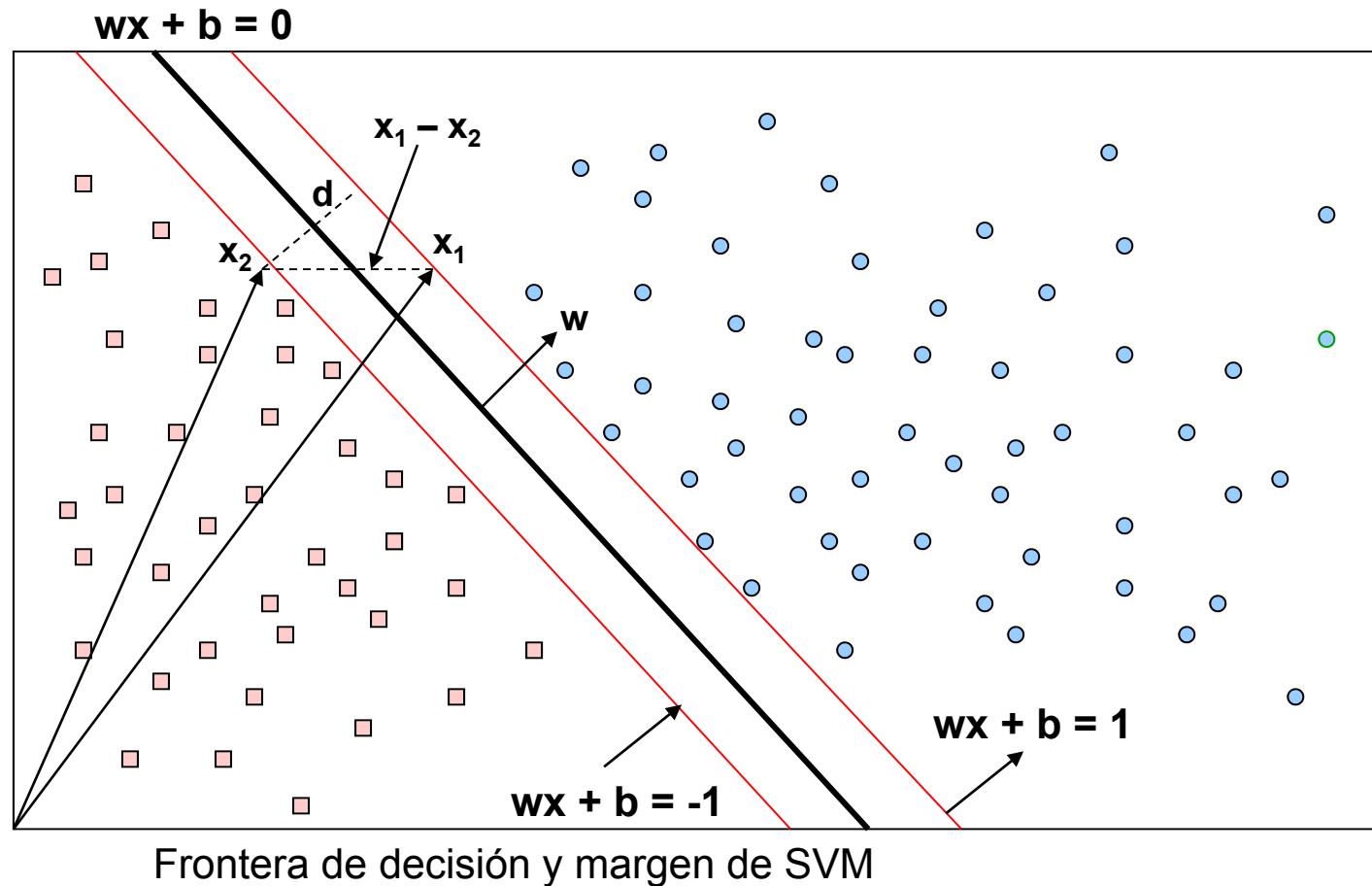
hiperplano “positivo”:  $w \cdot x + b = +1$

hiperplano “negativo”:  $w \cdot x + b = -1$



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Maximizar el ancho del margen,  $d$



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Se desea hallar  $w$  y  $b$  tales que se maximice el margen
  - Sea el “margen” d la distancia entre los hiperplanos “positivo” y “negativo”
    - Se desea maximizar ese valor d
    - Para cada ejemplo de entrenamiento:

$$\begin{aligned} w^T x_i + b &\leq -d/2 && \text{si } y_i = -1 \\ w^T x_i + b &\geq d/2 && \text{si } y_i = 1 \end{aligned}$$

- Es decir:

$$y_i(w^T x_i + b) \geq d/2$$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Para cada vector de soporte, la inecuación anterior es una igualdad

$y_i(w^T x_i + b) = d/2$  se puede transformar a la formulación anterior  $y_i(w^T x_i + b) = 1$   
reescalando w y b por d/2

- La distancia de un ejemplo  $x_i$  al plano separador es

$$r = \frac{w^T x_i + b}{\|w\|}$$

- Un vector de soporte está en el límite del margen
    - Uniendo ambas ecuaciones:  $r = \frac{y_i(w^T x_i + b)}{\|w\|} = \frac{1}{\|w\|}$

- Por lo tanto, el margen de separación a maximizar es:

$$d = 2r = \frac{2}{\|w\|}$$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- El margen es igual a:  $\frac{2}{\|\mathbf{w}\|}$
- La idea es encontrar un hiperplano con el máximo “margen”

- Esto es un problema de optimización:

maximizar:  $\frac{2}{\|\mathbf{w}\|}$       sujeto a:  $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$   
que también se puede expresar como

minimizar:  $\frac{1}{2} \|\mathbf{w}\|^2$       sujeto a:  $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

- donde  $\|\mathbf{w}\|^2$  es la norma euclídea de  $\mathbf{w}$
- Es decir, se necesita optimizar una función cuadrática sujeta a restricciones lineales
- Una restricción por cada patrón



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Los problemas de optimización cuadrática son una clase muy conocida de problemas matemáticos para los cuales existen varios algoritmos
- Pero el problema se puede transformar para que sea más fácil de manejar:
  - Se asocian multiplicadores de Lagrange ( $\alpha_i$ ) a cada una de las inecuaciones del problema original
  - Se construye el problema dual:

$$L_P \equiv \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1]$$

$$\alpha_i \geq 0 \quad \text{para todo } \alpha_i$$

- Se asocia un valor  $\alpha_i$  a cada patrón  $x_i$  y a cada inecuación

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Para construir el problema dual, se asocia un valor  $\alpha_i$  a cada inecuación (y, por tanto, a cada patrón), para incluir las restricciones dentro de la expresión

- Problema original:

$$\begin{array}{ll} \text{minimizar: } & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{sujeto a: } & y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \end{array}$$

$L_P = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1]$

$$\alpha_i \geq 0 \quad \text{para todo } \alpha_i$$

- Este problema debe de ser minimizado con respecto a  $w$  y  $b$ , y maximizado con respecto a  $\alpha_i \geq 0$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Problema dual minimizado con respecto a  $w$  y  $b$ :
    - Se halla el gradiente de  $L_p$  con respecto a  $w$  y  $b$  y se iguala a 0

$$\frac{\partial L_p}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^l \alpha_i y_i = 0$$

$$\frac{\partial L_p}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{i=1}^l \alpha_i y_i x_i$$

- Se sustituye esta expresión en la ecuación anterior:

$$L_p \equiv \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i [y_i (w \cdot x_i + b) - 1] \quad \left. \begin{array}{l} \\ w = \sum_{i=1}^l \alpha_i y_i x_i \end{array} \right\} \quad L_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j x_i^T x_j$$

- que debe ser maximizada con respecto a  $\alpha_i$ , con la restricción  $\sum_{i=1}^l \alpha_i y_i = 0$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - La forma para optimizar es, por tanto,

maximizar:  $L_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$

sujeto a:  $\sum_{i=1}^l \alpha_i y_i = 0$

$\alpha_i \geq 0 \text{ para todo } \alpha_i$

- En esta expresión ya no aparecen  $w$  ni  $b$ , solamente los  $\alpha_i$ 
  - Uno para cada patrón

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - La forma para optimizar es, por tanto,

maximizar:  $L_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$

sujeto a:  $\sum_{i=1}^l \alpha_i y_i = 0$

$\alpha_i \geq 0 \text{ para todo } \alpha_i$

- Esta optimización se puede resolver mediante técnicas de programación cuadrática (PQ) (Bertsekas 1995)



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables
  - Dada una solución  $\alpha$  a ese problema, la solución al problema “primario” sería, para  $w$ :
  - $$w = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i$$
  - Expresión anterior
  - Suma para cada patrón
  - Cada  $\alpha_i$  distinto de 0 indica que el correspondiente  $x_i$  es un **vector de soporte**
  - Si  $\alpha_i$  es 0, ese vector  $x_i$  no influye en el cálculo de  $w$  (producto por 0)

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Dada una solución  $\alpha$  a ese problema, la solución al problema “primario” sería, para  $b$ :
    - Un vector de soporte cumple

$$y_k(w^T x_k + b) = 1$$

- Entonces el parámetro  $b$  se calcula:

$$y_k(w^T x_k + b) = 1 \quad \xrightarrow{w^T x_k + b = 1/y_k = y_k} \quad b = y_k - w^T x_k$$

para cualquier vector de soporte  $\alpha_k > 0$

- Para cualquiera va a dar el mismo valor
  - En la práctica, el valor de  $b$  se promedia para evitar problemas de redondeo:

$$b = \frac{1}{N_{VS}} \sum_{i=1}^{N_{VS}} (y_i - w^T x_i)$$

para cada vector de soporte  
 $N_{VS}$ : número de vectores de soporte

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Por tanto, la solución es:

$$w = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad b = y_k - w^T x_k$$

para cualquier  $\alpha_k > 0$

$$b = y_k - \sum_{i=1}^l \alpha_i y_i x_i^T x_k$$

- De esta manera, la función de clasificación es:

$$f(x) = \sum_{i=1}^l \alpha_i y_i \boxed{x_i^T x} + b$$

- No se necesita calcular los  $w$
    - En esta función se realiza el **producto escalar** entre el punto a clasificar y cada uno los vectores de soporte
    - Los patrones que no son vectores de soporte tienen  $\alpha_i = 0$



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables

- Además, si los  $\alpha_i \geq 0$ , se pueden aplicar las condiciones de Karush Kuhn Tucker para convertir las desigualdades:

$$y_i(w^T x_i + b) \geq 1$$

en varias ecuaciones de igualdad para cada ejemplo  $x_i$ :

$$\alpha_i(y_i(w^T x_i + b) - 1) = 0$$

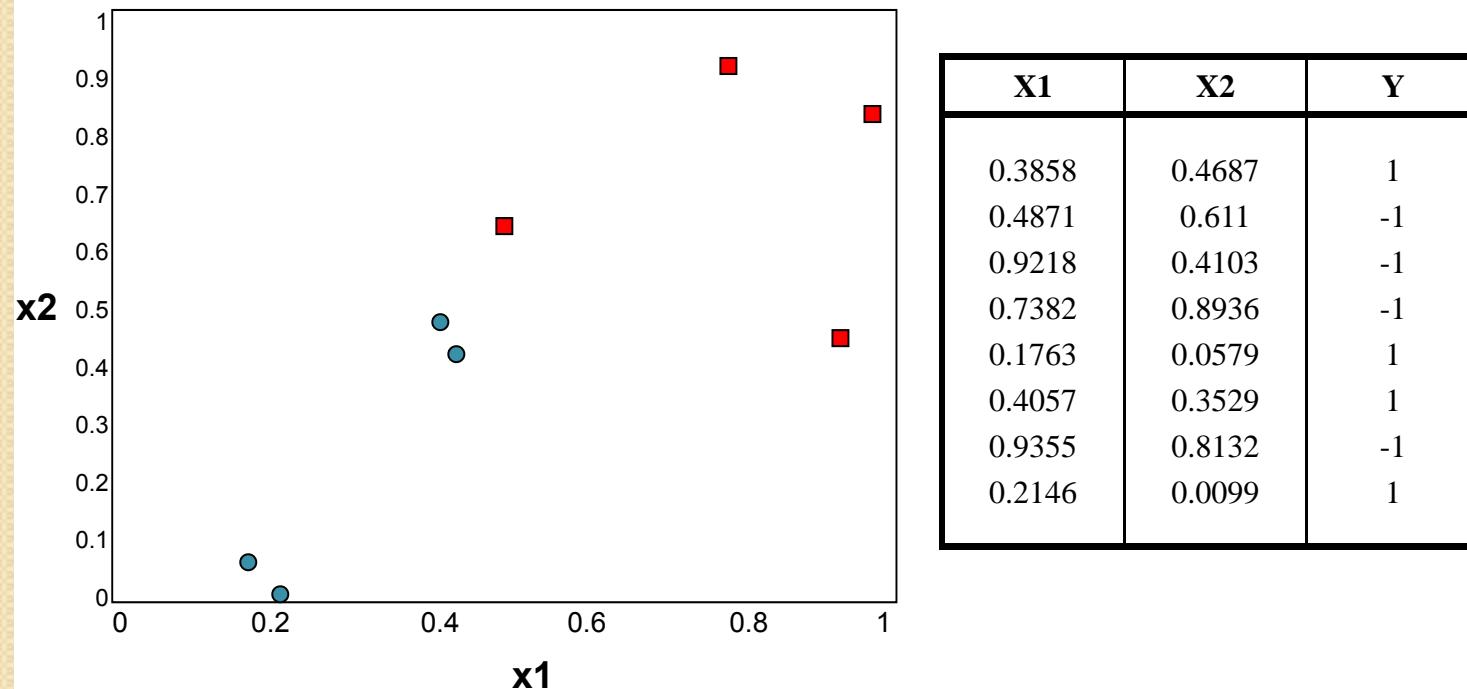
- Todos los ejemplos donde los  $\alpha_i > 0$  son los vectores de soporte
    - Si  $\alpha_i \neq 0$ , entonces

$$y_i(w^T x_i + b) - 1 = 0 \quad \Rightarrow \quad y_i(w^T x_i + b) = 1$$

y, por lo tanto,  $x_i$  está en el límite del margen de decisión

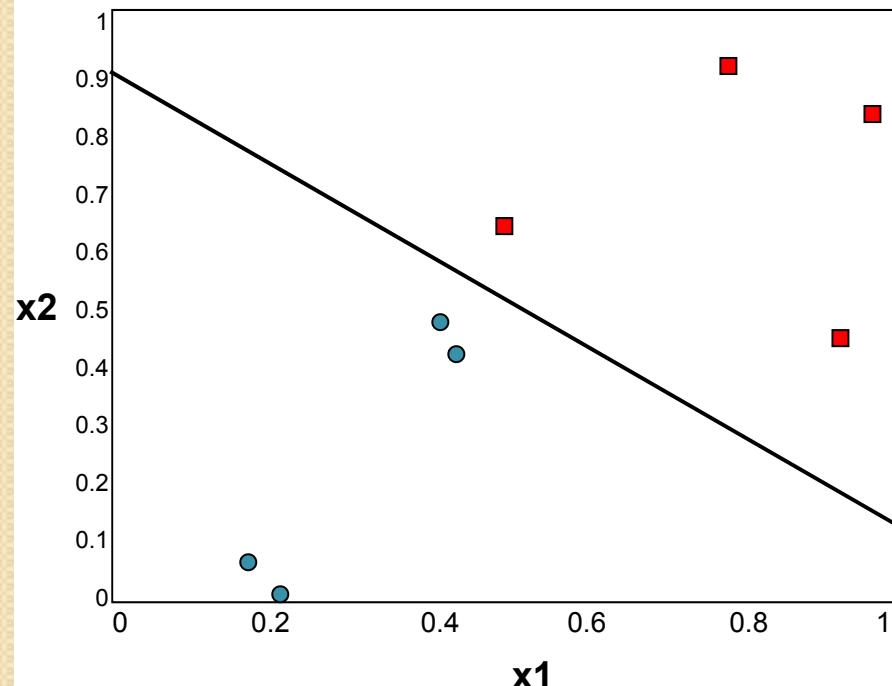
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables: Ejemplo:



# MÁQUINAS DE SOPORTE VECTORIAL

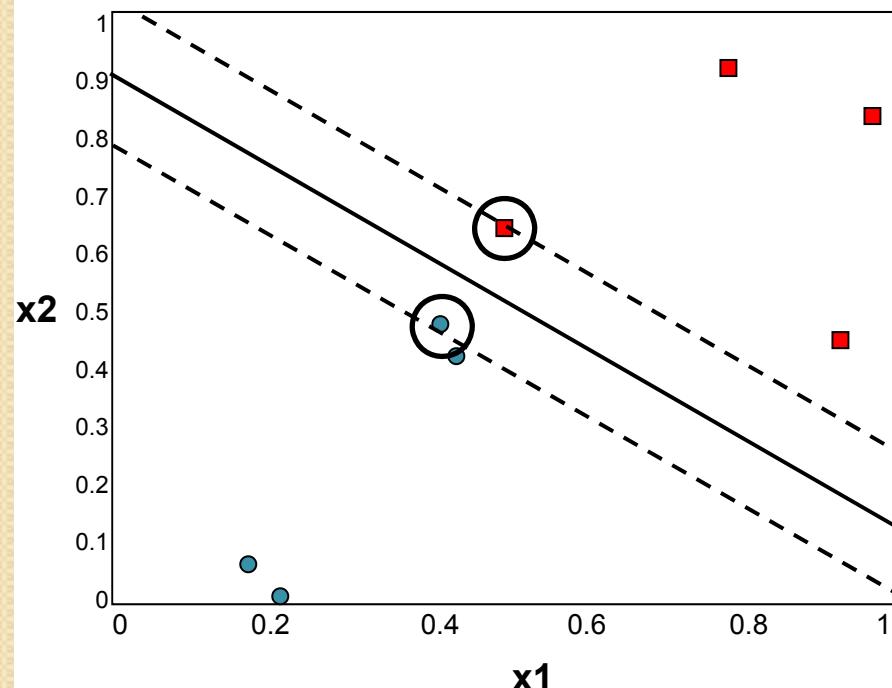
- Problemas linealmente separables: Ejemplo:



X1	X2	Y	$\alpha_i$
0.3858	0.4687	1	65.5261
0.4871	0.611	-1	65.5261
0.9218	0.4103	-1	0
0.7382	0.8936	-1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	-1	0
0.2146	0.0099	1	0

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables: Ejemplo:



X1	X2	Y	$\alpha_i$
0.3858	0.4687	1	65.5261
0.4871	0.611	-1	65.5261
0.9218	0.4103	-1	0
0.7382	0.8936	-1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	-1	0
0.2146	0.0099	1	0

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables: Ejemplo:
  - Solución:

- Cálculo de  $w$ :

$$w = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i = 65.5621 * 1 * \begin{bmatrix} 0.3858 \\ 0.4687 \end{bmatrix} + 65.5621 * (-1) * \begin{bmatrix} 0.4871 \\ 0.611 \end{bmatrix} = \begin{bmatrix} -6.64 \\ -9.32 \end{bmatrix}$$

- Cálculo de  $b$ :

- Con el primer patrón:

$$b = y_k - w^T x_k = 1 - [-6.64 \quad -9.32] * \begin{bmatrix} 0.3858 \\ 0.4687 \end{bmatrix} = 1 - (-6.929996) = 7.929996$$

- Con el segundo patrón:

$$b = y_k - w^T x_k = -1 - [-6.64 \quad -9.32] * \begin{bmatrix} 0.4871 \\ 0.611 \end{bmatrix} = -1 - (-8.928862) = 7.928864$$

- Se promedian ambos valores:  $b = 7.929$



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables: Ejemplo:
  - Para clasificar un nuevo patrón  $x$ :

$$f(x) = w \cdot x + b$$

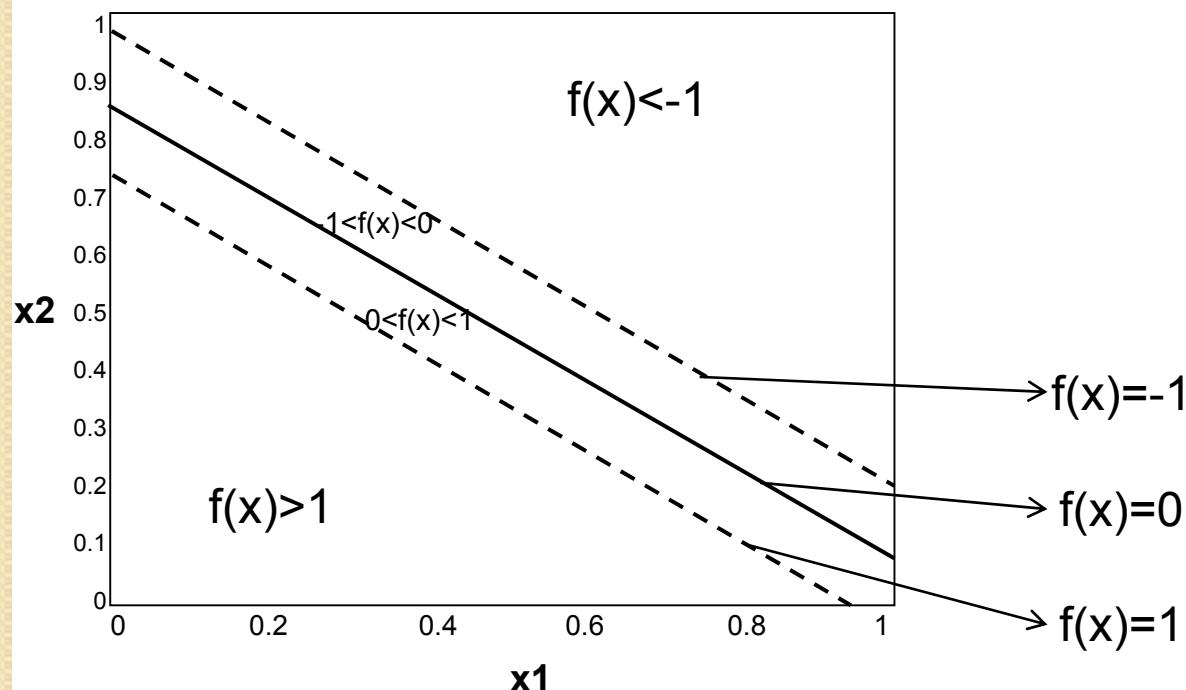
o bien

$$f(x) = \sum_{i=1}^l \alpha_i y_i x_i^T x + b$$

- Clasificación:
  - Si  $f(x) > 0 \rightarrow I$
  - Si  $f(x) < 0 \rightarrow -I$
  - $f(x)$  da un nivel de **confianza** sobre la pertenencia de ese patrón a una de las clases
  - Cuanto mayor es su valor absoluto, más confianza
    - Más alejado de la región de separación

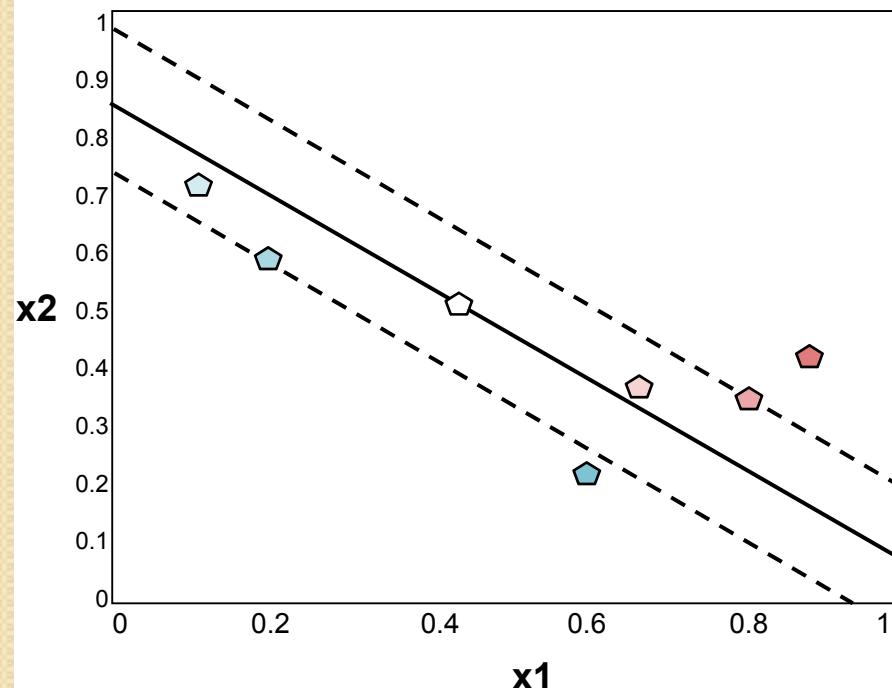
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente separables: Ejemplo:
  - Para clasificar un nuevo patrón  $x$ :



# MÁQUINAS DE SOPORTE VECTORIAL

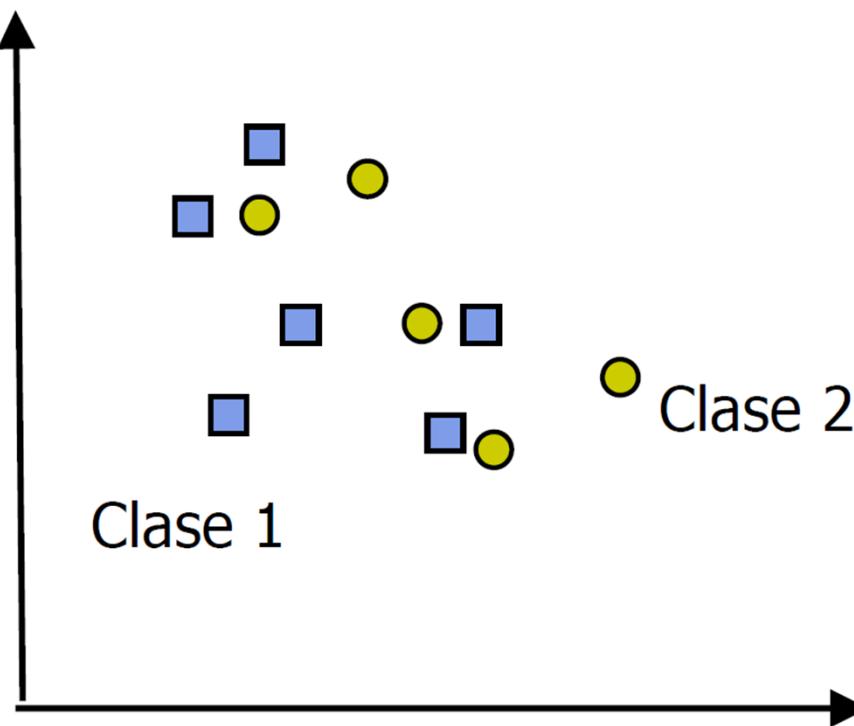
- Problemas linealmente separables: Ejemplo:
  - Para clasificar un nuevo patrón  $x$ :
    - Ejemplos:



	$x_1$	$x_2$	$f(x)$
Open Diamond	0.4	0.5658	0
Open Diamond	0.15	0.7042	0.37
Open Diamond	0.2	0.601	1
Solid Diamond	0.6	0.2623	1.5
Open Diamond	0.65	0.4091	-0.2
Solid Diamond	0.8	0.3881	-1
Solid Diamond	0.9	0.4563	-2.3

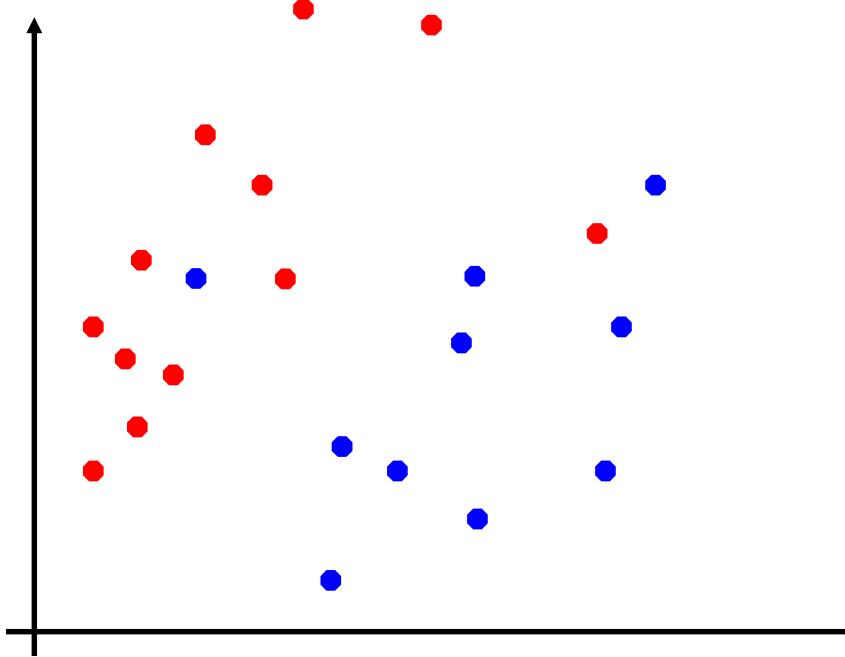
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables
  - Si el conjunto S no es linealmente separable, se debe permitir alguna violación a la clasificación en la formulación



# MÁQUINAS DE SOPORTE VECTORIAL

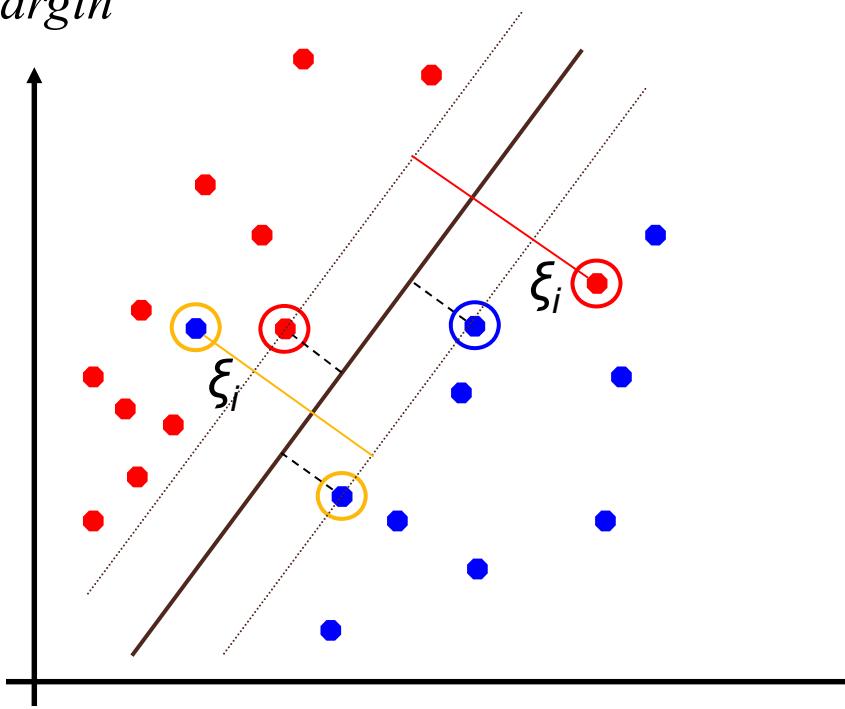
- Problemas no linealmente separables



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables

- En estos casos se añaden unas variables especiales  $\xi_i$  para permitir instancias incorrectamente clasificadas
  - Por ser difíciles de clasificar, o tener ruido
  - *Soft margin*



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables

- La formulación anterior:

Encontrar  $w$  y  $b$  tal que

$$\text{minimizar: } \frac{1}{2} \|w\|^2 \quad \text{sujeto a: } y_i(w^T x_i + b) \geq 1$$

cambiaría a:

Encontrar  $w$  y  $b$  tal que

$$\text{minimizar: } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad \text{sujeto a: } y_i(w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0$$

- El parámetro  $C$  puede verse como una forma de controlar el sobreajuste:
    - Es un compromiso entre la importancia relativa de maximizar el margen y ajustar los datos de entrenamiento

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables

$$\text{minimizar: } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i$$

- El parámetro C:

- Valores altos:

- Se da más importancia a minimizar los errores  $C \sum_{i=1}^l \xi_i$  y menos a maximizar el margen  $\frac{1}{2} \|\mathbf{w}\|^2$

- Es decir, que los errores en la clasificación sean los mínimos
      - Márgeles más estrechos

- Mayor precisión en la clasificación → Possible sobreentrenamiento

- Valores bajos:

- Se da menos importancia a minimizar los errores  $C \sum_{i=1}^l \xi_i$  y más a maximizar el margen  $\frac{1}{2} \|\mathbf{w}\|^2$

- Buscar márgenes más amplios, permitir algunos errores en la clasificación → menos sobreentrenamiento, más generalización

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables
  - Igual que antes, se construye el problema dual asociando multiplicadores de Lagrange  $\alpha_i, \beta_i$ :
    - Problema original:

$$\begin{aligned} & \text{minimizar: } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i \quad \text{sujeto a: } \\ & \qquad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \\ & \bullet \text{ Problema dual:} \\ & L_P = \frac{1}{2} \|\mathbf{w}\|^2 - C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^l \beta_i \xi_i \end{aligned}$$

- Este problema debe de ser minimizado con respecto a  $\mathbf{w}, b$  y  $\xi$ , y maximizado con respecto a  $\alpha_i$  y  $\beta_i$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables

- Problema dual minimizado con respecto a  $w, b$  y  $\xi$ :
    - Se halla el gradiente de  $L_p$  con respecto a  $w$  y  $b$  y se iguala a 0

$$\frac{\partial L_p}{\partial b} = 0 \rightarrow \sum_{i=1}^l \alpha_i y_i = 0 \quad \frac{\partial L_p}{\partial w} = 0 \rightarrow w = \sum_{i=1}^l \alpha_i y_i x_i$$

- Iguales al caso lineal

$$\frac{\partial L_p}{\partial \xi} = 0 \rightarrow \alpha_i + \beta_i = C$$

- Sustituyendo en la ecuación anterior:

$$L_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

- que debe ser maximizada con respecto a  $\alpha_i$ , con la restricción  $\sum_{i=1}^l \alpha_i y_i = 0$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables

- El problema dual se transformaría en

Encontrar  $\alpha_1 \dots \alpha_N$  tal que

$$\text{maximicen: } \sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j y_i y_j x_i^T x_j \quad \text{idéntica a la anterior}$$

$$\text{sujeto a : } \sum_{i=1}^L y_i \alpha_i = 0$$

$$0 < \alpha_i < C \quad i = 1, \dots, L$$

- Mismas técnicas para hallar los  $\alpha_i$
- Nuevamente, los patrones  $x_i$  con  $\alpha_i$  distintos de cero serán vectores de soporte

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables
  - Una vez hallados los  $\alpha$ , la solución sería similar:

- w:

$$w = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i$$

- b: para cualquier vector de soporte ( $\alpha_k > 0$ ):

$$y_k(w^T x_k + b) = 1 - \xi_k \quad \Leftrightarrow \quad w^T x_k + b = \frac{1 - \xi_k}{y_k} = y_k(1 - \xi_k) \quad \Leftrightarrow$$

$$\Leftrightarrow b = y_k(1 - \xi_k) - w^T x_k \quad b = y_k(1 - \xi_k) - \sum_{i=1}^l \alpha_i y_i x_i^T x_k$$

- Igual que antes, no es necesario calcular w para realizar la clasificación:

$$f(x) = \sum_{i=1}^l \alpha_i y_i x_i^T x + b$$

# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables

- Nuevamente, si los  $\alpha_i \geq 0$ , se puede aplicar el teorema de Kuhn-Tucker para convertir las desigualdades:

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

en varias ecuaciones de igualdad para cada ejemplo i:

$$\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) = 0 \quad (C - \alpha_i)\xi_i = 0$$

- Todos los  $x_i$  donde los  $\alpha_i > 0$  son los vectores de soporte
    - Si  $\alpha_i \neq 0$ , entonces

$$y_i(w^T x_i + b) - 1 + \xi_i = 0 \quad \rightarrow \quad y_i(w^T x_i + b) = 1 - \xi_i$$

y, por lo tanto,  $x_i$  podría estar en el límite del margen de decisión

- Si  $\xi_i > 0$ ,  $x_i$  no está en el límite del margen, pero es vector de soporte



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables
  - Los puntos  $x_i$  con  $\alpha_i > 0$  son los vectores de soporte.
    - Los que tocan y definen los límites del margen
  - Los vectores soporte son los elementos críticos del conjunto de entrenamiento y estos son los más cercanos a la cota de decisión.
  - En un caso no separable hay dos tipos:
    - $0 < \alpha_i < C$ :
      - Dado que  $(C - \alpha_i)\xi_i = 0$  y  $\alpha_i < C$ , se deduce que  $\xi_i = 0$
      - En este caso, el vector de soporte  $x_i$  satisface las igualdades
$$y_i(w \cdot x_i + b) = 1 \quad \xi_i = 0$$
      - El vector de soporte está en el límite del margen

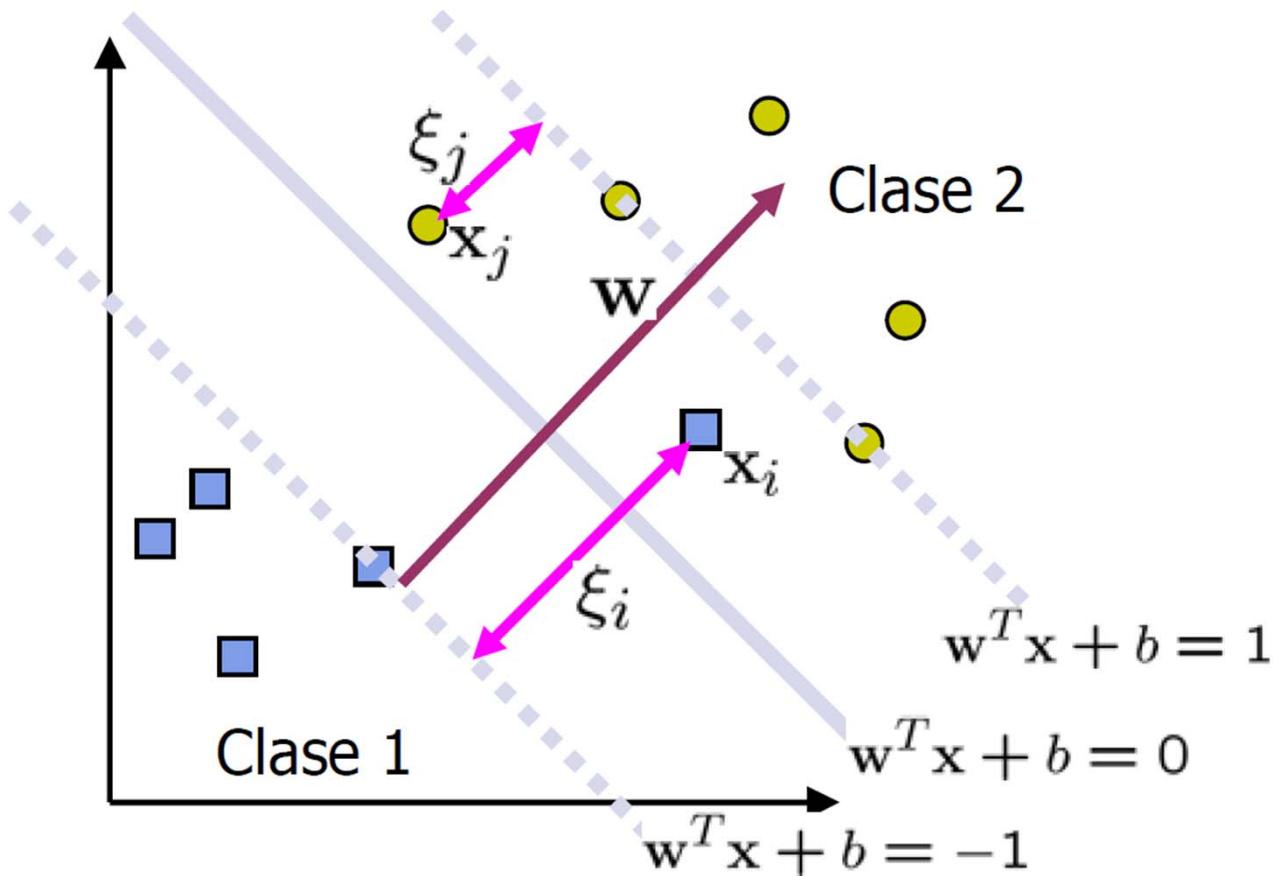


# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables
  - Los puntos  $x_i$  con  $\alpha_i > 0$  son los vectores de soporte.
    - Los que tocan y definen los límites del margen
  - Los vectores soporte son los elementos críticos del conjunto de entrenamiento y estos son los más cercanos a la cota de decisión.
  - En un caso no separable hay dos tipos:
    - $\alpha_i = C$ 
      - En este caso  $\xi_i \neq 0$
      - Por lo tanto, el correspondiente  $x_i$  no satisface  $y_i(w^T x_i + b) = 1$
      - Pero sí satisface  $y_i(w^T x_i + b) = 1 - \xi_i$
      - Es decir, es un error de clasificación
      - Pero si se le suma el valor de error  $\xi_i$ , se “desplaza” este  $x_i$  al límite del margen

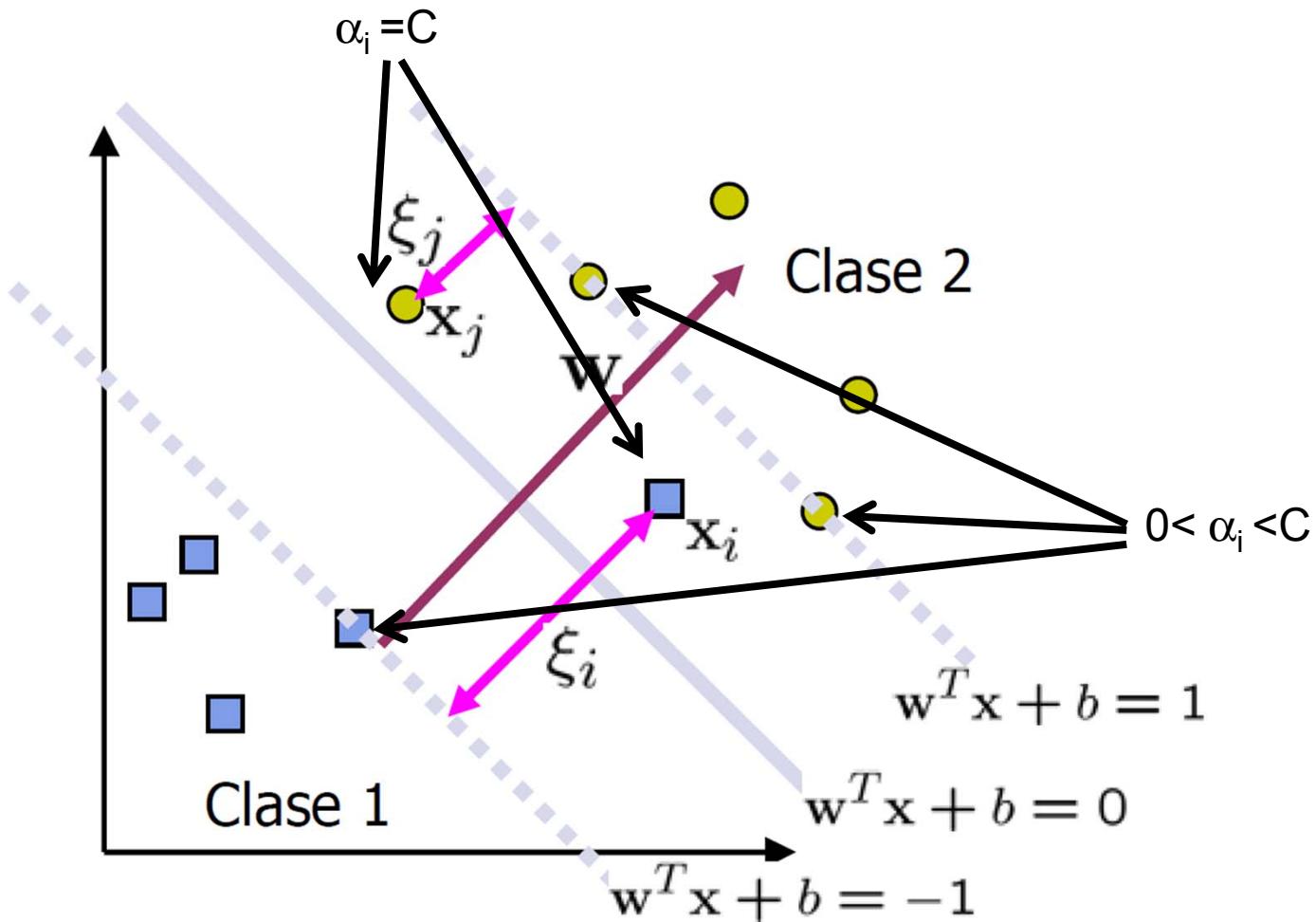
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables



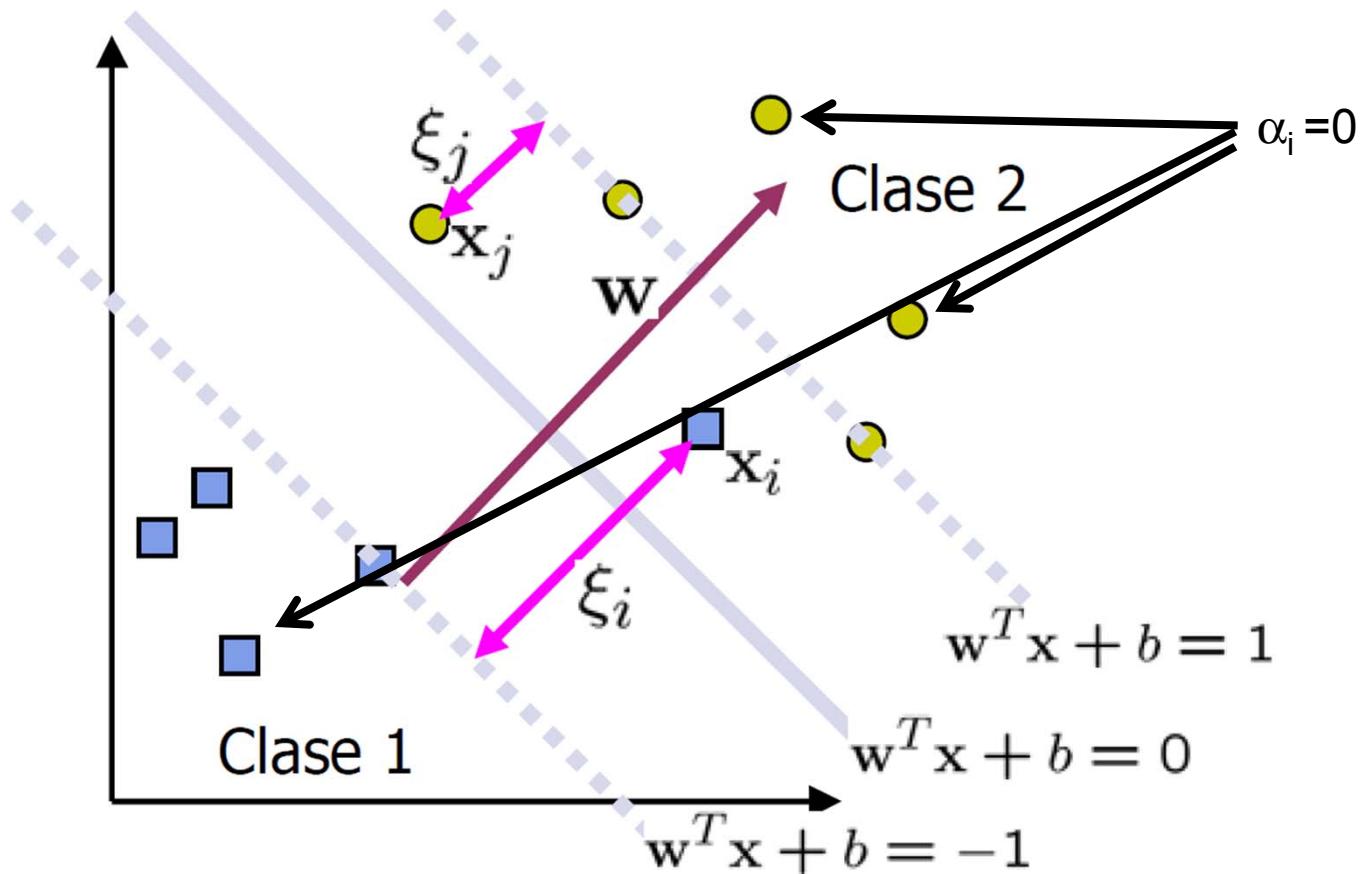
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables



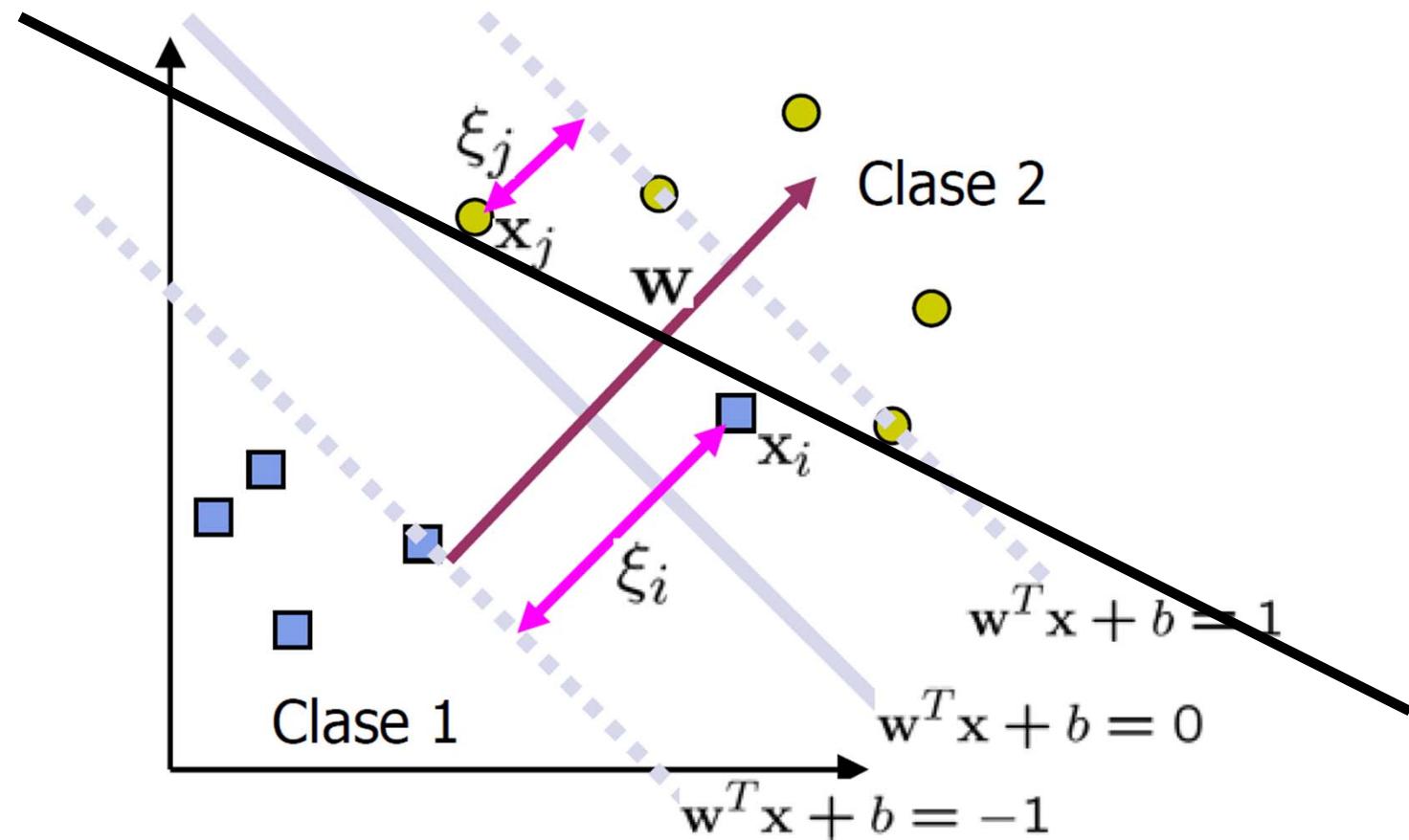
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas no linealmente separables
  - Además, el caso  $\alpha_i=0$  se corresponde con puntos que están claramente alejados del margen de decisión:



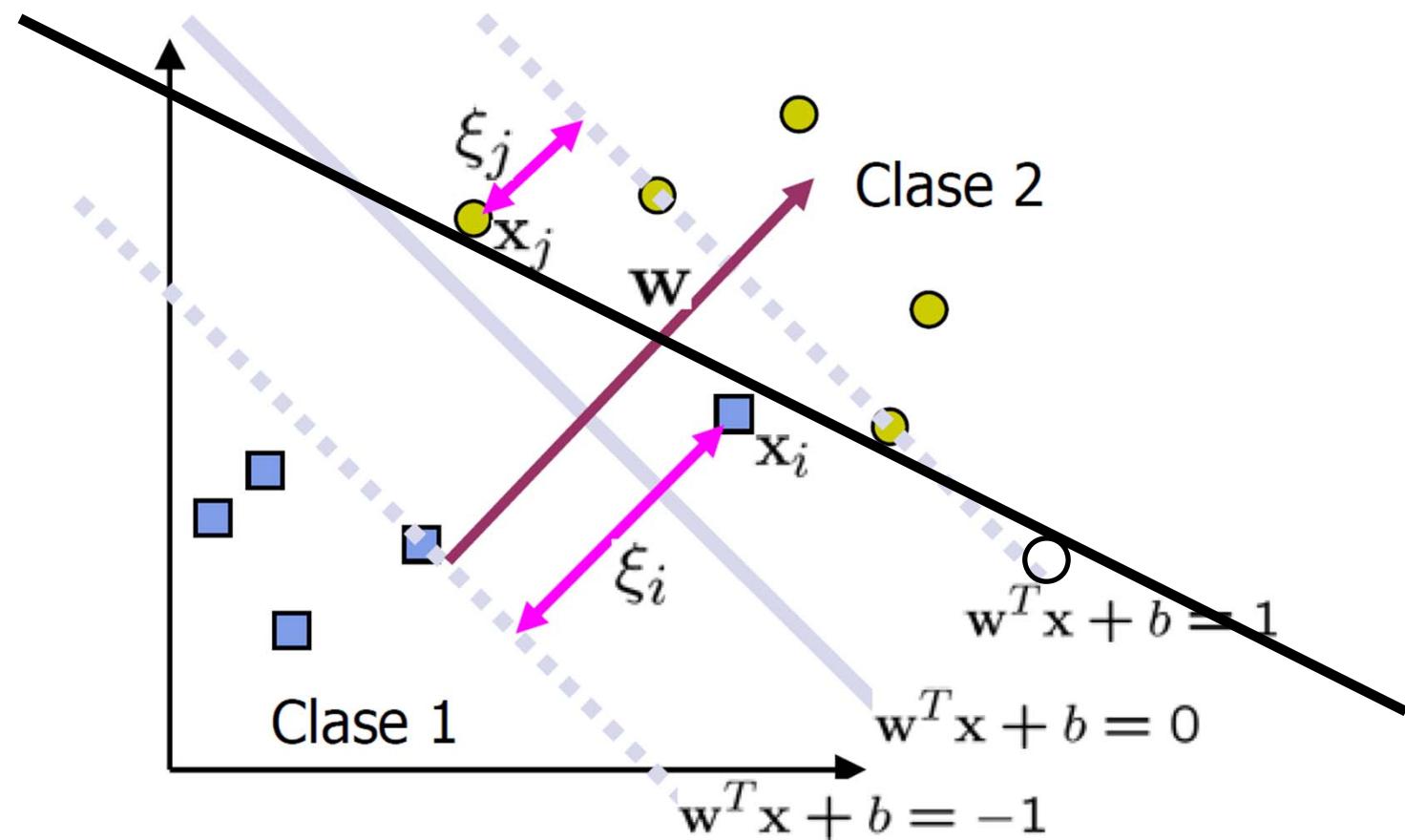
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente y no linealmente separables
  - Este problema es linealmente separable:



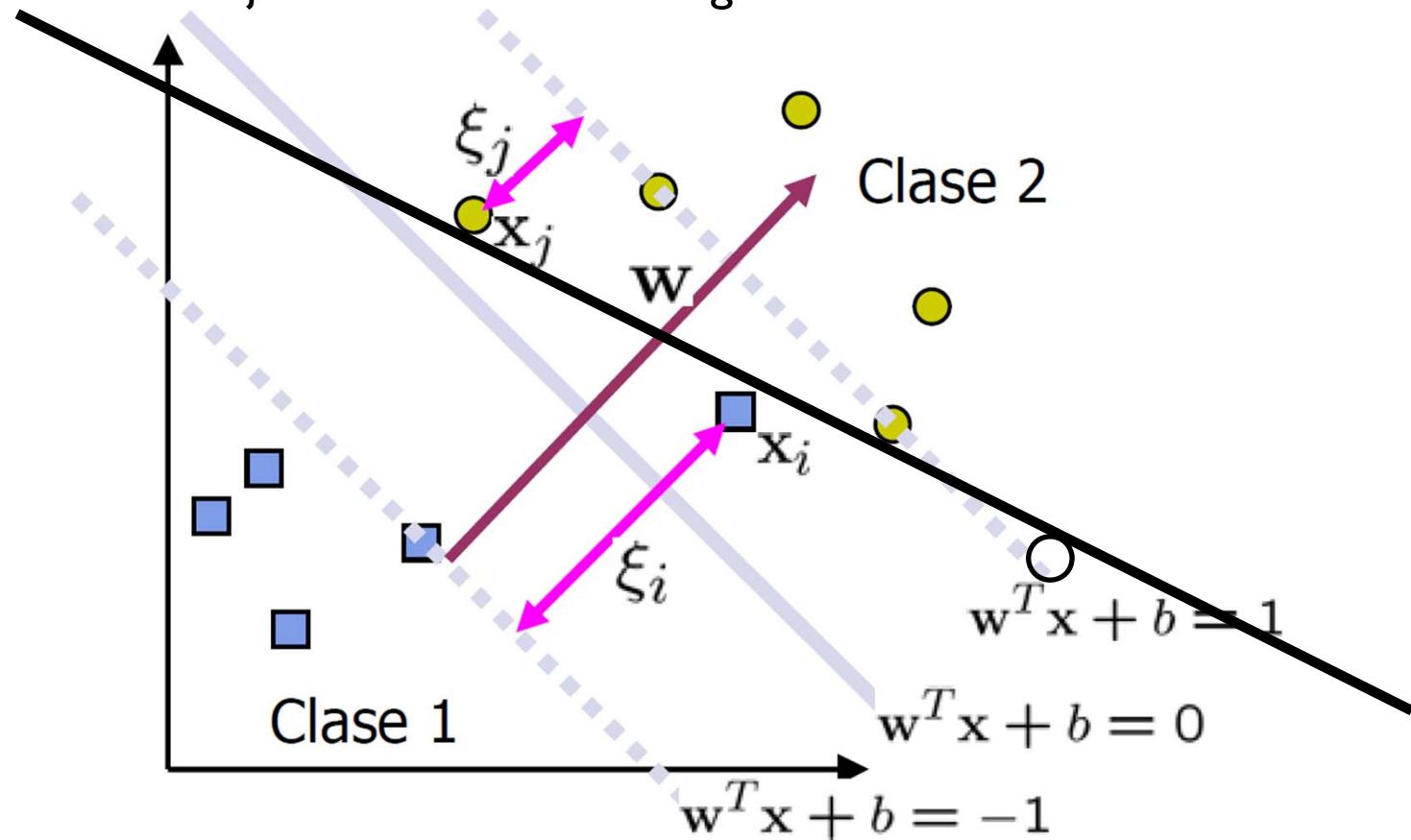
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente y no linealmente separables
  - ¿Pero qué ocurriría con este nuevo patrón?



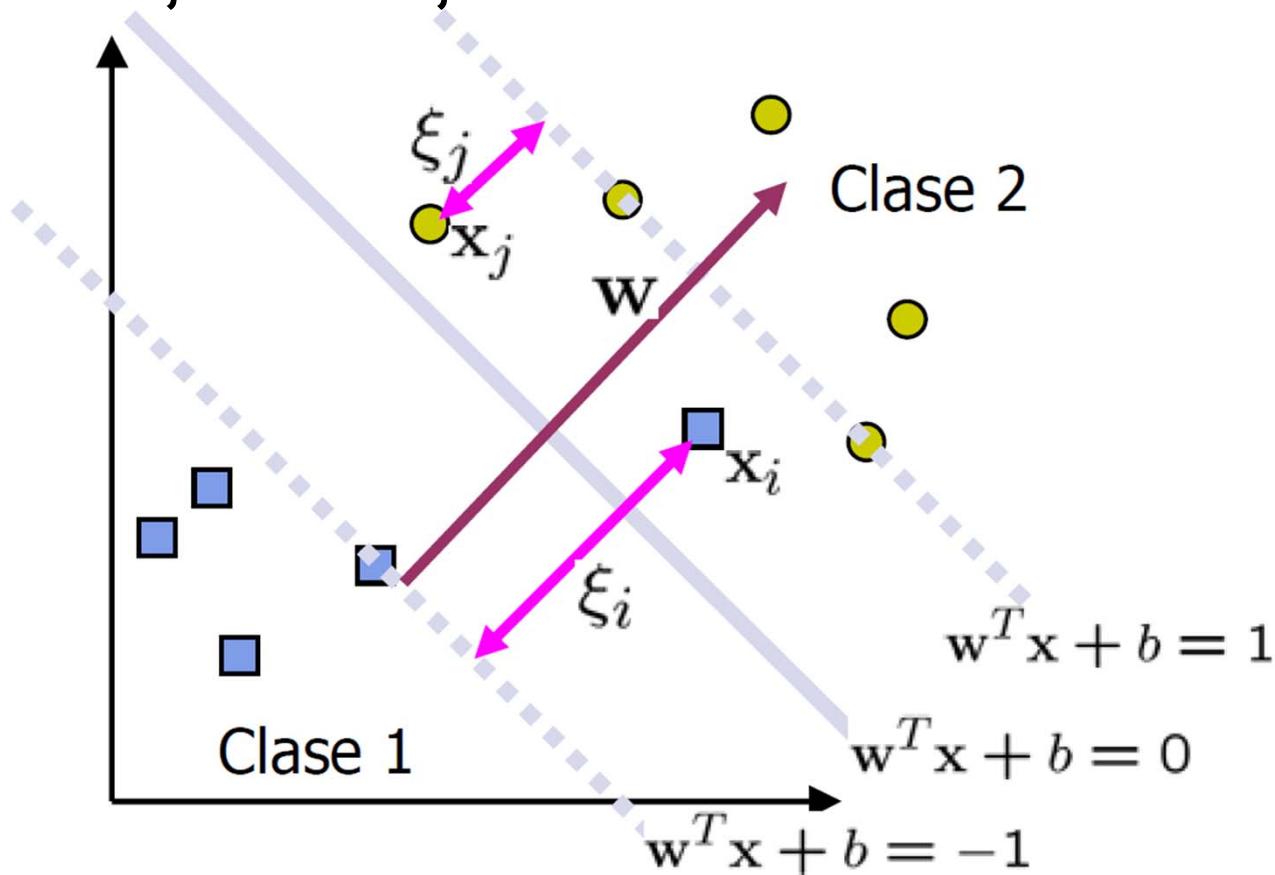
# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente y no linealmente separables
  - Margen demasiado estrecho
    - Sobreajustado: Problemas de generalización



# MÁQUINAS DE SOPORTE VECTORIAL

- Problemas linealmente y no linealmente separables
  - Permitir cierto nivel de error ( $\xi$ ) para evitar el sobreajuste: C bajo



# MÁQUINAS DE SOPORTE VECTORIAL

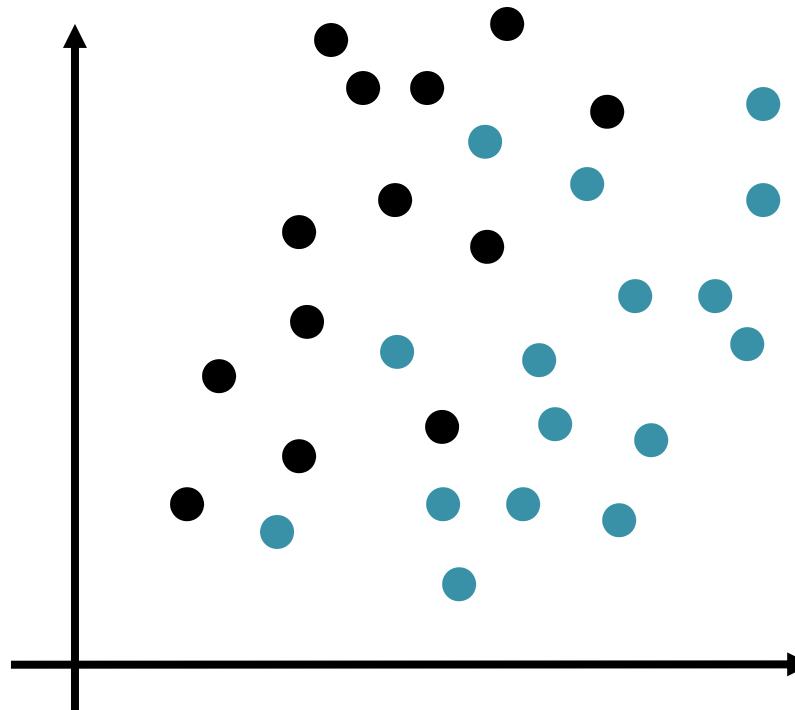
- SVM lineales:

- El clasificador es un hiperplano separador
- Los patrones de entrenamiento más importantes son vectores de soporte
  - Ellos definen el hiperplano
- Los algoritmos de optimización cuadrática permiten identificar qué patrones de entrenamiento  $x_i$  son vectores de soporte, con multiplicadores de Lagrange no nulos  $\alpha_i$
- Tanto en la formulación dual del problema como en la solución aparecen solamente **productos escalares**:

$$\sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j y_i y_j \boxed{x_i^T x_j} \quad f(x) = \sum_{i=1}^L \alpha_i y_i \boxed{x_i^T x} + b$$

# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales
  - El modelo anterior funciona bien con bases de datos con algo de ruido

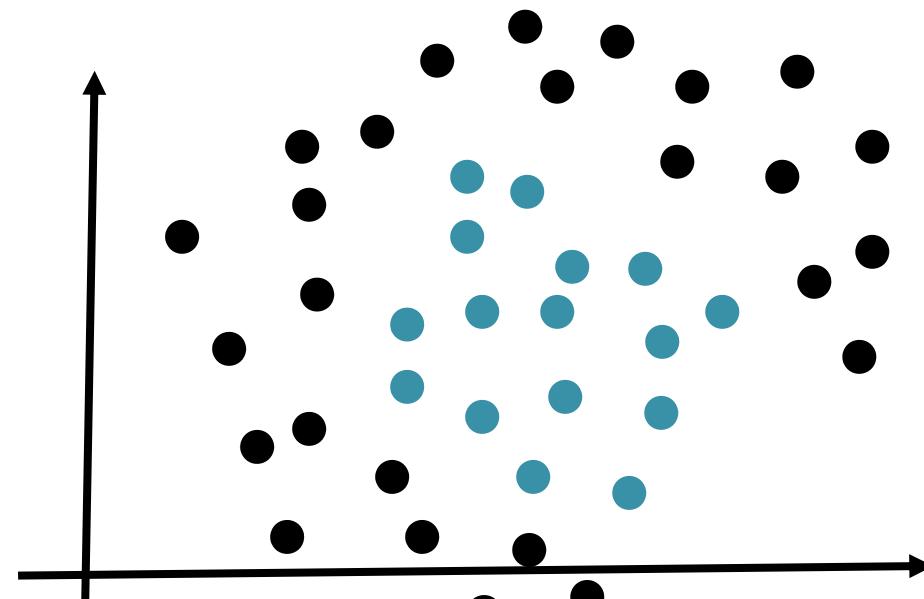


# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales

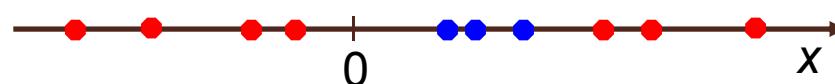
- Pero, ¿qué ocurre si los datos tienen una distribución muy difícilmente separable mediante un hiperplano?

- Por ejemplo:  
(2 dimensiones)



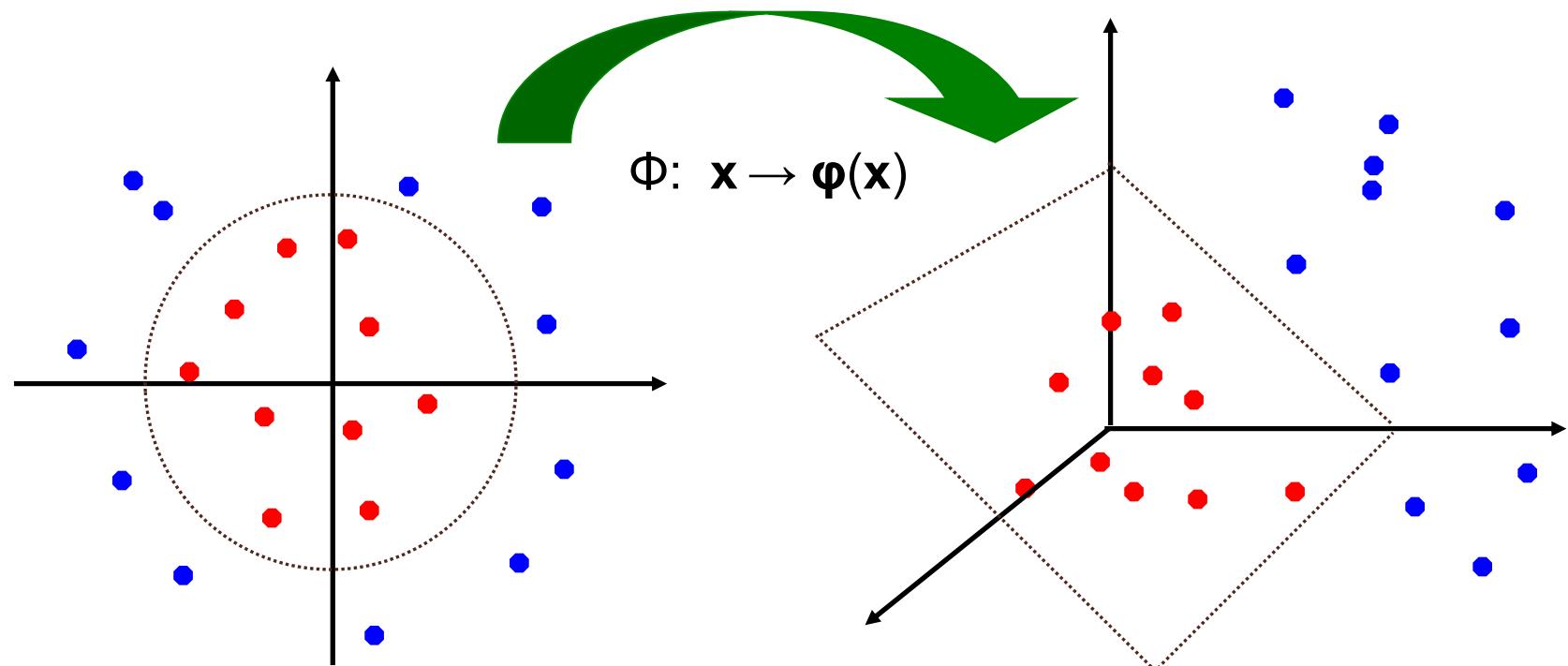
- Por ejemplo:

- (1 dimensión)



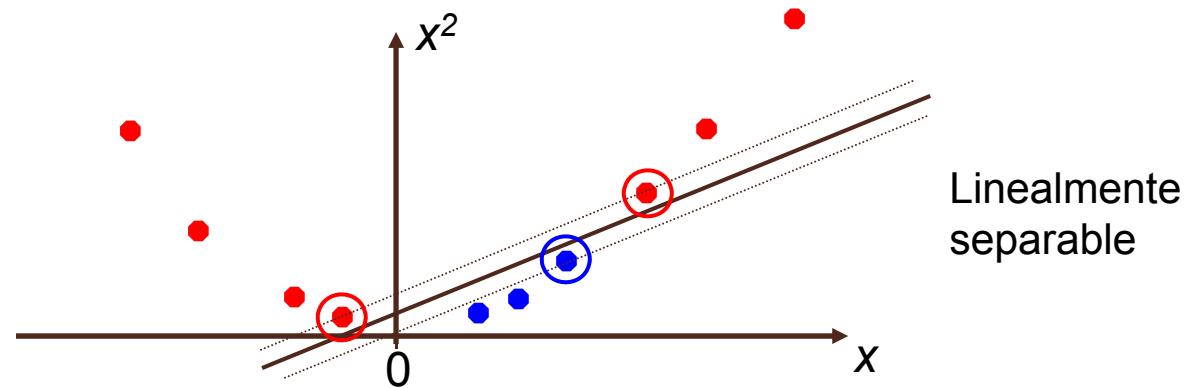
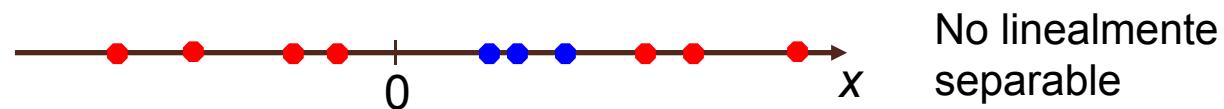
# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales
  - Solución: mapear los datos a un espacio de mayor dimensión donde el conjunto de entrenamiento sea separable



# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales
  - Por ejemplo:





# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales

- Los SVM lineales se basan en el producto escalar:

$$f(x) = \sum_{i=1}^l \alpha_i y_i x_i^T x + b$$

- La idea es proyectar cada patrón por medio de una transformación  $\Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x})$  a un nuevo espacio de mayor dimensionalidad de tal manera que en ese nuevo espacio los patrones sean linealmente separables
    - o se pueda aplicar una separación lineal tolerando cierto error
    - Se aplican las ecuaciones anteriores (separación lineal) en ese nuevo espacio:

$$f(x) = \sum_{i=1}^l \alpha_i y_i \varphi(x_i)^T \varphi(x) + b$$

- Se realiza producto escalar en el nuevo espacio
    - ¿Cuál es esta transformación  $\Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x})$  ?



# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales

- En lugar de definir esta transformación  $\Phi: \mathbf{x} \rightarrow \Phi(\mathbf{x})$ , se puede definir el producto escalar en este nuevo espacio, pero operando con los valores en el espacio original:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$

- Una función de *kernel* es una función que es equivalente a un producto escalar en un espacio determinado
    - Pero con los valores de  $\mathbf{x}_i$  y  $\mathbf{x}_j$  en el espacio original



# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales

- Por ejemplo, con vectores de 2 dimensiones, se podría definir la función de kernel  $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$
- Demostración de que  $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$  es una función de kernel:
  - Es necesario definir la función  $\Phi(\mathbf{x})$  tal que  $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$
  - $$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2 = 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2} = \\ = [1 \ x_{i1}^2 \ \sqrt{2} x_{i1} x_{i2} \ x_{i2}^2 \ \sqrt{2} x_{i1} \ \sqrt{2} x_{i2}]^T [1 \ x_{j1}^2 \ \sqrt{2} x_{j1} x_{j2} \ x_{j2}^2 \ \sqrt{2} x_{j1} \ \sqrt{2} x_{j2}] = \\ = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$
 donde  $\Phi(\mathbf{x}) = [1 \ x_1^2 \ \sqrt{2} x_1 x_2 \ x_2^2 \ \sqrt{2} x_1 \ \sqrt{2} x_2]$



# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales
  - El truco del kernel (1/2):
    - En el espacio original, se calcularía el producto escalar
      - $x_i^T x$
    - Sin embargo, como el problema es complejo los vectores se proyectan en un nuevo espacio
      - $\Phi: x \rightarrow \varphi(x)$
    - Por lo tanto, será necesario calcular el producto escalar en ese espacio  $\varphi(x_i)^T \varphi(x)$
    - Implicaría proyectar ambos vectores al nuevo espacio y calcular su producto escalar



# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales

- El truco del kernel (2/2):

- Sin embargo, se puede definir una función que calcule el producto escalar en el nuevo espacio  $\varphi(x_i)^T \varphi(x)$ , a partir de los vectores originales
      - De esta forma, no se necesita calcular las proyecciones en el nuevo espacio
        - No se necesita calcular  $\varphi(x)$  ni conocer el nuevo espacio
      - Función de kernel:  $\varphi(x_1)^T \varphi(x_2) = K(x_1, x_2)$
    - Por tanto, la función de kernel implícitamente mapea los datos a un espacio de mayor dimensionalidad
      - Sin necesidad de computar explícitamente cada  $\varphi(x)$
      - “El truco del kernel”, común a todos los métodos basados en kernels



# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*
  - Son una serie de métodos de reconocimiento y análisis de patrones, conocidos fundamentalmente por los SVM
  - La característica principal de estos métodos es la aproximación distinta que utilizan
    - Estos métodos mapean los datos en espacios de más dimensiones con la esperanza de que en este espacio los datos se vuelvan más fácilmente separables, o mejor estructurados
    - No hay restricciones en la forma de mapear
    - Esta función de mapear, sin embargo, apenas necesita ser computada gracias a una herramienta llamada **el truco del kernel**



# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*
  - El truco del kernel es una herramienta matemática que puede ser aplicada a cualquier algoritmo que **solamente dependa del producto escalar de dos vectores**
  - Cuando un producto escalar se vaya a realizar, se reemplaza por una función de kernel
  - Si se aplican adecuadamente, esos **algoritmos lineales candidatos se transforman en algoritmos no lineales**
    - Esos algoritmos no lineales son equivalentes a los originales lineales que operan en el espacio  $\Phi$
    - Sin embargo, dado que se usan los kernels, la función  $\Phi(\mathbf{x})$  no necesita ser explícitamente computada

# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*
  - Teorema de Mercer:
    - Todas las funciones simétricas definidas semipositivas son kernels
  - Las funciones simétricas definidas semipositivas se corresponden a una matriz de Gram simétrica definida semipositiva:

K=

$K(\mathbf{x}_1, \mathbf{x}_1)$	$K(\mathbf{x}_1, \mathbf{x}_2)$	$K(\mathbf{x}_1, \mathbf{x}_3)$	...	$K(\mathbf{x}_1, \mathbf{x}_n)$
$K(\mathbf{x}_2, \mathbf{x}_1)$	$K(\mathbf{x}_2, \mathbf{x}_2)$	$K(\mathbf{x}_2, \mathbf{x}_3)$		$K(\mathbf{x}_2, \mathbf{x}_n)$
...	...	...	...	...
$K(\mathbf{x}_n, \mathbf{x}_1)$	$K(\mathbf{x}_n, \mathbf{x}_2)$	$K(\mathbf{x}_n, \mathbf{x}_3)$	...	$K(\mathbf{x}_n, \mathbf{x}_n)$



# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*
  - Las funciones de kernel deben de ser continuas, simétricas, y, preferiblemente, tener una matriz de Gram semidefinida positiva
  - Los kernels que satisfacen el teorema de Mercer son semidefinidos positivos
    - Esto quiere decir que sus matrices de kernel no tienen valores propios no negativos
    - El uso de un kernel definido positivo garantiza que el problema de optimización convergirá y la solución será única
    - Sin embargo, muchas funciones de kernels no son estrictamente definidas positivas, pero han mostrado un buen comportamiento en distintos problemas



# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*
  - Escoger un kernel apropiado depende del problema que se quiera resolver
  - Por ejemplo:
    - Un kernel polinómico permite modelizar conjunciones de los valores hasta el orden del polinomio
    - Una función de base radial (*radial basis function*) permite construir círculos (o hiperesferas)
    - Un kernel lineal permite construir líneas (o hiperplanos)
  - Además, ajustar los parámetros puede ser un proceso tedioso



# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*

- Algunos kernels más usados son:

- Lineal: el más sencillo.  $\varphi(x) = x$
    - Los algoritmos kernel que utilizan este tipo, suelen ser iguales a los correspondientes no-kernel
    - Por ejemplo, KPCA (kernel PCA, versión de PCA con kernels) es igual que PCA cuando el kernel es lineal
    - Un SVM con kernel lineal es igual que un SVM lineal

$$k(x, y) = x^T y + c$$

- Polinómico.

- Son adecuados para problemas donde todos los datos de entrenamiento están normalizados

$$k(x, y) = (\alpha x^T y + c)^d$$

- Parámetros ajustables son la pendiente  $\alpha$ , el término constante  $c$ , y el grado del polinomio  $d$

# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*
  - Algunos kernels más usados son:
    - Gausiano: este es un ejemplo de kernel de función de base radial

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

$\varphi(x)$  es infinito dimensional: cada punto se mapea a una función (Gausiana)

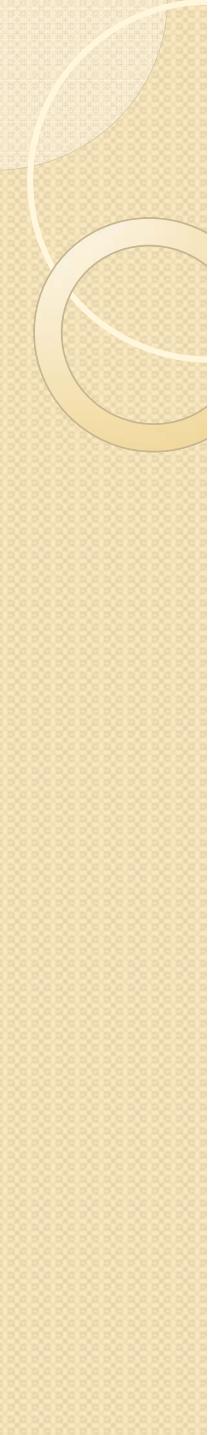
- El parámetro sigma juega un papel importante en el funcionamiento del kernel, y su valor debe de ser cuidadosamente fijado:
  - Si se sobreestima, la exponencial se comportará casi linealmente, y la proyección en un espacio de más dimensiones empezará a perder su poder no lineal
  - Por la contra, si toma valores demasiado bajos, los límites de decisión se volverán muy sensibles al ruido en los datos de entrenamiento
- Exponencial: también es un kernel de función de base radial
  - Muy parecido al anterior

$$k(x, y) = \exp\left(-\frac{\|x - y\|}{2\sigma^2}\right)$$



# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*
  - Algunos kernels más usados son:
    - Laplaciano: también kernel de función de base radial
      - Totalmente equivalente al exponencial, excepto por ser menos sensible a los cambios en el parámetro sigma
$$k(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma}\right)$$
      - Las observaciones realizadas sobre el parámetros sigma en el kernel Gausiano también se pueden aplicar al exponencial y al laplaciano.
    - Hiperbólico tangente
      - También se conoce como kernel sigmoidal, o kernel perceptrón multicapa
      - Viene del mundo de las RR.NN.AA. (se usa como función de transferencia)
      - Un SVM que utilice un kernel sigmoidal es equivalente a un perceptrón de dos capas
$$k(x, y) = \tanh(\alpha x^T y + c)$$
      - Dos parámetros: la pendiente  $\alpha$ , y la constante  $c$ .
        - Un valor común para alfa suele ser  $1/N$ , donde  $N$  es el número de dimensiones.



# MÁQUINAS DE SOPORTE VECTORIAL

- Métodos basados en *kernels*

- Otros kernels:
  - ANOVA
  - Cuadrático racional
  - Multicuadrático
  - Multicuadrático inverso
  - Circular
  - Esférico
  - Logarítmico
  - Spline
  - B-Spline
  - Cauchy
  - Chi-Cuadrado
  - Bayesiano
  - Wavelet
  - etc.

# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales

- La formulación anterior:

Encontrar  $w$  y  $b$  tal que

minimizar:  $\frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i$  sujeto a:  $y_i(w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0$   
cambiaría a:

Encontrar  $w$  y  $b$  tal que

minimizar:  $\frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i$  sujeto a:  $y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i \quad \xi_i \geq 0$

- $\varphi(x)$  realiza una transformación de los datos a un espacio de mayor dimensionalidad
    - Se busca el hiperplano que separe los datos en ese nuevo espacio
    - Sin embargo, no es necesario conocer esta transformación
      - Es suficiente con conocer una función, kernel, que calcule el producto escalar

# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales
  - El problema dual se transformaría:

Encontrar  $\alpha_1 \dots \alpha_N$  tal que maximicen:

$$\sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j y_i y_j \varphi(x_i) \varphi(x_j)$$

sujeto a :  $\sum_{i=1}^L y_i \alpha_i = 0 \quad 0 < \alpha_i < C \quad i = 1, \dots, L$


  - Encontrar  $\alpha_1 \dots \alpha_N$  tal que maximicen:
$$\sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

sujeto a :  $\sum_{i=1}^L y_i \alpha_i = 0 \quad 0 < \alpha_i < C \quad i = 1, \dots, L$

# MÁQUINAS DE SOPORTE VECTORIAL

- SVM no lineales
  - Solución al problema:
    - Función de decisión:

$$f(x) = \sum_{i=1}^l \alpha_i y_i \varphi(x_i)^T \varphi(x) + b$$

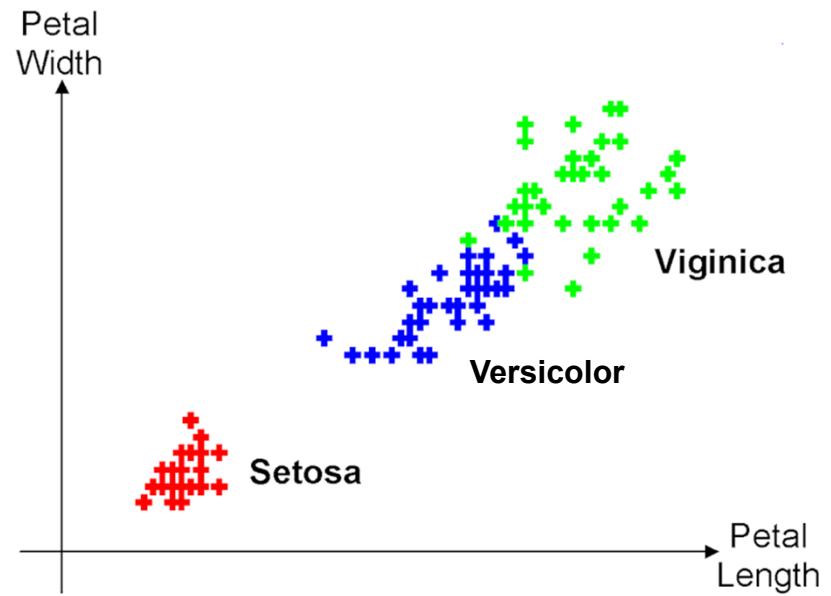


$$f(x) = \sum_{i=1}^l \alpha_i y_i K(x_i, x) + b$$

- La técnica de optimización para hallar los  $\alpha_i$  sigue siendo la misma

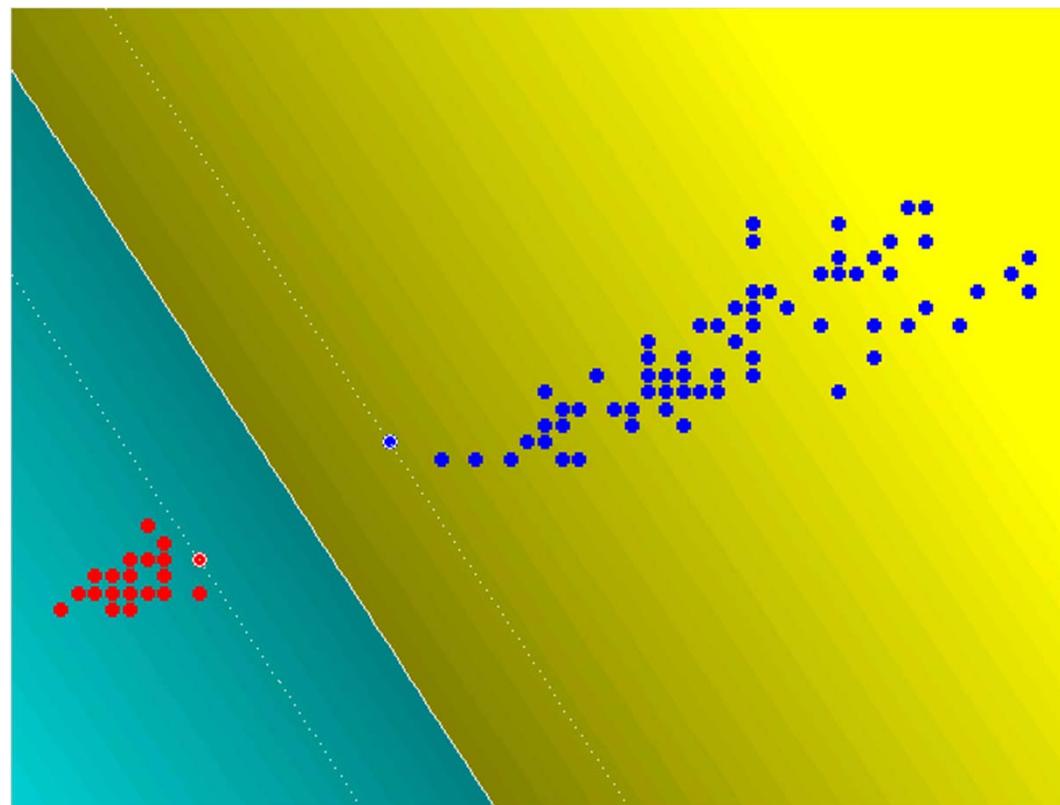
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:
  - Conocida base de datos de clasificación
  - Contiene datos de 150 flores iris, cada una con 4 atributos, clasificables en 3 posibles clases: setosa, virginica y versicolor
    - Los patrones se pueden representar mediante los 2 atributos que contienen más información
    - Se entrena con esos dos atributos
  - Estrategia “uno contra el resto”



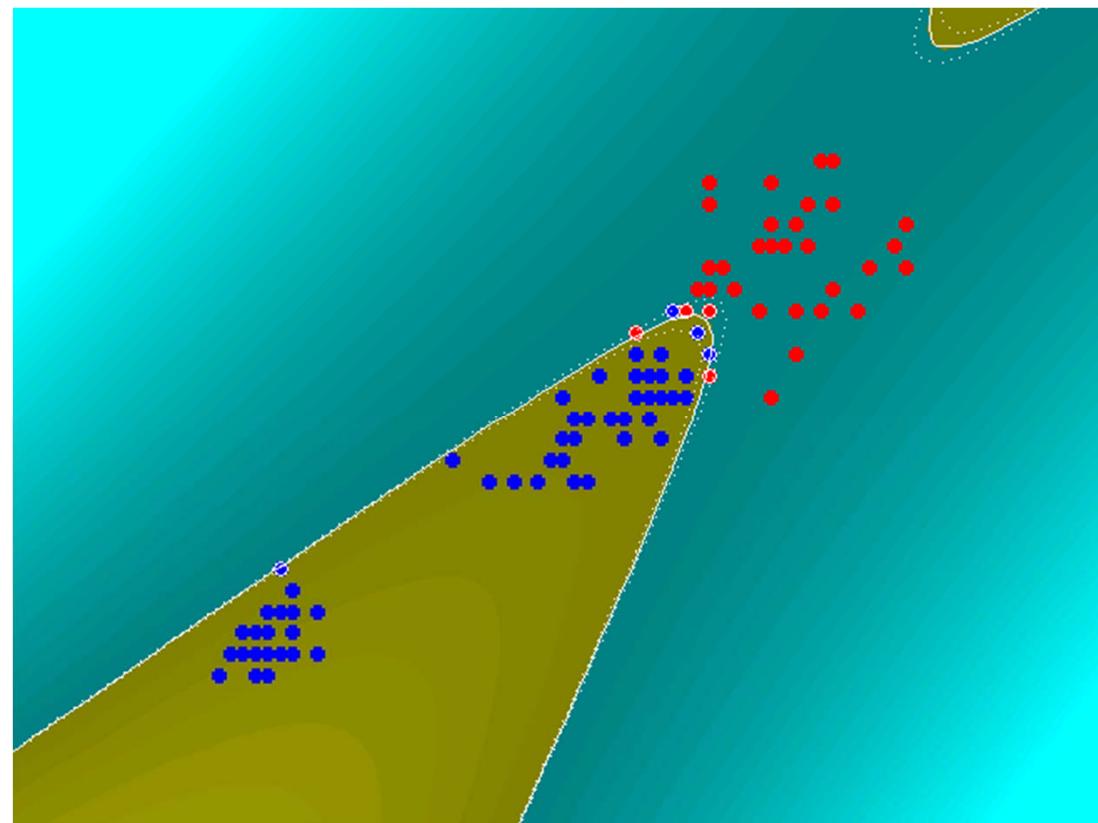
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:
  - Separación de “setosa” mediante un SVM lineal ( $C=\infty$ )



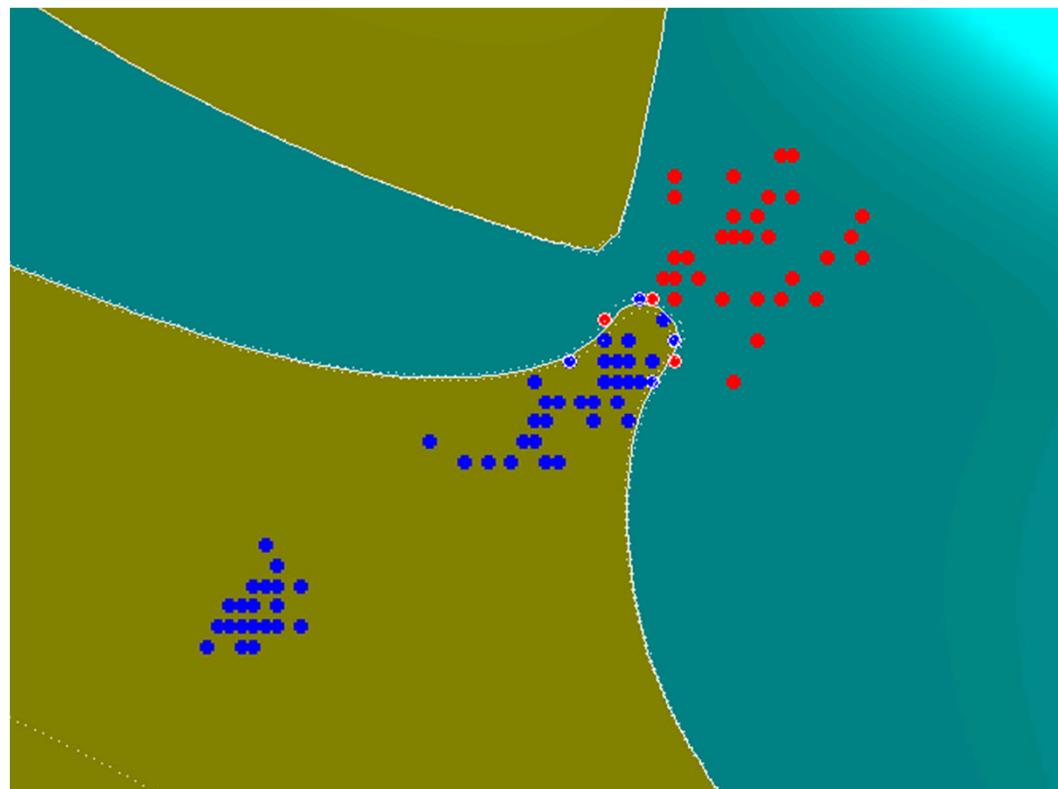
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:
  - Separación de “Virginica” con un SVM polinómico (grado 2,  $C = \inf$ )



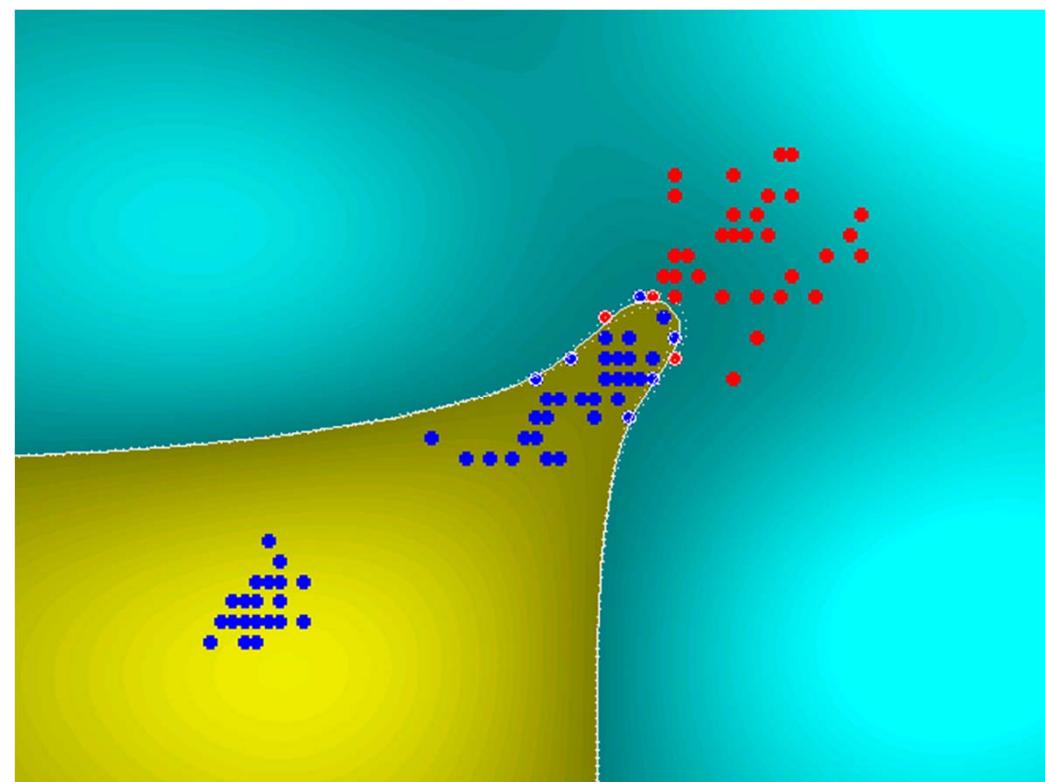
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:
  - Separación de “Virginica” con un SVM polinómico (grado 10,  $C = \inf$ )
    - Hiperplano en un espacio de 55 dimensiones
    - Sobreajustado



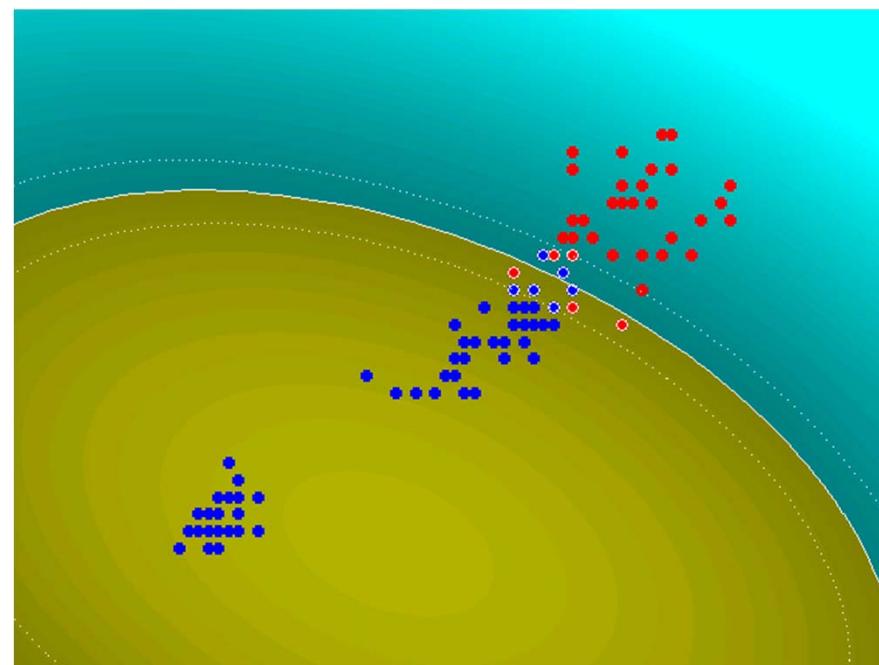
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:
  - Separación de “Virginica” con un SVM de función de base radial ( $\sigma = 1.0$ ,  $C = \infty$ )
    - Bastante similar al polinómico de grado 2



# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:
  - “Virginica” con un SVM polinómico (grado 2,  $C = 10$ )
    - $C=10 \rightarrow$  Alguna tolerancia a errores en la clasificación
    - Parece dar una buena generalización
      - Esto enfatiza la importancia de tolerar ciertos errores en la clasificación
      - Esto es necesario debido a la naturaleza no separable de los datos usando sólo dos atributos





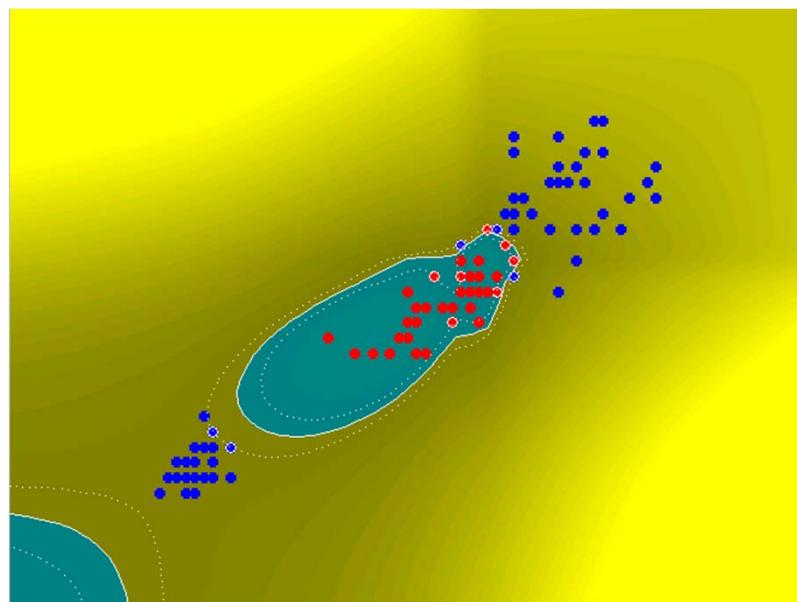
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:
  - Para mostrar el efecto de la tolerancia a los errores en la clasificación en los límites del clasificador, se muestran distintos ejemplos
    - Spline lineal, con varios grados de tolerancia
    - Separación de “Versicolor”
    - Valores altos de C dan unos límites más cerrados

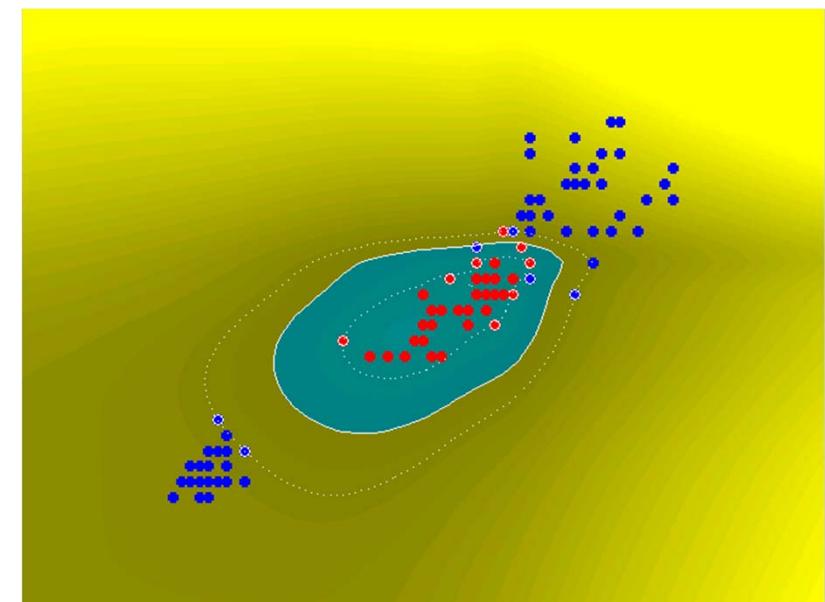
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:

$C = \inf$



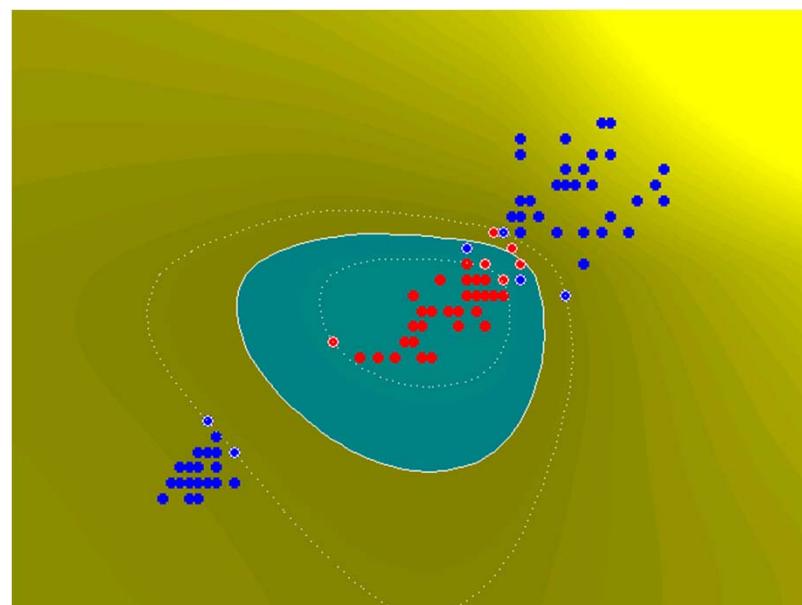
$C = 1000$



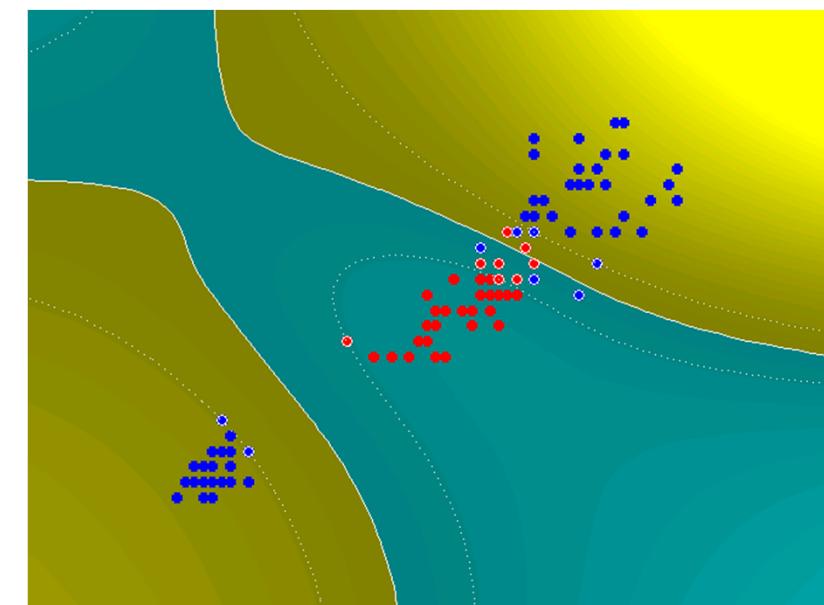
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:

$C = 100$

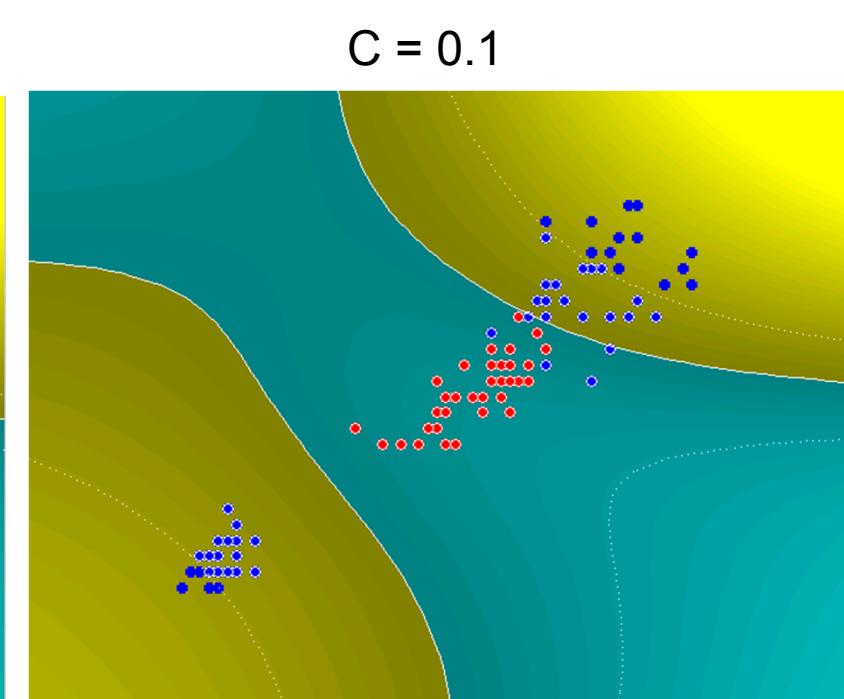
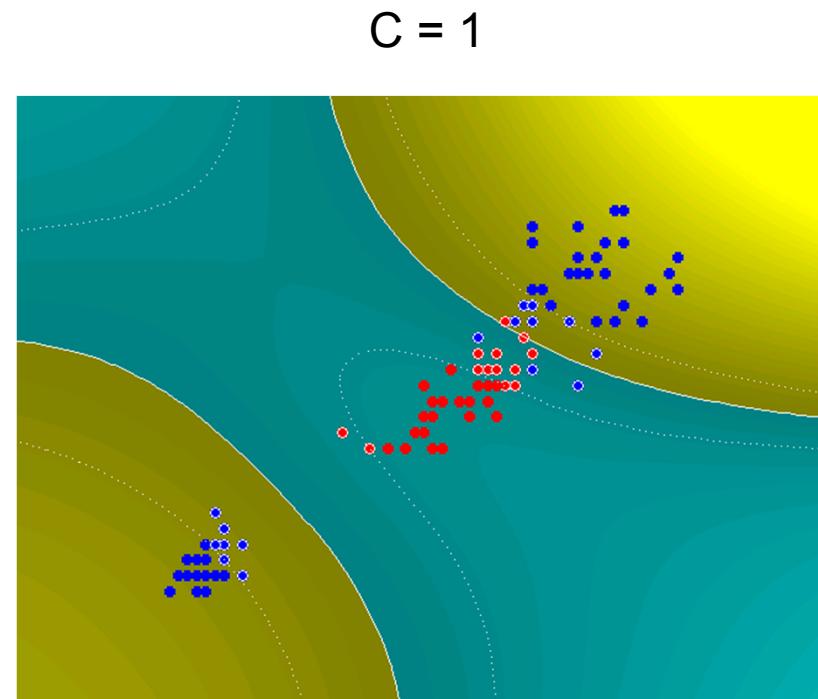


$C = 10$



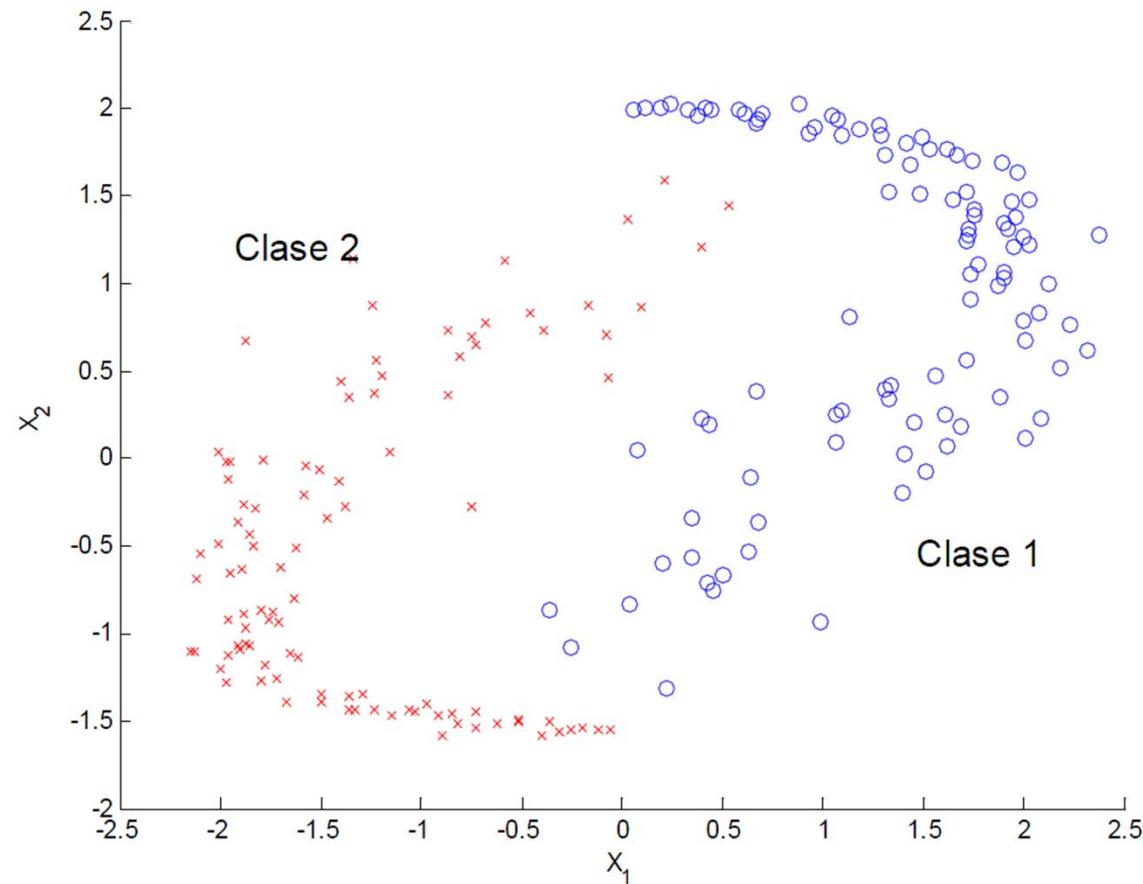
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Clasificación de flores iris:



# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Doble espiral:
  - 200 puntos en dos clases, no linealmente separables



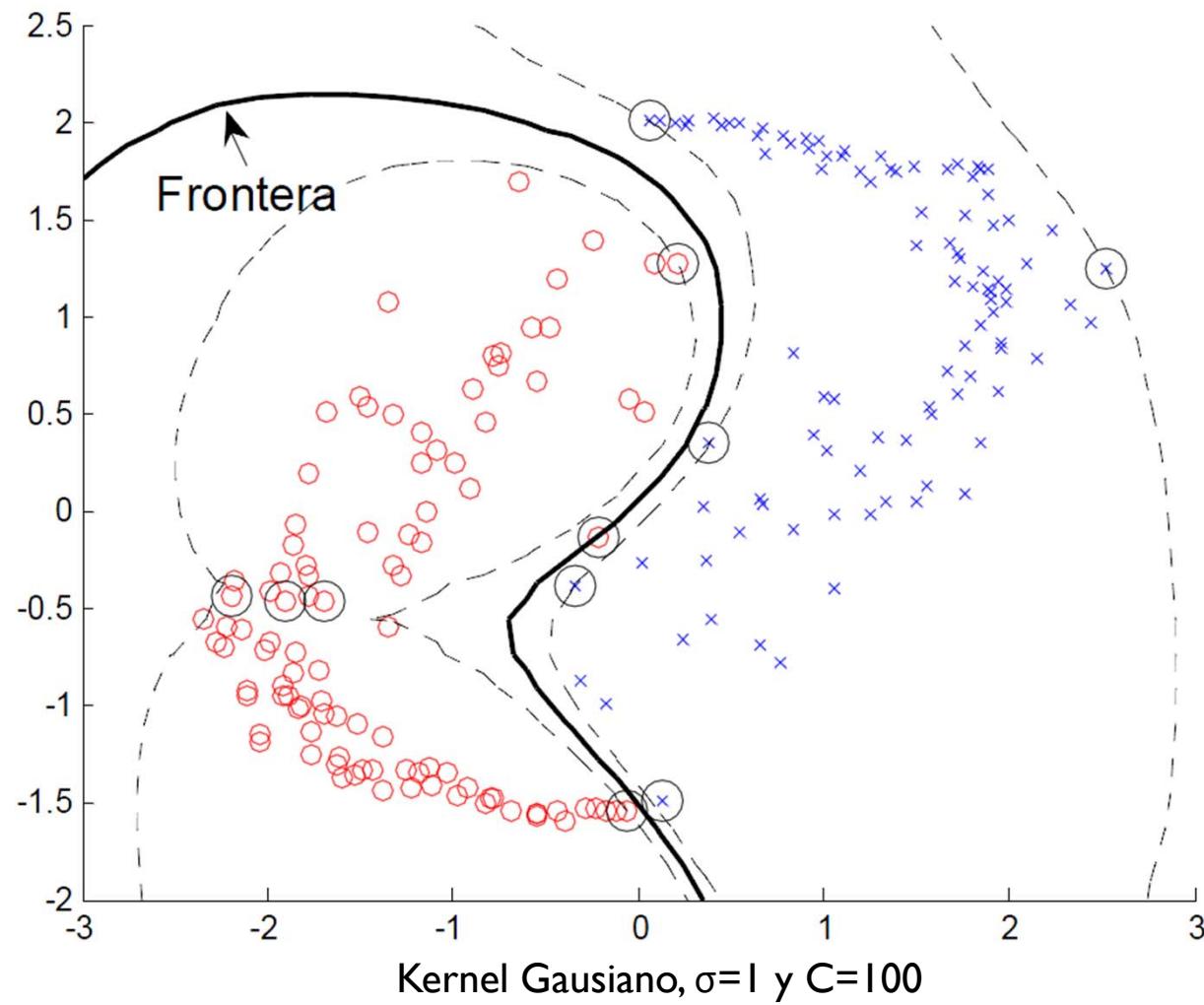


# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Doble espiral:
  - kernel Gaussiano con  $\sigma=1$  y  $C=100$
  - Entrenado el SVM con el algoritmo SMO (*Sequential Minimal Optimizer*)
  - Resultados:
    - $\alpha = [4.3475 \ 0.1072 \ 19.8084 \ 75.8903 \ 86.2008 \ -94.5622 \ -0.5096 \ -0.1663 \ -1.2177 \ -84.3138 \ -5.5846]$
    - $b = 0.1484$
    - 11 vectores de soporte: 0.0625 2.0007  
2.5263 1.2379  
0.3844 0.3465  
-0.3321 -0.3822  
0.1254 -1.4875  
-0.0573 -1.5489  
-2.1932 -0.4391  
-1.6895 -0.4672  
-1.9075 -0.4692  
-0.2136 -0.1409  
0.2116 1.2649

# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Doble espiral:





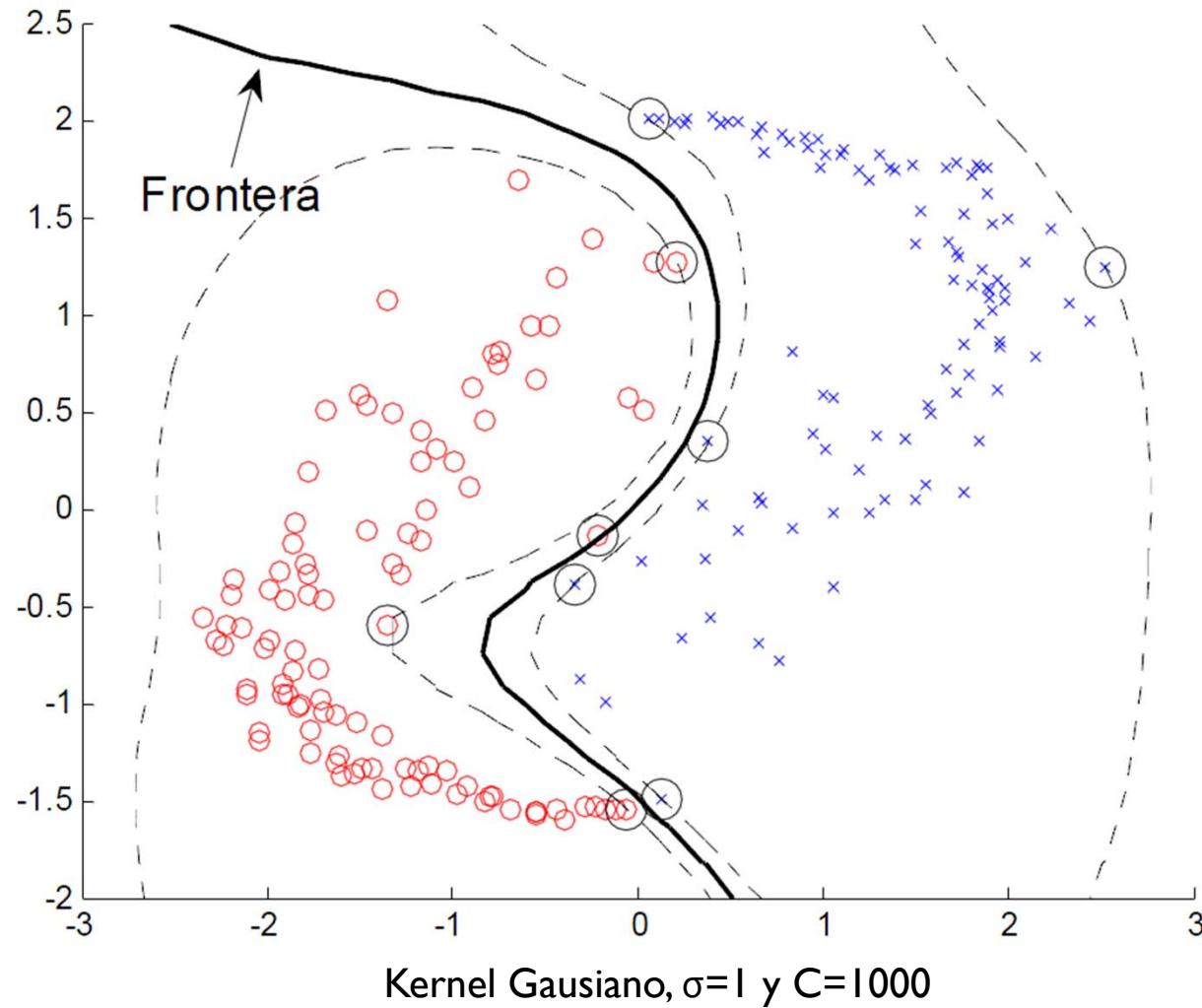
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Doble espiral:
  - Se refina más la frontera de decisión con kernel Gaussiano con  $\sigma=1$  y  $C=1000$
  - Resultados:
    - $\alpha = [5.0305 \ 0.1825 \ 22.7773 \ 85.4252 \ 60.4037 \ -69.9884 \ -4.1113 \ -93.0078 \ -6.7117]$
    - $b = -0.0223$
    - 9 vectores de soporte:

0.0625 2.0007  
2.5263 1.2379  
0.3844 0.3465  
-0.3321 -0.3822  
0.1254 -1.4875  
-0.0573 -1.5489  
-1.3457 -0.6016  
-0.2136 -0.1409  
0.2116 1.2649

# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Doble espiral:



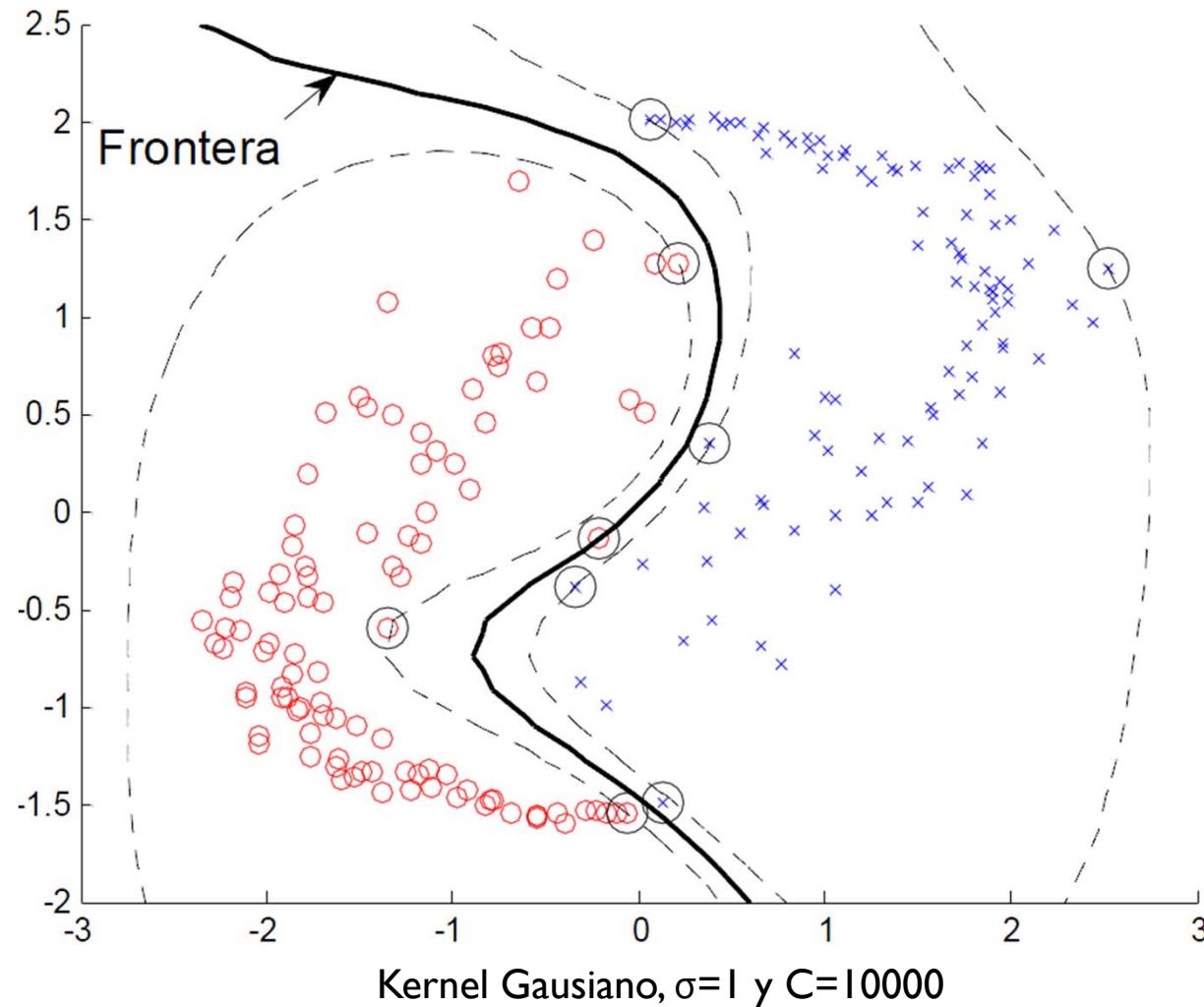
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Doble espiral:
  - Se refina más la frontera de decisión con kernel Gaussiano con  $\sigma=1$  y  $C=10000$
  - Resultados:
    - $\alpha = [5.4094 \ 0.2671 \ 22.1051 \ 80.4316 \ 49.6585 \ -58.8928 \ -4.3958 \ -87.2509 \ -7.3322]$
    - $b = -0.0552$
    - 9 vectores de soporte (los mismos que antes):

0.0625 2.0007  
2.5263 1.2379  
0.3844 0.3465  
-0.3321 -0.3822  
0.1254 -1.4875  
-0.0573 -1.5489  
-1.3457 -0.6016  
-0.2136 -0.1409  
0.2116 1.2649

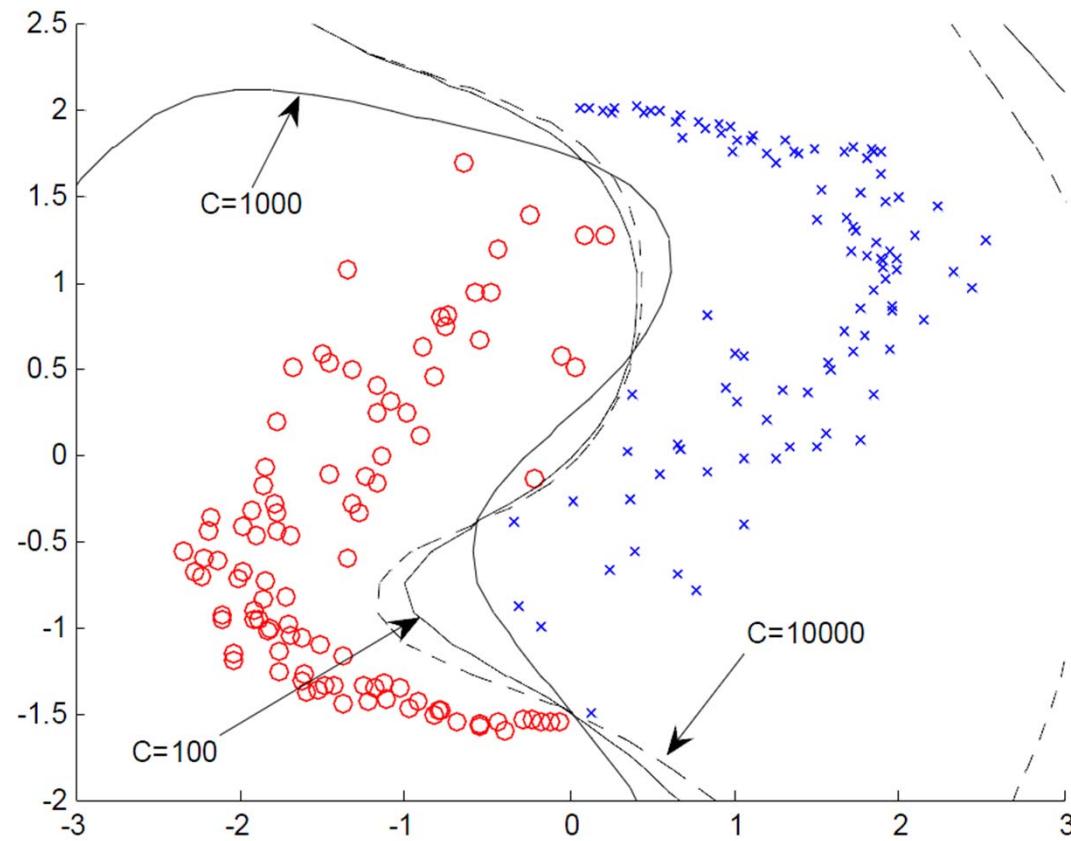
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Doble espiral:



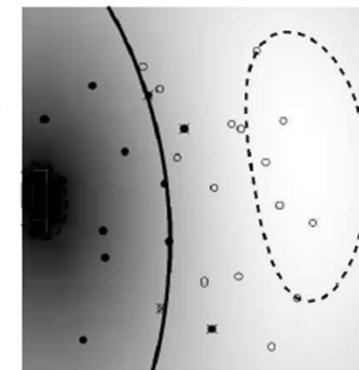
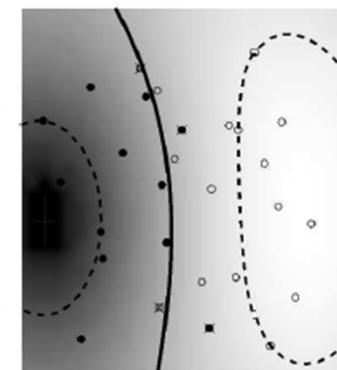
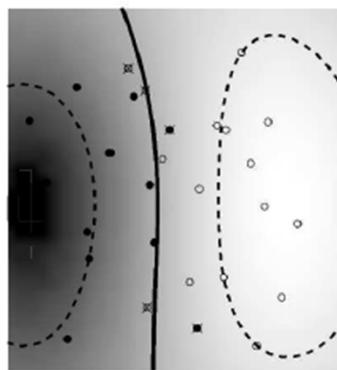
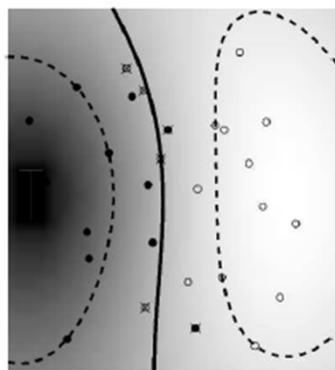
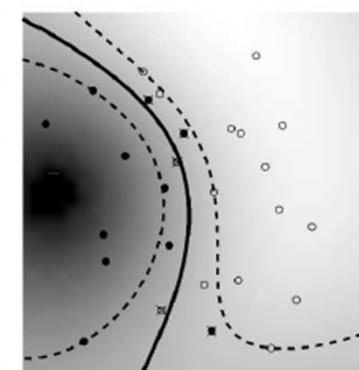
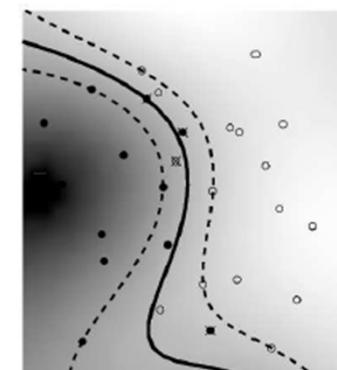
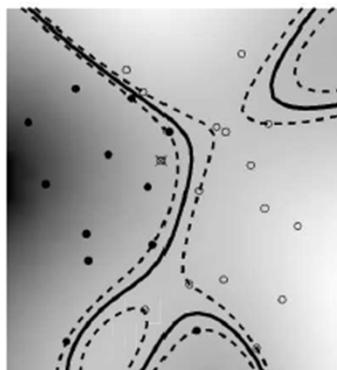
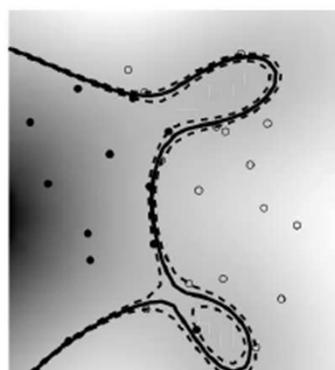
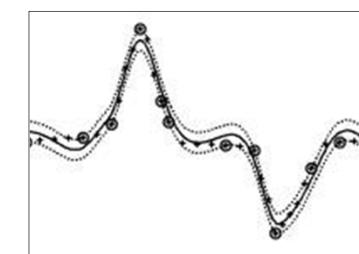
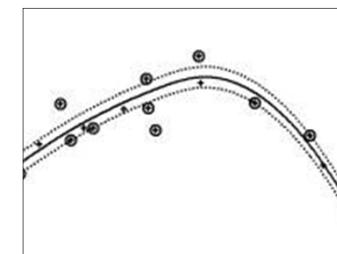
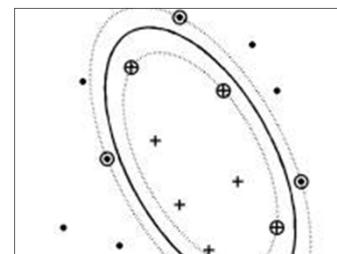
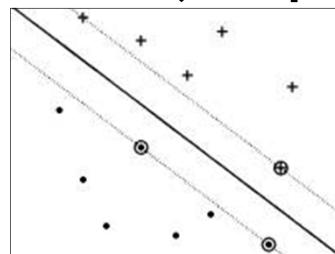
# MÁQUINAS DE SOPORTE VECTORIAL

- Ejemplo: Doble espiral:
  - Uniendo las tres gráficas en una:
    - “Idealmente”, la frontera óptima se logra cuando  $C = \text{infinito}$



# MÁQUINAS DE SOPORTE VECTORIAL

- Más ejemplos:





# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:

- El planteamiento para un problema de clasificación es:

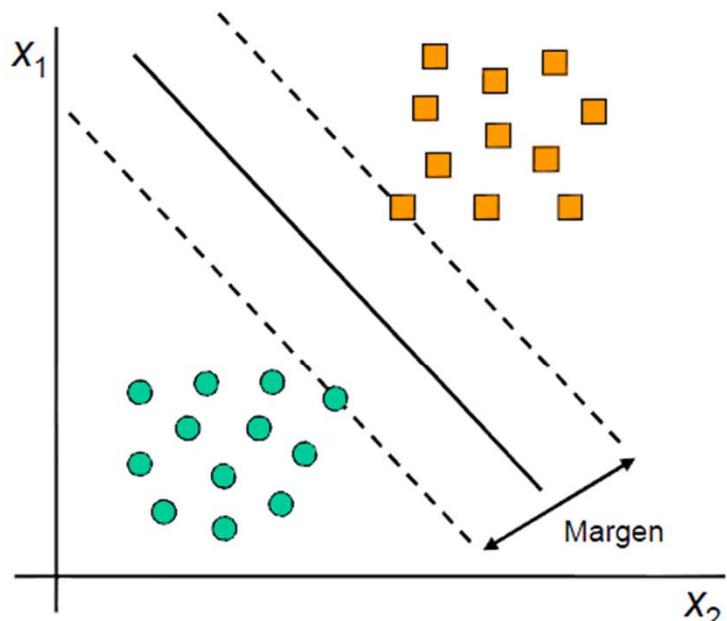
$$\text{minimizar: } \|\mathbf{w}\|^2 \quad \text{sujeto a : } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall i = 1, \dots, l$$

- En cambio, para un problema de regresión es:

$$\text{minimizar: } \|\mathbf{w}\|^2 \quad \text{sujeto a : } \begin{aligned} (\mathbf{w}^T \mathbf{x}_i + b) - y_i &\leq \varepsilon \\ y_i - (\mathbf{w}^T \mathbf{x}_i + b) &\leq \varepsilon \end{aligned} \quad \forall i = 1, \dots, l$$

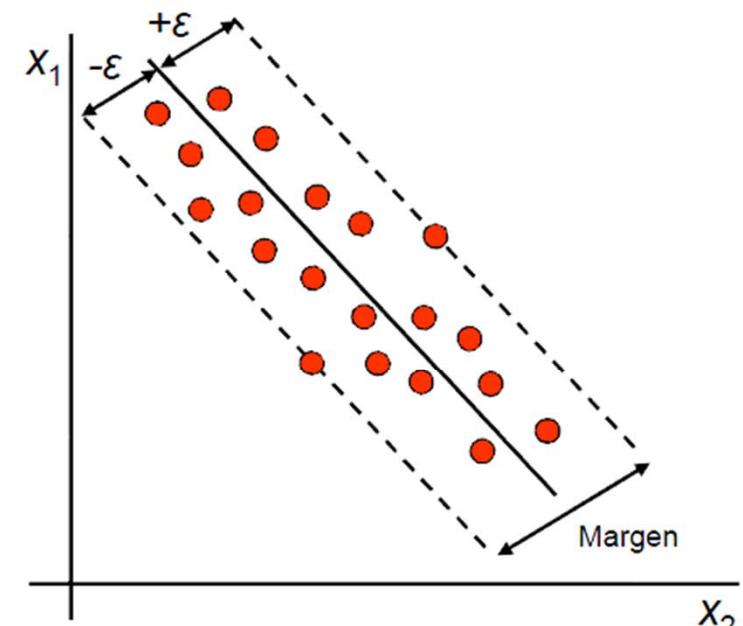
# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:
  - Diferencia del planteamiento:



## Clasificación:

Todos los datos quedan fuera del margen

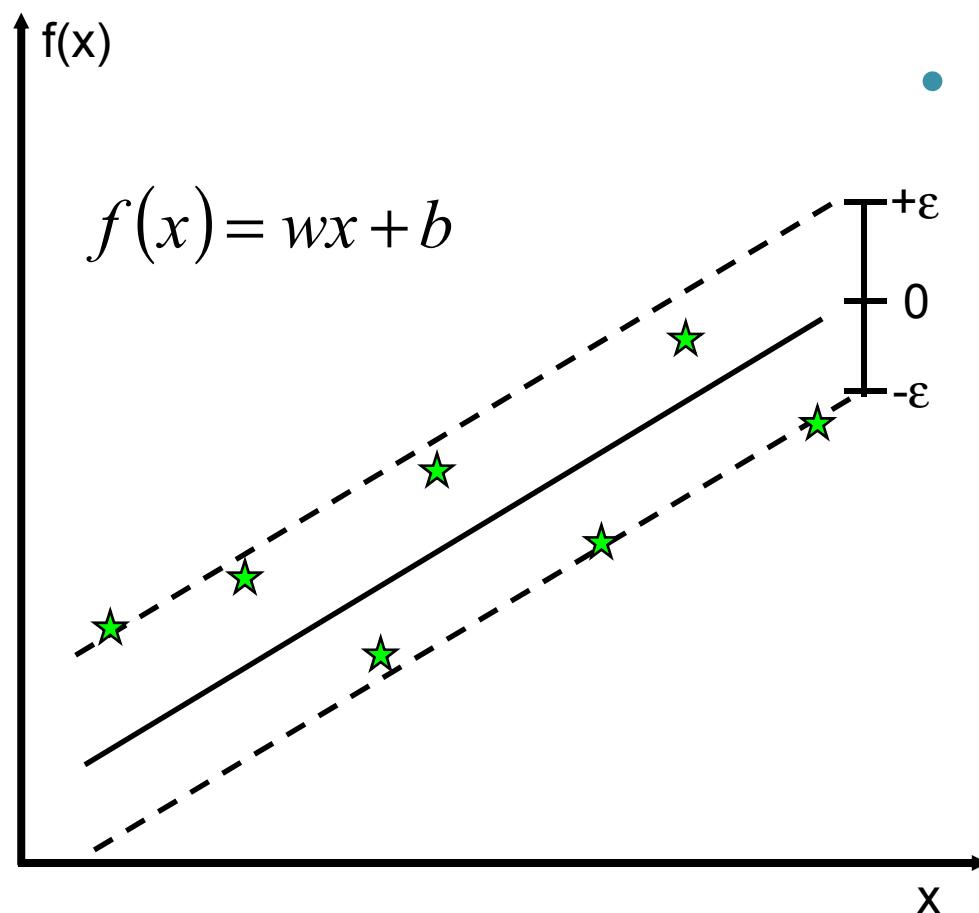


## Regresión:

Todos los datos quedan dentro del margen

# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:

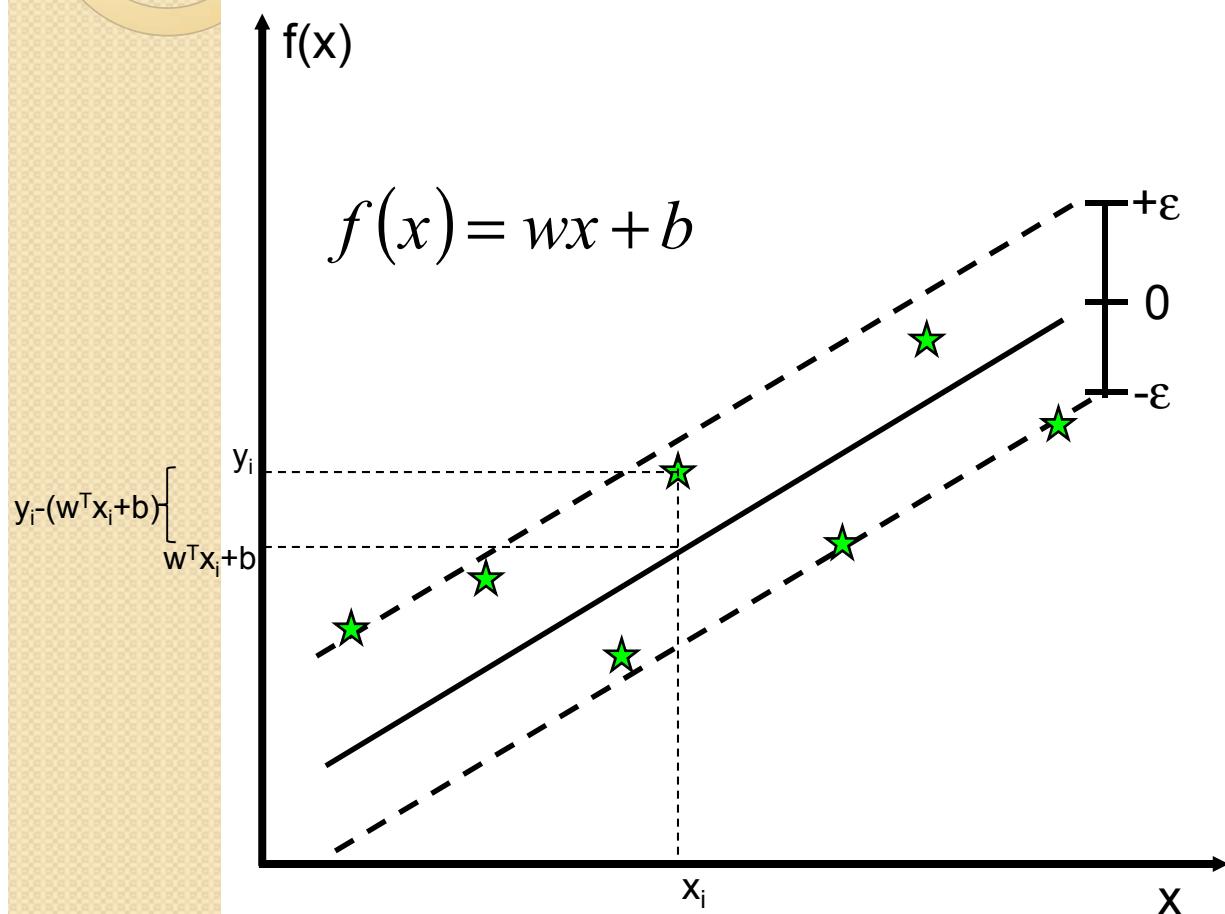


- Planteamiento:

- Encontrar el hiperplano  $f(x)=wx+b$  que aproxime lo mejor posible los puntos

# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:



- Solución:

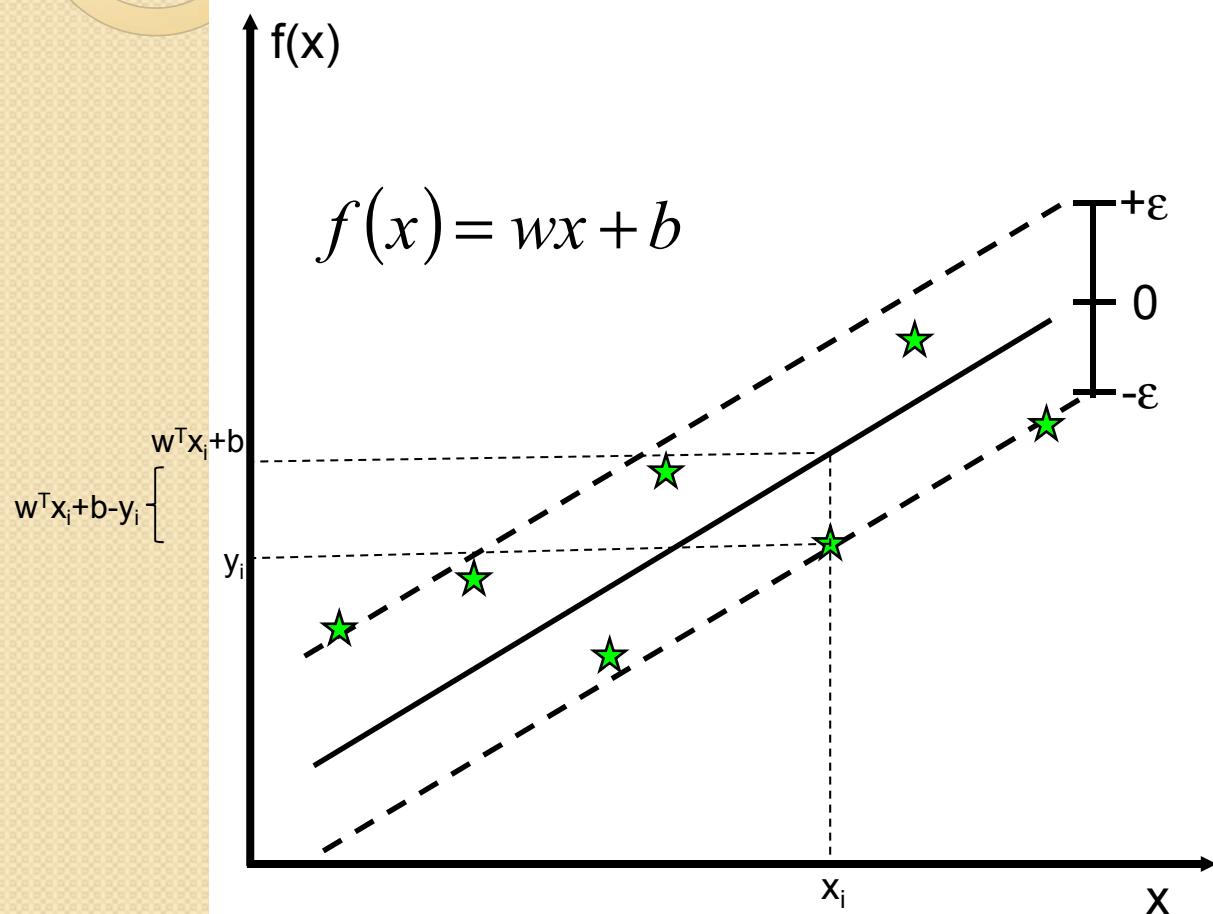
$$\text{Min} \frac{1}{2} w^T w$$

- Restricciones:

$$y_i - w^T x_i - b \leq \varepsilon$$

# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:



- Solución:

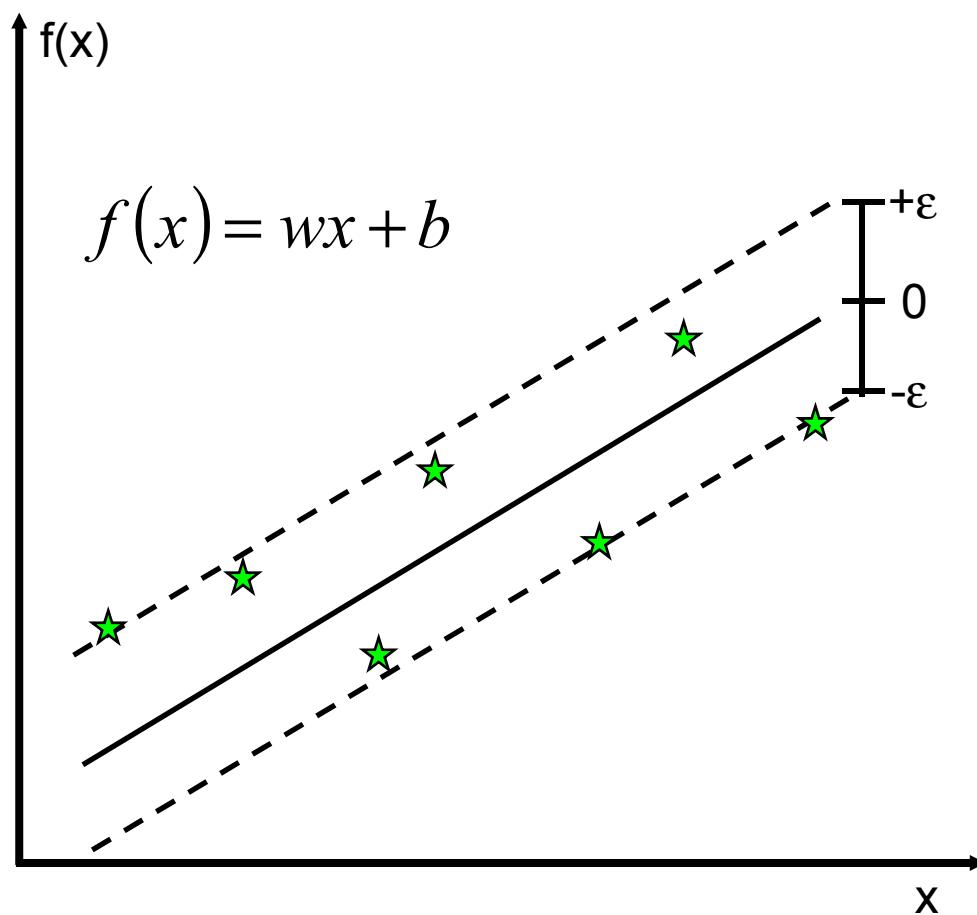
$$\text{Min} \frac{1}{2} w^T w$$

- Restricciones:

$$w^T x_i + b - y_i \leq \varepsilon$$

# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:



- Solución:

$$\text{Min} \frac{1}{2} w^T w$$

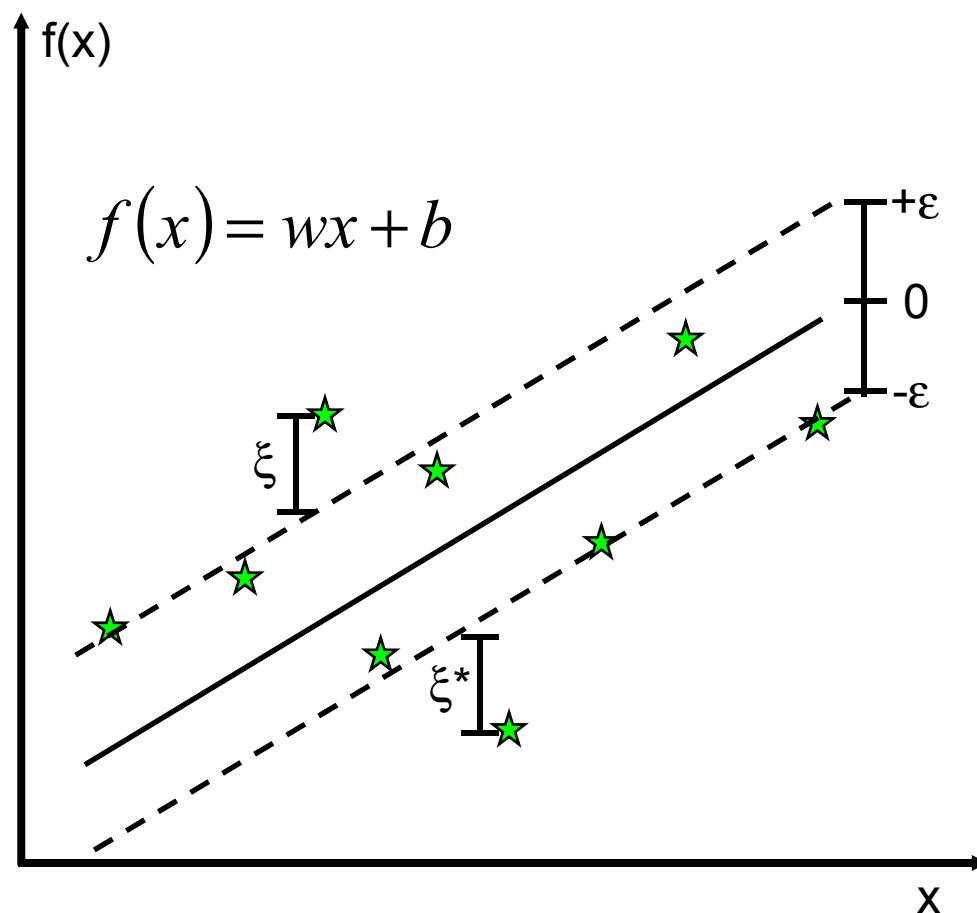
- Restricciones:

$$y_i - w^T x_i - b \leq \varepsilon$$

$$w^T x_i + b - y_i \leq \varepsilon$$

# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:



- Minimizar:

$$\frac{1}{2} w^T w + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

- Restricciones:

$$y_i - w^T x_i - b \leq \epsilon + \xi_i$$

$$w^T x_i + b - y_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:
  - Aplicando Lagrange:

$$L = \frac{1}{2} w^T w + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

Objetivo

$$- \sum_{i=1}^N \alpha_i (\varepsilon + \xi_i - y_i + w^T x_i + b)$$

$$- \sum_{i=1}^N \alpha_i^* (\varepsilon + \xi_i^* - y_i + w^T x_i + b)$$

$$- \sum_{i=1}^N (\eta_i \xi_i + \eta_i^* \xi_i^*)$$

Restricciones



# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:
  - Ecuación de regresión resultante:

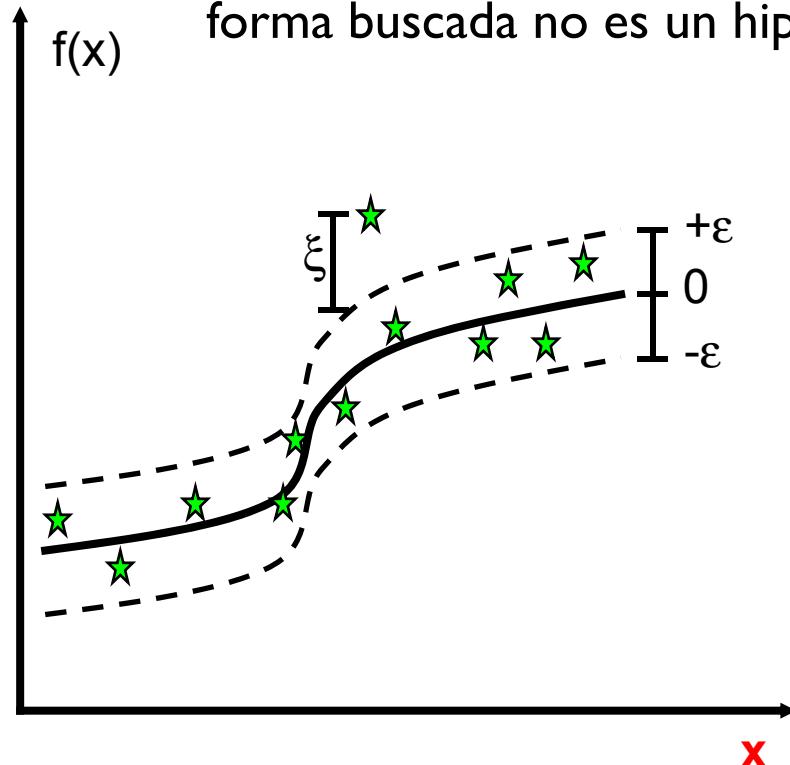
$$y(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot \langle x_i, x \rangle + b$$

$\langle x_i, x \rangle$  : producto escalar entre ambos vectores

- Extensión no lineal

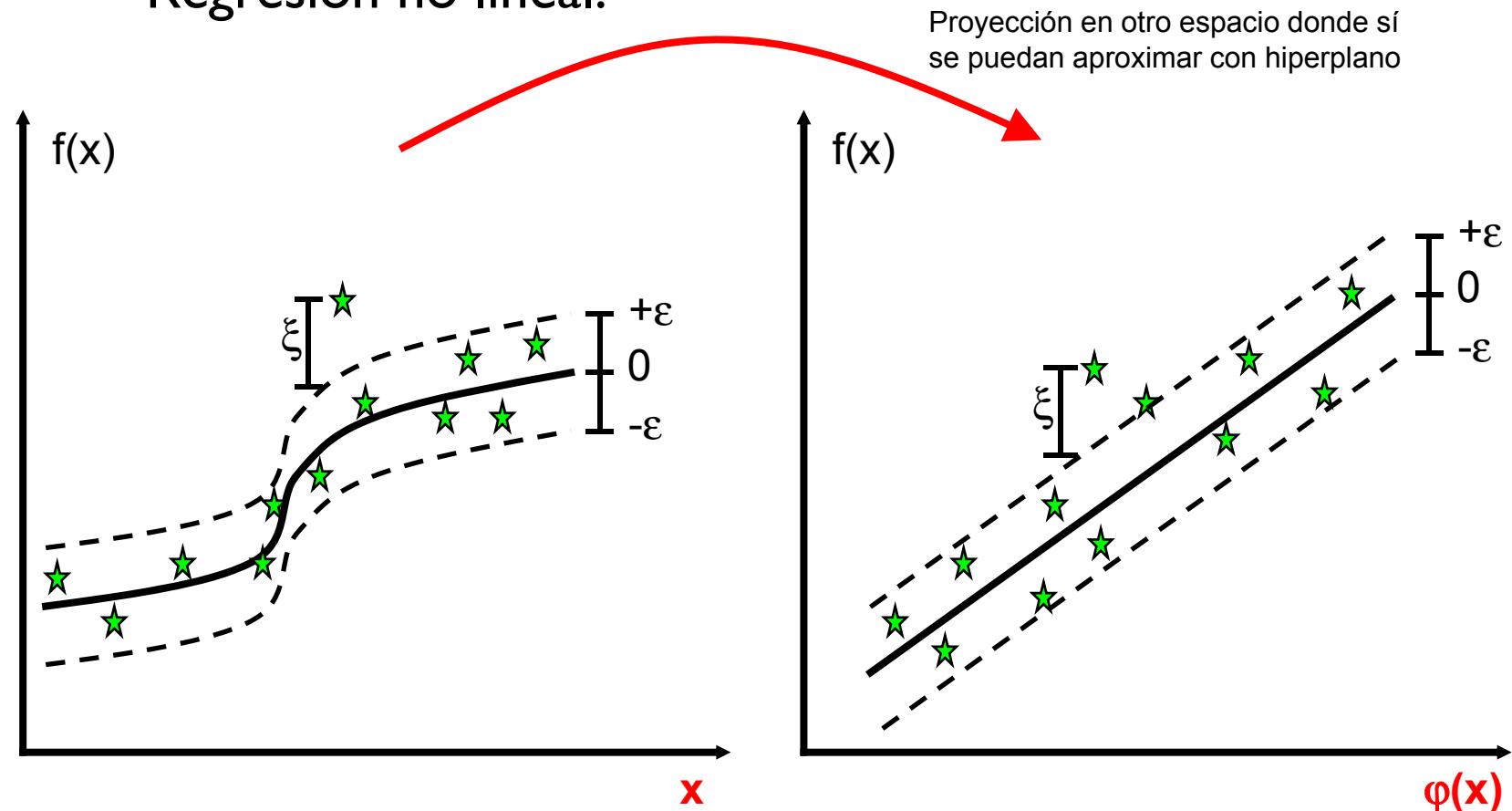
# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:
  - Regresión no lineal:
    - ¿Y si los puntos no se pueden aproximar por un hiperplano o la forma buscada no es un hiperplano?



# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:
  - Regresión no lineal:





# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:

- Fórmulas de regresión:

- Lineal:

$$y(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot \langle x_i, x \rangle + b$$

$\langle x_i, x_j \rangle$ : producto escalar entre ambos vectores

- No lineal:

$$y(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot \langle \varphi(x_i), \varphi(x) \rangle + b$$

- General:

$$y(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot K(x_i, x) + b$$

# MÁQUINAS DE SOPORTE VECTORIAL

- SVMs para regresión:

- Tipos de kernel:

- Lineal:



- Polinómico:

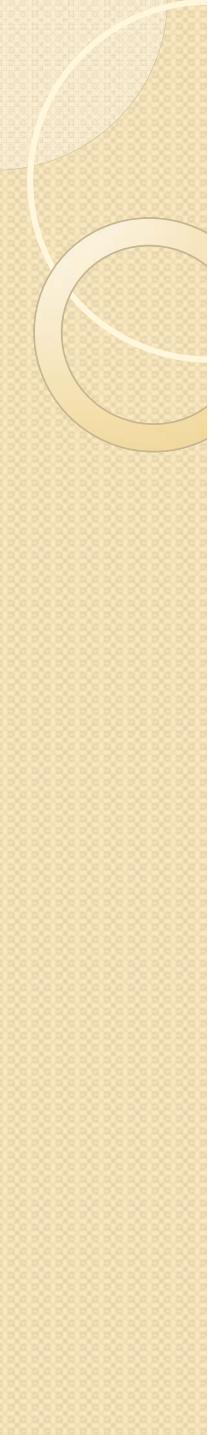
$$K(x, x_i) = \langle x, x_i \rangle^d$$

- Función de base radial:  $K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)$

- RBF exponencial:

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|}{2\sigma^2}\right)$$

- etc.



# MÁQUINAS DE SOPORTE VECTORIAL

- Los SVMs fueron propuestos originalmente por Boser, Guyon y Vapnik en 1992, y su popularidad se fue incrementando a finales de los 90.
- Los SVM actualmente están entre los sistemas clasificadores que ofrecen los mejores resultados para un gran número de tareas de clasificación, desde texto a datos genómicos.
- Los SVM se pueden aplicar a tipos de datos complejos que se basan en vectores de características (por ejemplo, gráficos, secuencias, datos relacionales) mediante el diseño de funciones de kernel para esos datos.
- Los SVM han sido extendidos para un número de tareas como regresión [Vapnik *et al.* '97], análisis de componentes principales [Schölkopf *et al.* '99], etc.
- Los algoritmos de optimización para SVM más populares utilizan descomposición para realizar un *hill-climbing* sobre un conjunto de  $\alpha_i$  de cada vez, por ejemplo SMO [Platt '99] [Joachims '99]



# MÁQUINAS DE SOPORTE VECTORIAL

- Las SVM son básicamente clasificadores para 2 clases.
  - Se puede cambiar la formulación del algoritmo QP para permitir clasificación multiclas.
  - Más comúnmente, los datos son divididos “inteligentemente” en dos partes de diferentes formas y una SVM es entrenada para cada forma de división.
  - La clasificación multiclas es hecha combinando la salida de todos los clasificadores
- El modelado de la SVM es tal que no necesita de toda la totalidad de puntos disponibles para hallar una solución al problema de maximización de la separación entre clases



# MÁQUINAS DE SOPORTE VECTORIAL

- Ventajas:

- El entrenamiento es relativamente fácil.
- No hay óptimo local, como en las redes neuronales.
- Se escalan relativamente bien para datos en espacios dimensionales altos.
- El compromiso entre la complejidad del clasificador y el error puede ser controlado explícitamente.
- Datos no tradicionales como cadenas de caracteres y árboles pueden ser usados como entrada a la SVM, en vez de vectores de características.



# MÁQUINAS DE SOPORTE VECTORIAL

- Debilidades:
  - Se necesita una “buena” función *kernel*, es decir, se necesitan metodologías eficientes para sintonizar los parámetros de inicialización de la SVM.
  - Ajustar los parámetros de los SVM continúa siendo un proceso laborioso
    - Seleccionar un kernel específico y los parámetros generalmente se realiza mediante prueba y error.