TEMA 8
MÁQUINAS DE VECTORES SOPORTE
SUPPORT VECTOR MACHINES (SVMs)

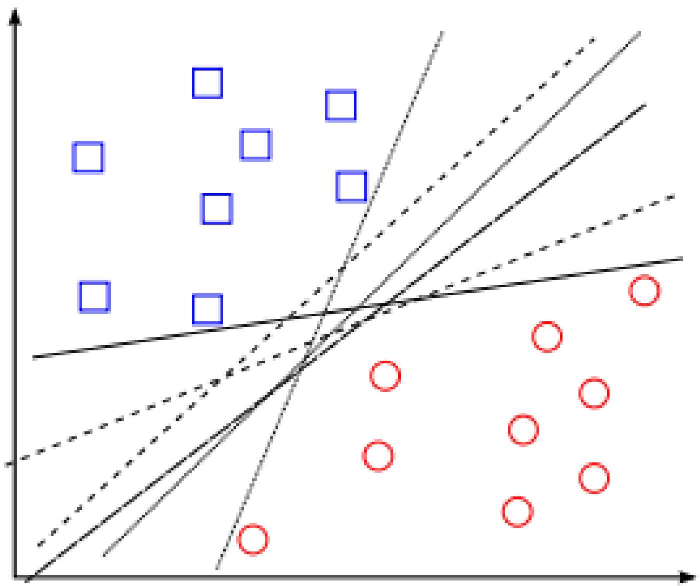
Índice

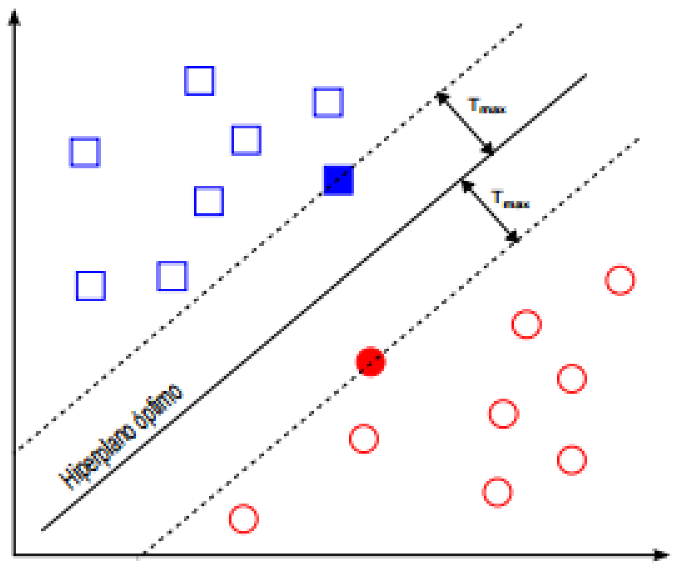
1. Introducción
2. Regresión logística, SVMs y el margen
3. Frontera de decisión en SVM
4. Funciones Kernel y fronteras no lineales
5. SVM para problemas de regresión
6. Comentarios finales

Índice

Máquinas de Vectores Soporte
Support Vector Machines (SVMs)

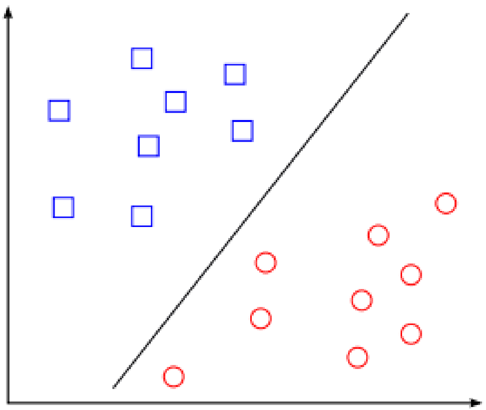
- Uno de los mejores algoritmos de aprendizaje
  - Algunos creen (erróneamente – no free lunch) que no puede haber otro mejor
- **Clasificador de margen óptimo**





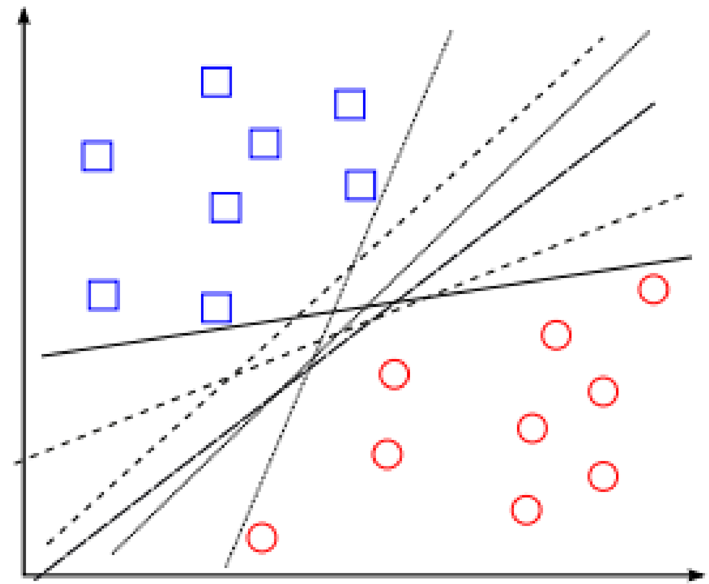
Intuición

- Dado un conjunto separable de m ejemplos de entrenamiento
  - Cada ejemplo con n atributos
  - Cada ejemplo pertenece a una clase {+1, -1}
- Se puede definir un hiperplano que los separe
  - $D(x) = b + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta^T x + b = < \theta, x > + b$
    - θ_i y b son coeficientes reales
  - Regresión logística: $\mathrm{b} = \theta_0$



Intuición

- El **hiperplano de separación** cumple las siguientes **restricciones** para todos los ejemplos de entrenamiento x_i
  - $\theta^T x^{(i)} \geq 0$ si $y_i = +1$
  - $\theta^T x^{(i)} \leq 0$ si $y_i = -1$
    - $\theta^T x^{(i)} y_i \geq 0$

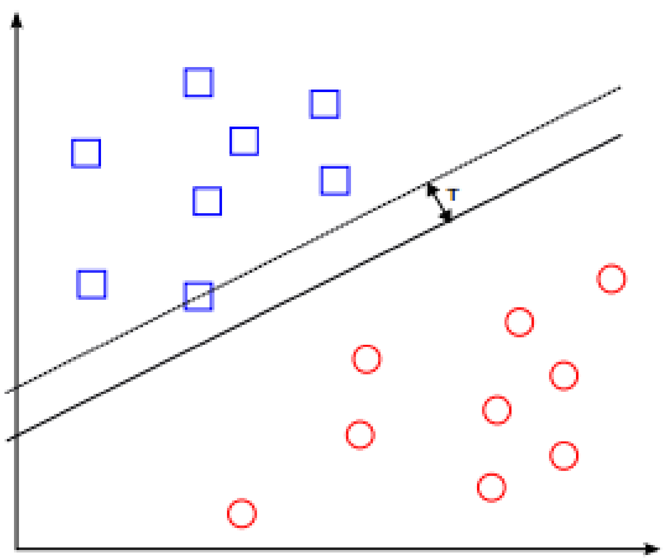


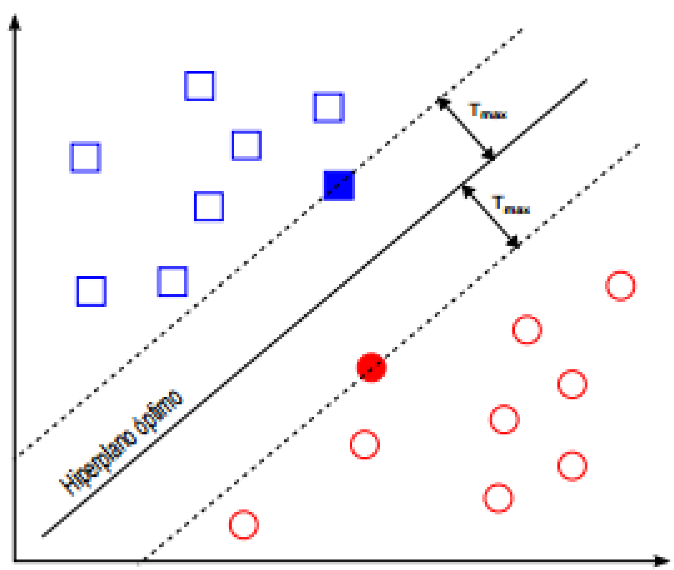
- **El hiperplano no es único**
  - Hay infinitos

¿Existe algún criterio que permita establecer el hiperplano óptimo?

Intuición

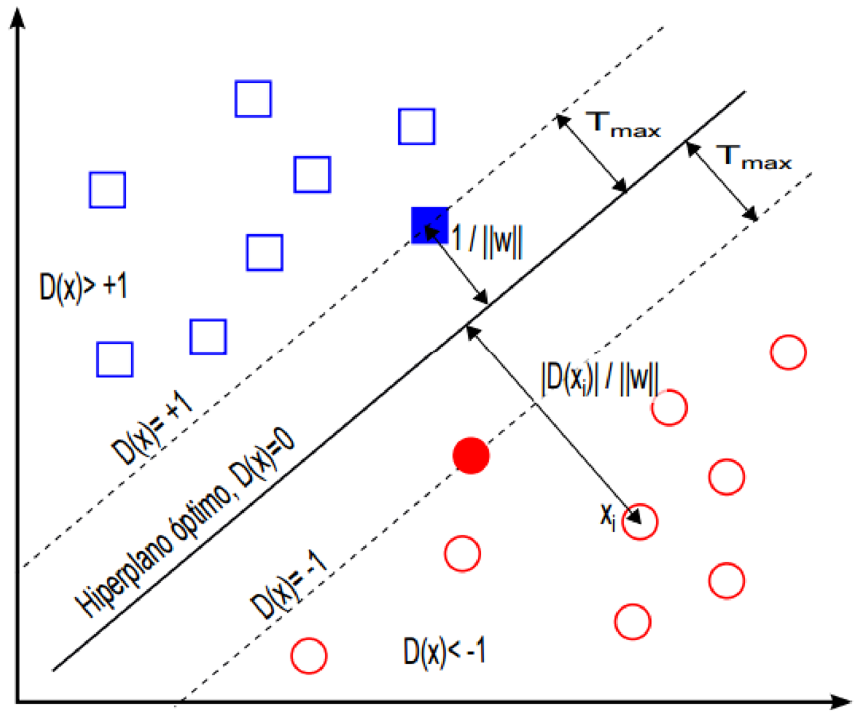
- **Margen**
  - Distancia mínima entre el hiperplano y el(los) ejemplo(s) más cercano(s) de cualquier clase
- Un **hiperplano será óptimo si su margen es de tamaño máximo**
  - Equidista del ejemplo(s) más cercano(s) de cada clase





Intuición

- Los ejemplos que definen el margen son llamados **vectores soporte**
  - Son los únicos utilizados a la hora de construir el hiperplano óptimo

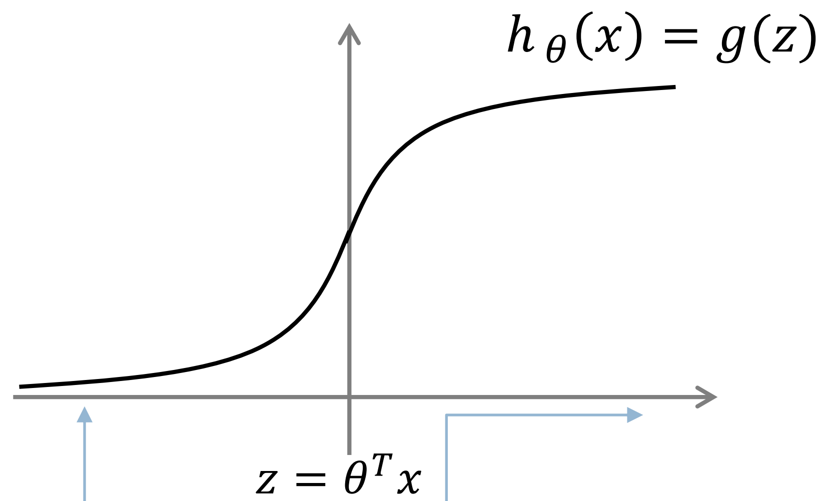


Índice

Regresión logística

- Buscamos una **clasificación lo más segura posible**

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$



Si $y = 1$, queremos $h_\theta(x) \approx 1, \theta^T x \gg 0$

Si $y = 0$, queremos $h_\theta(x) \approx 0, \theta^T x \ll 0$

$y = 0$ en regresión logística es y =-1 en SVM

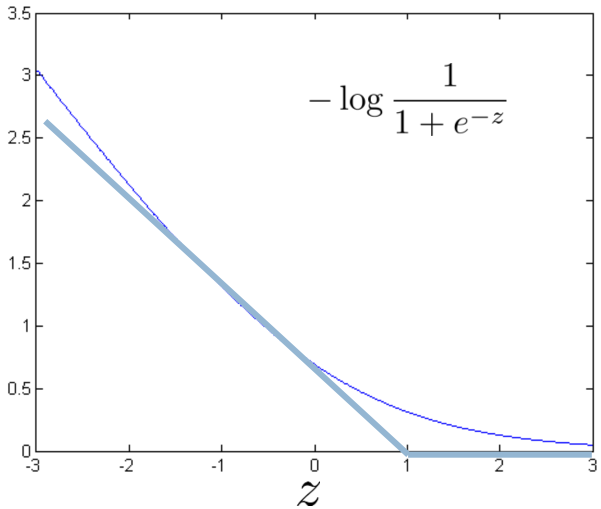
Visión alternativa de la regresión logística

- **Cambiamos la función de coste** para un ejemplo (x, y)

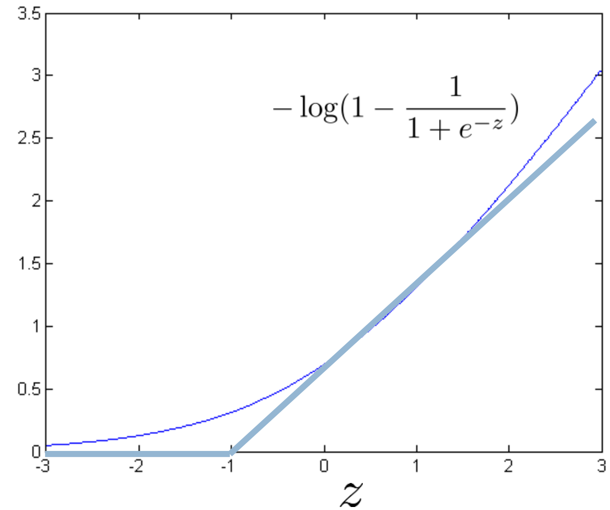
Regr. Logística: $-(y\log\left(\frac{1}{1+e^{-\theta^T x}}\right) + (1-y)\log(1-\frac{1}{1+e^{-\theta^T x}}))$

SVM (Hinge loss): $coste(D(x), y) = \max(0, 1 - y * \theta^T x)$

Si $y = 1$ (queremos $\theta^T x \gg 0$):



Si $y = -1$ (queremos $\theta^T x \ll 0$):



Hinge loss en SVMs

Support Vector Machine

- **Regresión logística**

$$\min_{\theta} \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(-\log\left(h_\theta\left(x^{(i)}\right)\right)\right) + \left(1-y^{(i)}\right)\left(-\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

- **Support vector machine**

$$\min_{\theta} C\sum_{i=1}^{m} cost\left(\theta^T * x^{(i)}, y^{(i)}\right) + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

  - Eliminamos $\frac{1}{m}$ (no cambia el resultado)
  - El parámetro de regularización es C
    - Juega el papel contrario a λ
    - $C = \frac{1}{\lambda}$, es decir, si C es muy grande no regularizamos

Support Vector Machine

$$\min_{\theta} C \sum_{i=1}^{m} cost(\theta^T * x^{(i)}, y^{(i)}) + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

- **Hipótesis**

$$h_\theta(x) = \begin{cases} 1 & si \quad \theta^T x \geq 0 \\ -1 & si \quad \theta^T x < 0 \end{cases}$$

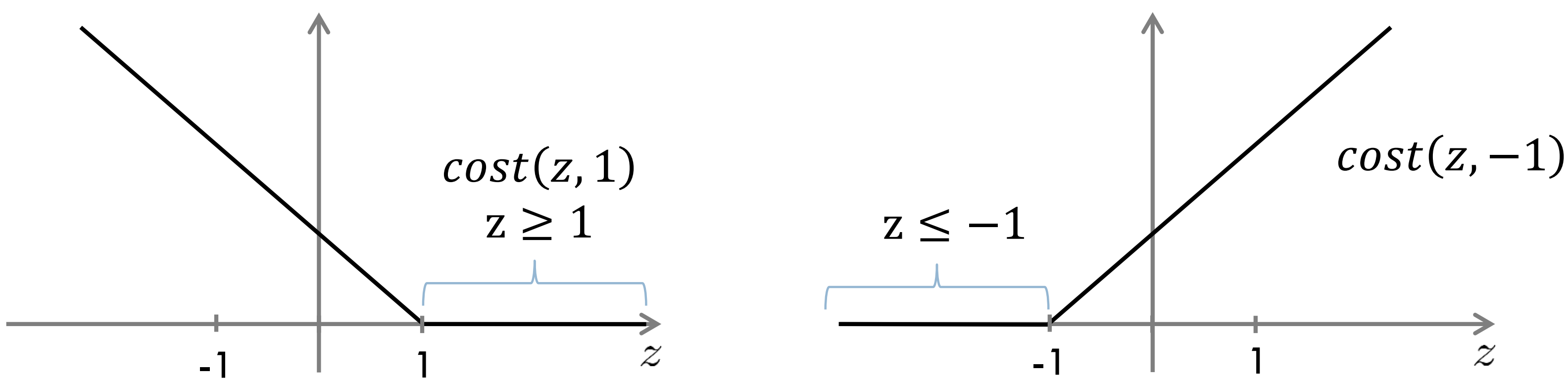
  - Es decir, **las SVMs no devuelven una probabilidad**
    - Es un clasificador discriminativo (pero no probabilístico)

Intuición sobre el margen

- **Función de coste**

$$\min_{\theta} C \sum_{i=1}^{m} cost(\theta^T * x^{(i)}, y^{(i)}) + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

  - **No nos basta con acertar, queremos hacerlo con un margen**



Si $y = 1$, queremos $\theta^T x \geq 1$ (no solo ≥ 0)

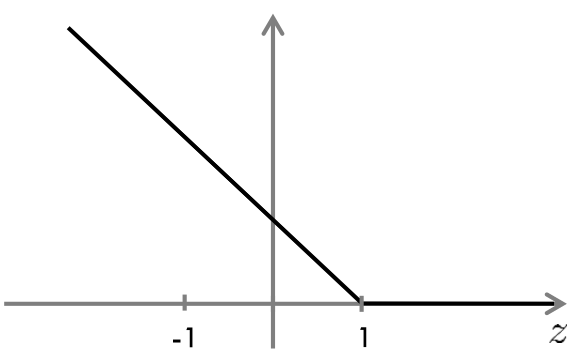
Si $y = -1$, queremos $\theta^T x \leq -1$ (no solo < 0)

Intuición sobre el margen

- **Función de coste**

$$\min_{\theta} C \boxed{\sum_{i=1}^{m} cost(\theta^T * x^{(i)}, y^{(i)})} + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

- Si tomamos un **valor de C muy grande** (100000)
  - Nos centraremos en que **la primera parte sea 0**
  - Es decir, nos centramos en acertar todos los ejemplos
    - CON UN **MARGEN** DE SEGURIDAD



Intuición sobre el margen

- **Función de coste**

$$\min_{\theta} C \boxed{\sum_{i=1}^{m} cost(\theta^T * x^{(i)}, y^{(i)})} + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

= 0

  - Si tomamos un **valor de C muy grande** (100000)
    - Para acertar todos los ejemplos (y que el coste sea 0)

Si $y^{(i)} = 1$:

$$\theta^T x^{(i)} \geq 1$$

Si $y^{(i)} = -1$:

$$\theta^T x^{(i)} \leq -1$$

Podemos rescribirlo

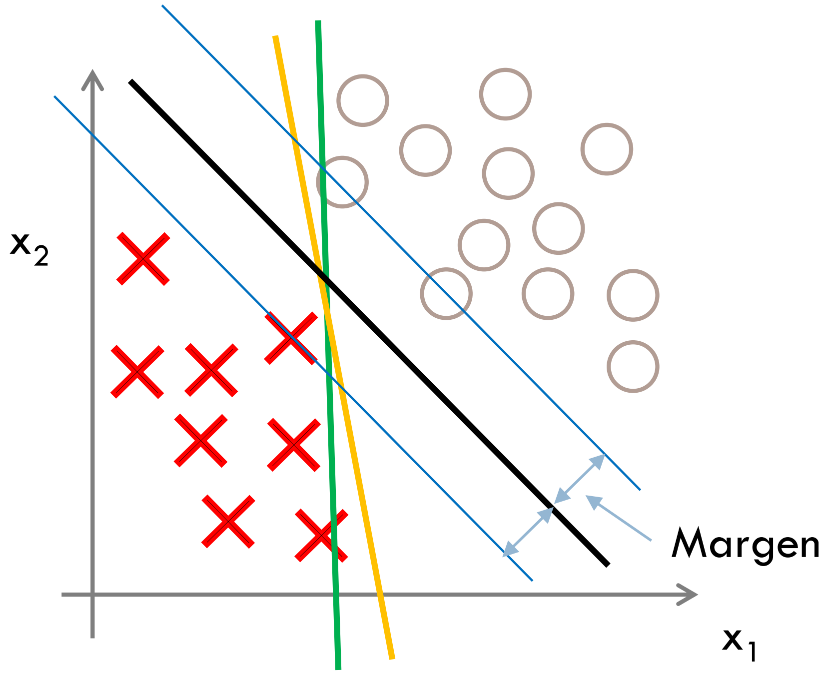
$$\min_{\theta} \cancel{C \cdot 0 +} \frac{1}{2}\sum_{i=1}^{n} \theta_j^2$$

$$sujeto\ a \quad \theta^T x^{(i)} \geq 1 \quad si \quad y^{(i)} = 1$$

$$\theta^T x^{(i)} \leq -1 \quad si \quad y^{(i)} = -1$$

Intuición sobre el margen

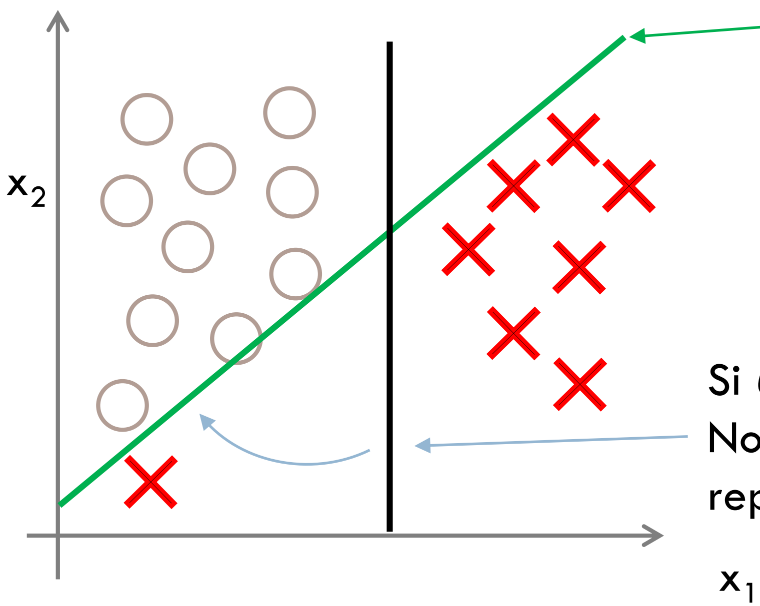
- **Datos linealmente separables**
  - Se pueden separar con una línea recta



"Large margin classifier"

Intuición sobre el margen

- **El margen en la presencia de outliers**



Si C es muy grande

(C juega un papel parecido a $\frac{1}{\lambda}$)

Me ajusto al máximo a mis ejemplos

Trato de acertarlos todos

Si C no es demasiado grande → más interesante

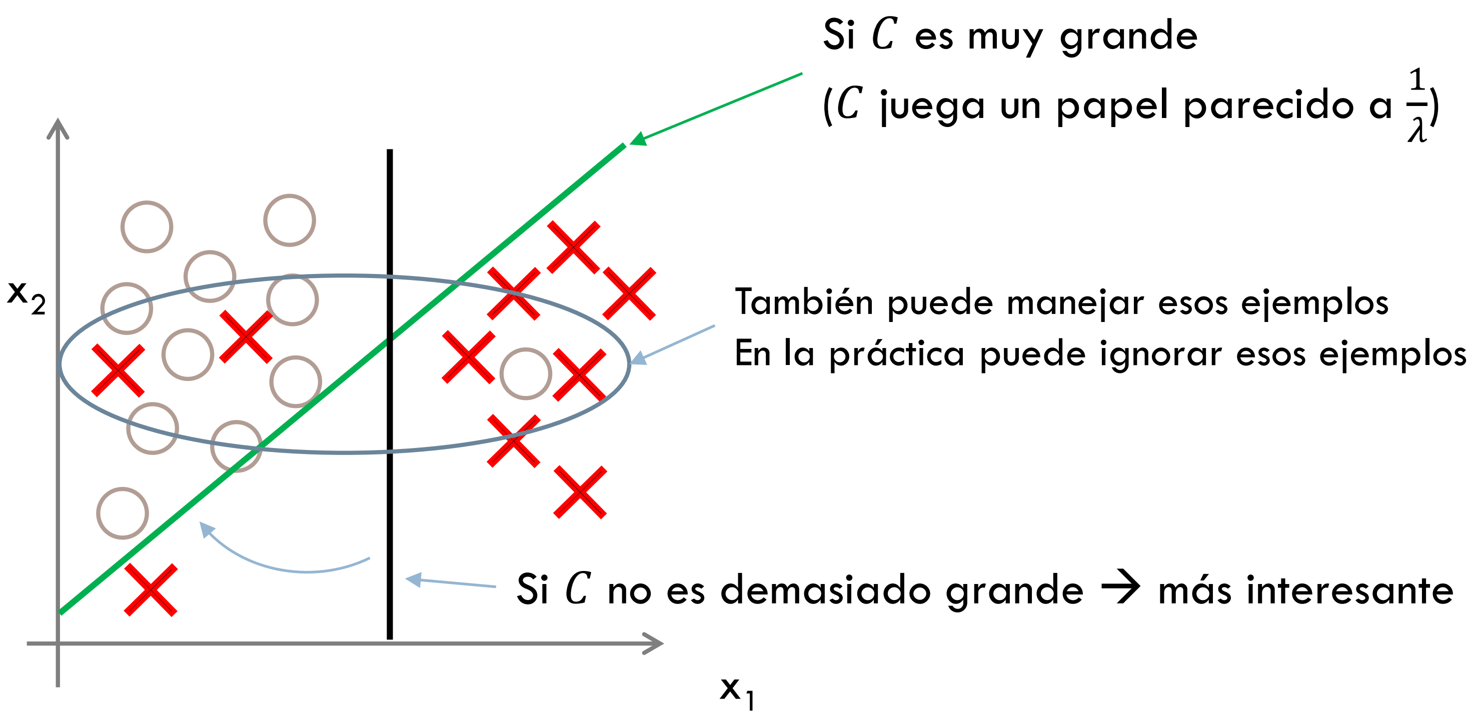
No me importa fallar uno, si el margen obtenido representa mejor el problema

La SVM es capaz de manejar estos casos gracias al margen “suave”

Podemos jugar con C para permitir “fallos” a costa de un mejor margen para el resto de ejemplos

Intuición sobre el margen

- **El margen en la presencia de outliers**



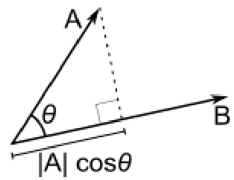
La SVM es capaz de manejar estos casos gracias al margen “suave”

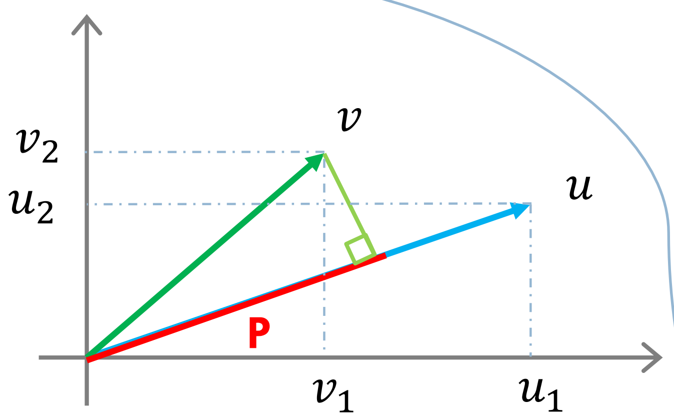
Podemos jugar con C para permitir “fallos” a costa de un mejor margen para el resto de ejemplos

Producto interno (escalar) - repaso

$$u^T v = ||u|| \cdot ||v|| \cdot \cos(\theta)$$

$\boxed{||v|| \cos(\theta)}$ es la longitud de la proyección de v en $u = \mathrm{P}$





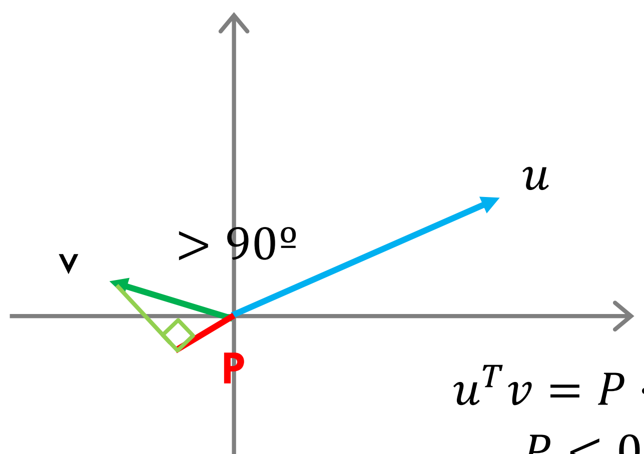
$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$u^T v = ? \qquad [u_1, u_2] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$||u||$ = longitud del vector $u = \sqrt{u_1^2 + u_2^2} \in \mathbb{R}$

P = longitud de la proyección de v en u (con signo)

$$u^T v = P \cdot ||u|| \quad = v^T u$$
$$= u_1 v_1 + u_2 v_2 \qquad\qquad P \in \mathbb{R}$$

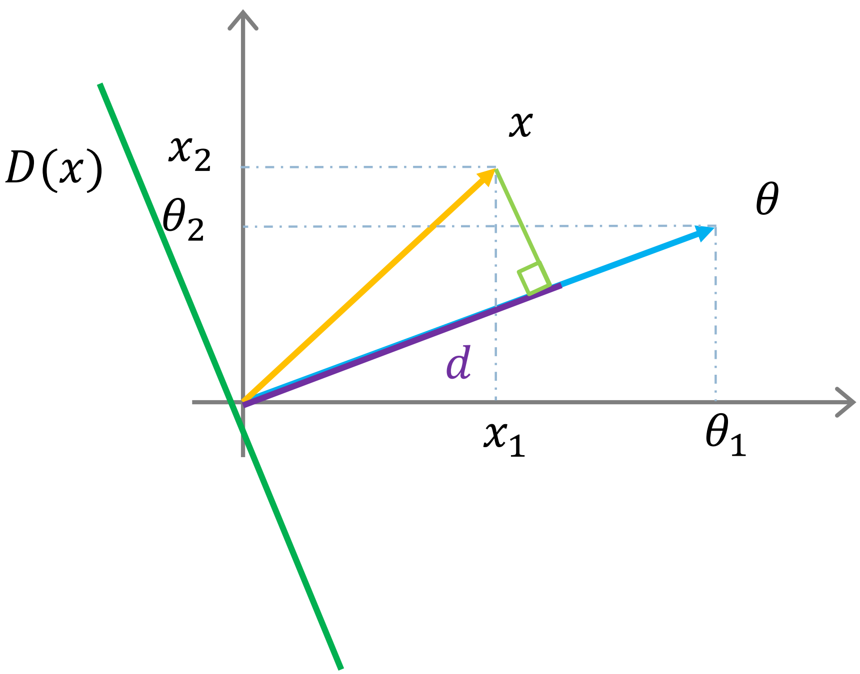


$$u^T v = P \cdot ||u||$$
$$P < 0$$

Índice

Frontera de decisión en SVM

- ¿Cómo calcular el hiperplano óptimo?
  - La distancia entre un hiperplano de separación, $D(x)$, y un ejemplo x es



$$\theta^T x = P \cdot ||\theta|| \qquad P \in \mathbb{R}$$

$$\theta^T x = d \cdot ||\theta|| \qquad d \in \mathbb{R}$$

$$d = \frac{|\theta^T x|}{||\theta||}$$

Frontera de decisión en SVM

- ¿Cómo calcular el hiperplano óptimo?
  - Tenemos que $y_i\theta^T x_i \geq 0$ para todos los ejemplos de entrenamiento
    - Por tanto

$$d = \frac{|\theta^T x|}{||\theta||} = \frac{y_i\theta^T x_i}{||\theta||}$$

    - Y queremos que esta distancia sea mayor que el margen τ

$$\frac{y_i\theta^T x_i}{||\theta||} \geq \tau \rightarrow y_i\theta^T x_i \geq \tau||\theta||$$

Encontrar el hiperplano óptimo es equivalente a encontrar el vector θ que maximice el margen

Frontera de decisión en SVM

- Infinitas soluciones que difieren solamente en la escala de θ
  - Las funciones lineales $\lambda((\theta^T x_i + b)$ con $\lambda \in \mathbb{R}$ representan el mismo plano
- Para evitarlo se limita a la unidad (arbitrariamente) la escala del producto de T y θ

$$\tau\left||\theta|\right| = 1 \qquad \tau = \frac{1}{\left||\theta|\right|}$$

- Aumentar el margen es equivalente a reducir la norma de θ
  - Por tanto: un hiperplano de separación óptimo es el que posee un margen máximo (valor mínimo de $||\theta||$) sujeto a 1 (por la restricción)

$$y_i * (\theta^T x_i + b) \geq 1, i \in \{1, \dots, m\}$$

  - El concepto de margen máximo está relacionado con la capacidad de generalización
  - Los ejemplos que cumplen $y_i * (\theta^T x_i + b) = 1$ son los vectores soporte

Frontera de decisión en SVM

Objetivo SVMs

Asumiendo la simplificación de $\theta_0 = 0$ y $n = 2$ (conjunto bidimensional)
(Es extensible al modelo completo)

$$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n} \theta_j^2 \Rightarrow = \frac{1}{2}(\theta_1^2 + \theta_2^2) = \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2 = \frac{1}{2}||\theta||^2$$

$$sujeto\ a \quad \theta^T x^{(i)} \geq 1 \quad si \quad y^{(i)} = 1$$
$$\theta^T x^{(i)} \leq -1 \quad si \quad y^{(i)} = -1$$

Por tanto, la SVM se centra en minimizar la norma de θ

¿Qué significa esto?

Frontera de decisión en SVM

- **Objetivo SVMs**

Asumiendo la simplificación de $\theta_0 = 0$ y $n = 2$ (conjunto bidimensional)
Es extensible al modelo completo

$$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n} \theta_j^2 \Rightarrow = \frac{1}{2}(\theta_1^2 + \theta_2^2) = \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2 = \frac{1}{2}||\theta||^2$$

$$sujeto\ a \begin{cases} \theta^T x^{(i)} \geq 1 \quad si \quad y^{(i)} = 1 \\ \theta^T x^{(i)} \leq -1 \quad si \quad y^{(i)} = -1 \end{cases}$$

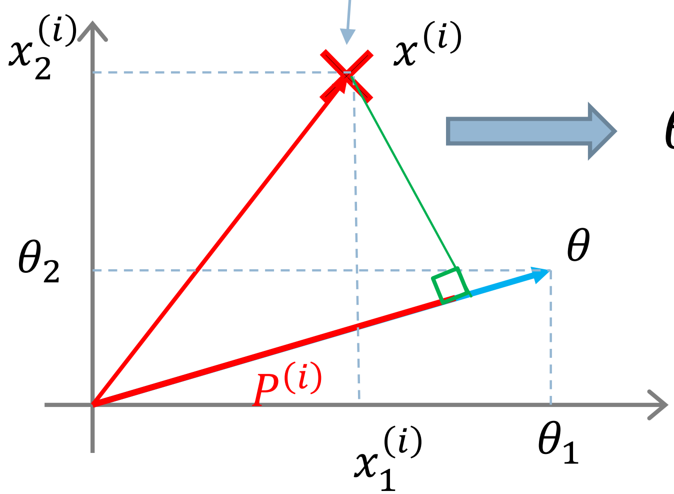
Por tanto, la SVM se centra en minimizar la norma de θ

Veamos cómo calcular esta parte

$$\theta^T x^{(i)} = ?$$

$$u^T v$$

Ejemplo positivo



$$\theta^T x^{(i)} = \boxed{P^{(i)} \cdot ||\theta||}$$

$$= \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$$

Podemos rescribir las restricciones

Frontera de decisión en SVM

- **Objetivo SVMs**

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{n} \theta_j^2 = \frac{1}{2} \left|\left|\theta\right|\right|^2$$

$$sujeto\ a\ \ P^{(i)} \cdot \left|\left|\theta\right|\right| \geq 1 \qquad si \quad y^{(i)} = 1$$

$$P^{(i)} \cdot \left|\left|\theta\right|\right| \leq -1 \quad si \quad y^{(i)} = -1$$

Donde $P^{(i)}$ es la proyección de $x^{(i)}$ en el vector θ

Simplificación: $\theta_0 = 0$

Frontera de decisión en SVM

- **Objetivo SVMs**

El vector θ es perpendicular a la frontera de decisión

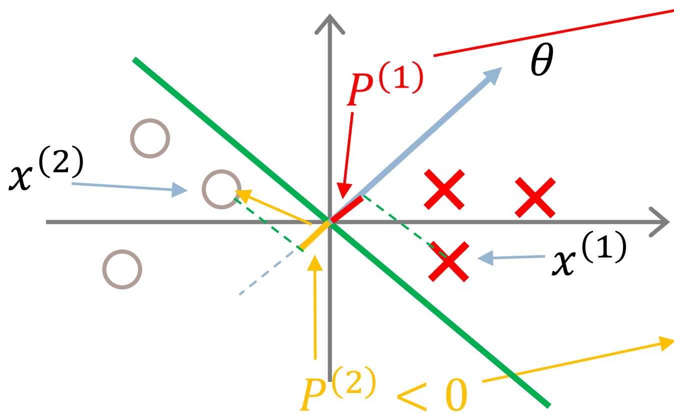
$$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n} \theta_j^2 = \frac{1}{2}\left|\left|\theta\right|\right|^2$$

$$\left.\begin{array}{l} sujeto\ a\ \ P^{(i)} \cdot \left|\left|\theta\right|\right| \geq 1 \quad si \ \ y^{(i)} = 1 \\ \qquad\qquad P^{(i)} \cdot \left|\left|\theta\right|\right| \leq -1 \quad si \ \ y^{(i)} = -1 \end{array}\right\} \ C \text{ muy grande}$$

Donde $P^{(i)}$ es la proyección de $x^{(i)}$ en el vector θ

Simplificación: $\theta_0 = 0$

Opción 1



$P^{(1)} \cdot \left|\left|\theta\right|\right| \geq 1$

Como $P^{(1)}$ es pequeño $\left|\left|\theta\right|\right|$debe ser grande

$P^{(2)} \cdot \left|\left|\theta\right|\right| \leq -1$

Como $P^{(2)}$ es pequeño $\left|\left|\theta\right|\right|$debe ser grande

Pero queremos minimizar $\left|\left|\theta\right|\right|$

Frontera de decisión en SVM

- **Objetivo SVMs**

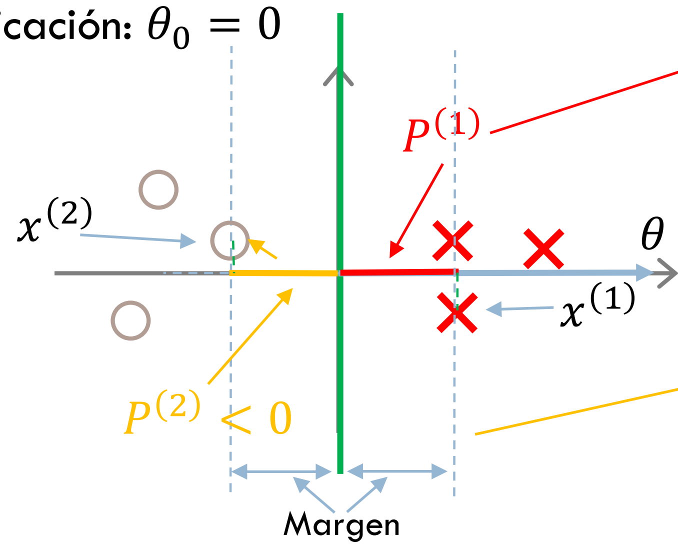
$$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n}\theta_j^2 = \frac{1}{2}\left|\left|\theta\right|\right|^2$$

$$\left.\begin{aligned} &sujeto\ a\ \ P^{(i)}\cdot\left|\left|\theta\right|\right| \geq 1 \quad si \ \ y^{(i)} = 1 \\ &P^{(i)}\cdot\left|\left|\theta\right|\right| \leq -1 \quad si \ \ y^{(i)} = -1 \end{aligned}\right\} C \text{ muy grande}$$

Donde $P^{(i)}$ es la proyección de $x^{(i)}$ en el vector θ

Simplificación: $\theta_0 = 0$

Opción 2



$P^{(1)}\cdot\left|\left|\theta\right|\right| \geq 1$

Como $P^{(1)}$ es más grande que antes $\left|\left|\theta\right|\right|$Puede ser más pequeño

$P^{(2)}\cdot\left|\left|\theta\right|\right| \leq -1$

Como $P^{(2)}$ más grande que antes $\left|\left|\theta\right|\right|$puede ser más pequeño

Logramos minimizar $\left|\left|\boldsymbol{\theta}\right|\right|$

Frontera de decisión en SVM

- **Objetivo SVMs**

$$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n}\theta_j^2 = \frac{1}{2}\left|\left|\theta\right|\right|^2$$

$$\left.\begin{array}{ll} sujeto\ a\ \ P^{(i)}\cdot\left|\left|\theta\right|\right| \geq 1 & si\ \ y^{(i)} = 1 \\ P^{(i)}\cdot\left|\left|\theta\right|\right| \leq -1 & si\ \ y^{(i)} = -1 \end{array}\right\} C \text{ muy grande}$$

Donde $P^{(i)}$ es la proyección de $x^{(i)}$ en el vector θ

Simplificación: $\theta_0 = 0$

Por tanto

- Buscamos que las proyecciones de los ejemplos sobre θ sean lo más grandes posibles
- Esto es lo que provoca que la SVM busque márgenes grandes
- Haciendo los márgenes grandes, la SVM puede obtener una norma de θ menor

Frontera de decisión en SVM

- **Objetivo SVMs**

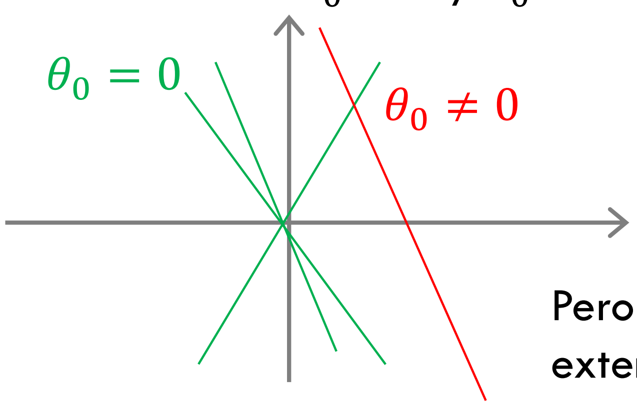
$$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n}\theta_j^2 = \frac{1}{2}\left|\left|\theta\right|\right|$$

$$\left.\begin{array}{ll} sujeto\ a\ \ P^{(i)}\cdot\left|\left|\theta\right|\right| \geq 1 & si\ \ y^{(i)} = 1 \\ P^{(i)}\cdot\left|\left|\theta\right|\right| \leq -1 & si\ \ y^{(i)} = -1 \end{array}\right\} \ C \text{ muy grande}$$

Donde $P^{(i)}$ es la proyección de $x^{(i)}$ en el vector θ

Simplificación: $\theta_0 = 0$

Diferencia entre $\theta_0 = 0$ y $\theta_0 \neq 0$



Pero lo estudiado anteriormente es extensible al caso donde $\theta_0 \neq 0$

Frontera de decisión en SVM

Objetivo SVMs

$$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n} \theta_j^2 = \frac{1}{2}\left|\left|\theta\right|\right|^2 \quad \Rightarrow \quad \min_{\theta} \frac{1}{2}\sum_{j=1}^{n} \theta_j^2 = \frac{1}{2}\left|\left|\theta\right|\right|^2$$

$$sujeto\ a\ \ P^{(i)} \cdot \left|\left|\theta\right|\right| \geq 1 \quad si \quad y^{(i)} = 1$$

$$P^{(i)} \cdot \left|\left|\theta\right|\right| \leq -1 \quad si \quad y^{(i)} = -1$$

$$sujeto\ a \quad y_i * (\theta^T x_i + b) \geq 1$$

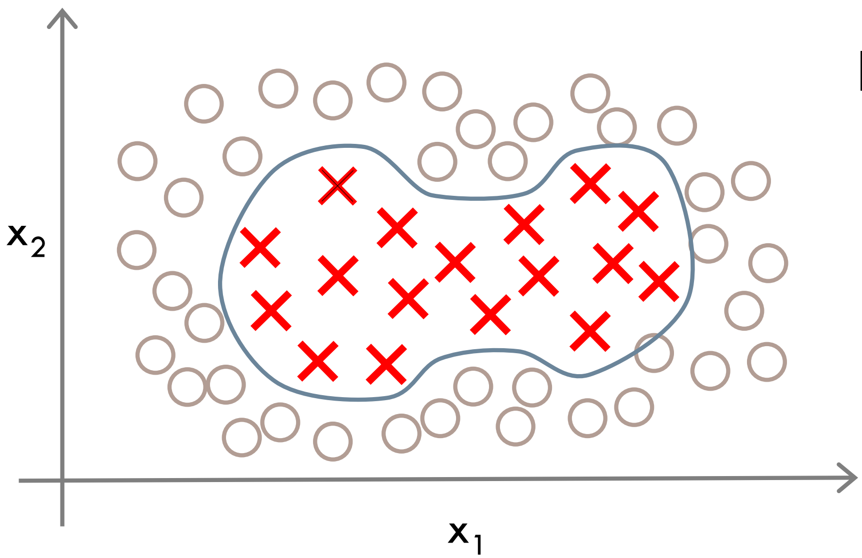
Donde $P^{(i)}$ es la proyección de $x^{(i)}$ en el vector θ

- Es un problema de minimización cuadrática con restricciones
  - La función de coste de las SVMs, al igual que la de la regresión logística, es convexa
    - Tiene una única solución global – El hiperplano con margen máximo
  - Las restricciones también son convexas
- No entraremos en el algoritmo que permite encontrar dicho mínimo
  - Sequential Minimization Optimization (SMO)
- Utilizaremos el software incluido en scikit-learn (otros interesantes son Liblinear y libSVM)

Índice

Fronteras de decisión no lineales

- **Características polinomiales**



Predecimos $y = 1$ si

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2$$
$$+ \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots \geq 0$$

$$h_\theta(x) = \begin{cases} 1 & si \quad \theta_0 x_0 + \theta_1 x_1 + \cdots \geq 0 \\ 0 & en\ otro\ caso \end{cases}$$

Podemos rescribir el modelo

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$$
$$f_1 = x_1, f_2 = x_2, f_3 = x_1 x_2, f_4 = x_1^2, f_5 = x_2^2, \ldots$$

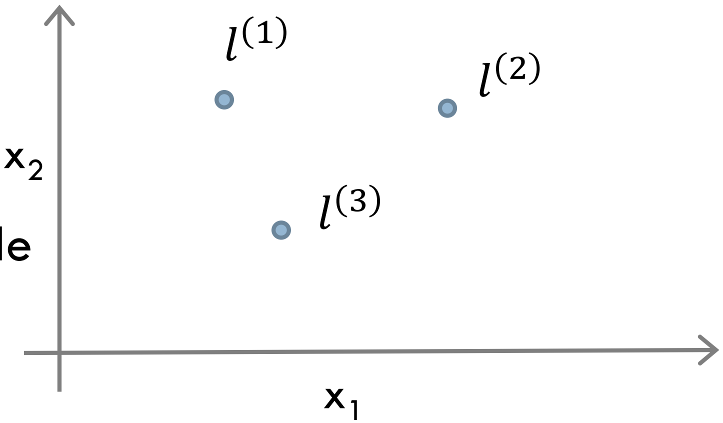
¿Hay una forma diferente/mejor de elegir las características $f_1, f_2, f_3, \ldots$?

Funciones KERNEL: Además de facilitarnos la tarea, podemos introducirlas de forma eficiente en las SVMs

Funciones Kernel

- **Cálculo de características**

Dado x, calculamos nuevas características dependiendo de su proximidad a los puntos de referencia $l^{(1)}, l^{(2)}, l^{(3)}$



Dado x

$$f_1 = similitud\left(x, l^{(1)}\right) = \exp\left(-\frac{\overbrace{\left\|x - l^{(1)}\right\|^2}^{\text{Distancia euclidea entre } x \text{ y } l^{(1)}}}{2\sigma^2}\right)$$

$$f_2 = similitud\left(x, l^{(2)}\right) = \exp\left(-\frac{\left\|x - l^{(2)}\right\|^2}{2\sigma^2}\right)$$

$$f_3 = similitud\left(x, l^{(3)}\right) = \underbrace{\exp\left(-\frac{\left\|x - l^{(3)}\right\|^2}{2\sigma^2}\right)}_{\text{Kernel Gaussiano}}$$

Kernel $k\left(x, l^{(i)}\right)$

Funciones Kernel y similitud

- El **kernel mide la similitud entre un ejemplo y cada marcador**
  - Mapeamos el ejemplo a un espacio de características mayor

$$f_1 = similitud\left(x, l^{(1)}\right) = \exp\left(-\frac{\left\|x - l^{(1)}\right\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^{n}\left(x_j - l_j^{(1)}\right)^2}{2\sigma^2}\right)$$

Si $x \approx l^{(1)}$:

$$f_1 \approx \exp\left(-\frac{0^2}{2\sigma^2}\right) \approx 1$$

Si x está lejos de $l^{(1)}$:

$$f_1 = \exp\left(-\frac{(número\ grande)^2}{2\sigma^2}\right) \approx 0$$

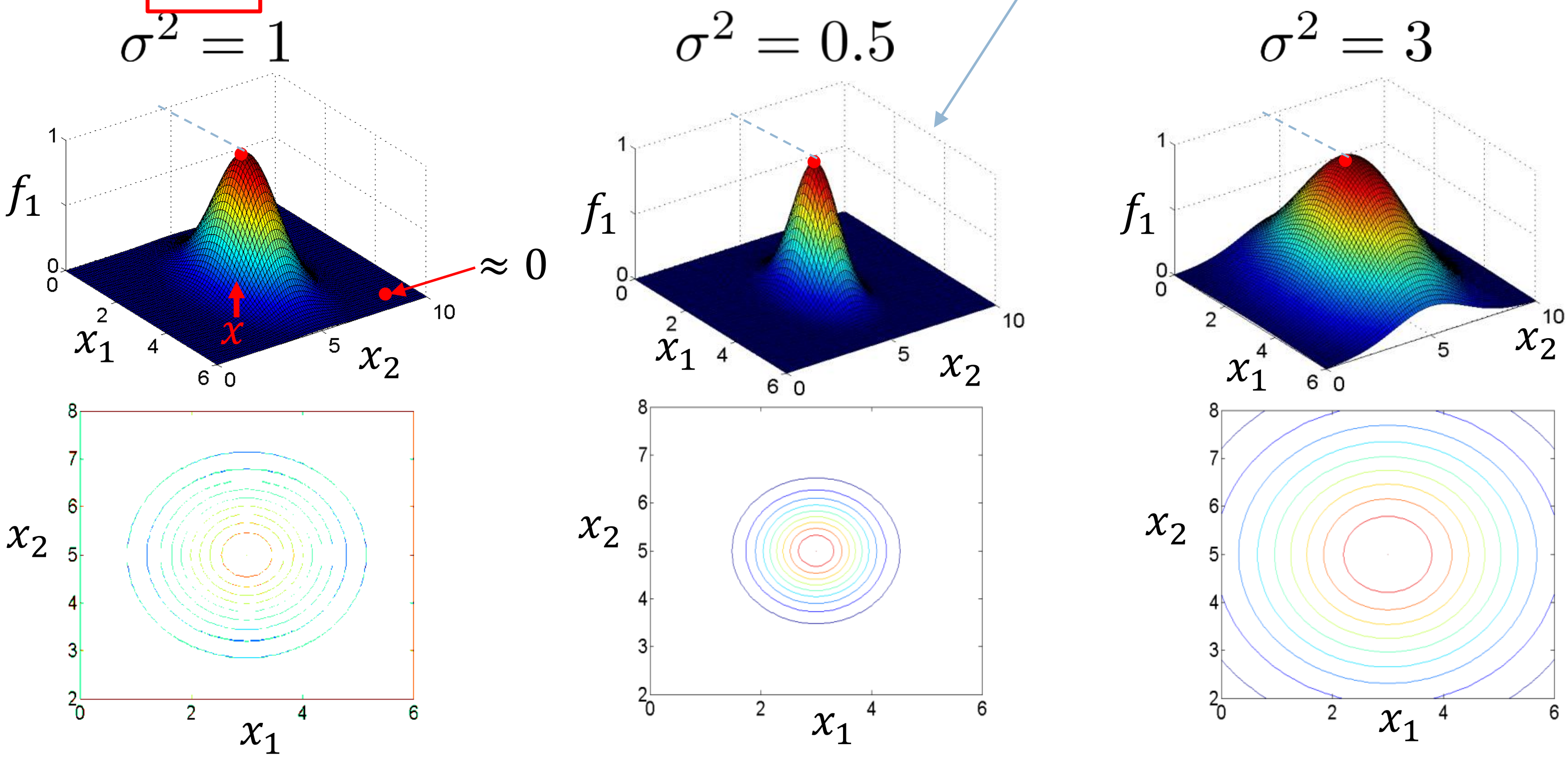
$$l^{(1)} \rightarrow f_1$$
$$l^{(2)} \rightarrow f_2$$
$$l^{(3)} \rightarrow f_3$$

Nueva representación x

Ejemplo

$$l^{(1)} = \boxed{\begin{bmatrix}3\\5\end{bmatrix}}, \quad f_1 = \exp\left(-\frac{\|x-l^{(1)}\|^2}{2\boxed{\sigma^2}}\right)$$

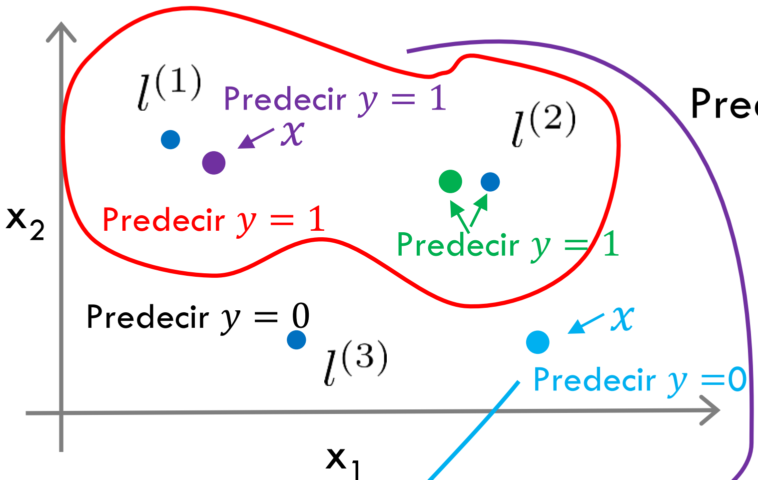
Menor sigma, la similitud tiende a 0 más rápidamente



Mayor sigma, la similitud tiende a 0 más lentamente

Fronteras de decisión no lineales

- Nuevo modelo con **nuevas características**



Predecir "1" cuando

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$$

Similitud de x con cada marcador

$$\theta_0 = -0.5, \theta_1 = 1, \theta_2 = 1, \theta_3 = 0$$

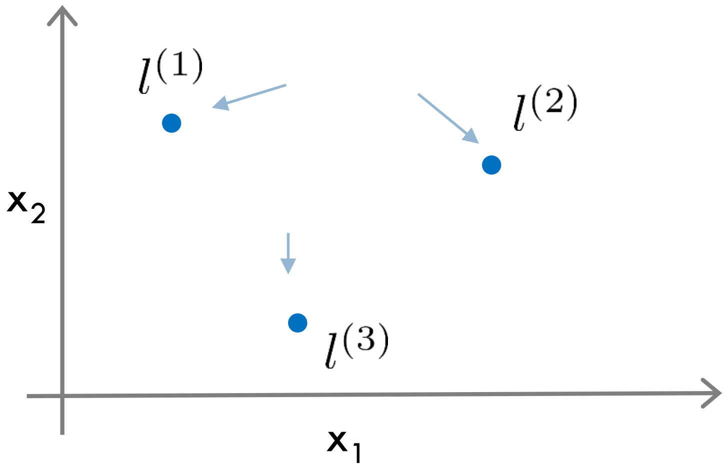
$$f_1 \approx 1, f_2 \approx 0, f_3 \approx 0$$

$$\theta_0 + \theta_1 \cdot 1 + \theta_2 \cdot 0 + \theta_3 \cdot 0 \approx -0.5 + 1 \approx 0.5 \geq 0$$

$$f_1 \approx 0, f_2 \approx 0, f_3 \approx 0$$

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \approx -0.5 < 0$$

Cómo elegir los marcadores



$x^{(1)}$

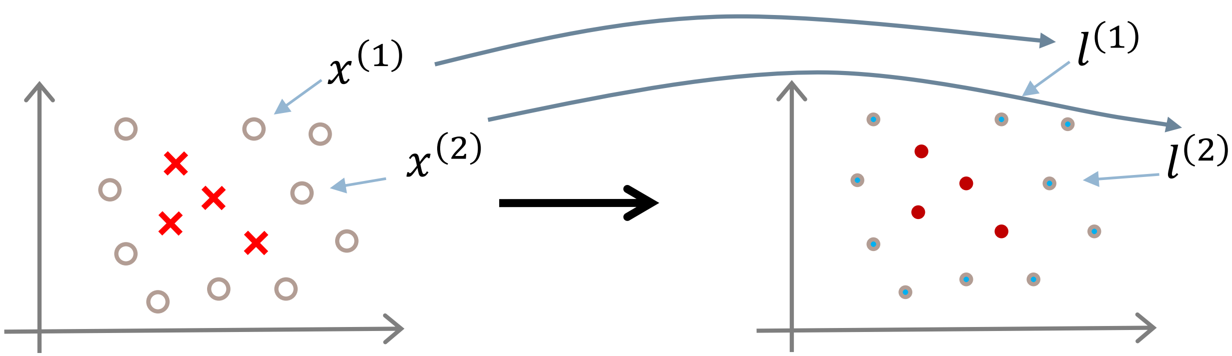
Dado x:

$$f_i = similitud\left(x, l^{(i)}\right)$$
$$= \exp\left(-\frac{\left\|x - l^{(i)}\right\|^2}{2\sigma^2}\right)$$

Predecir $y = 1$ si $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

¿De dónde sacamos $l^{(1)}, l^{(2)}, l^{(3)}, \ldots$?

Utilizamos cada ejemplo como marcador



$l^{(1)}$
$l^{(2)}$
$\vdots$
$l^{(1)}$
$l^{(m)}$

SVM con Kernels

Dados $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$,
elegir $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \ldots, l^{(m)} = x^{(m)}$.

Dado un ejemplo x

$x^{(1)}$

$$f_1 = similitud\left(x, l^{(1)}\right)$$
$$f_2 = similitud\left(x, l^{(2)}\right)$$
$$\ldots$$

$$f = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} \qquad f_0 = 1$$

Para el ejemplo de entrenamiento $(x^{(i)}, y^{(i)})$:

$$x^{(i)} \rightarrow \begin{bmatrix} f_1^{(i)} \\ f_2^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix} \begin{matrix} = sim\left(x^{(i)}, l^{(1)}\right) \\ = sim\left(x^{(i)}, l^{(2)}\right) \\ \leftarrow f_i^{(i)} = sim\left(x^{(i)}, l^{(i)}\right) = \exp\left(-\frac{0}{2\sigma^2}\right) = 1 \\ = sim\left(x^{(i)}, l^{(3)}\right) \end{matrix}$$

$$x^{(i)} \in \mathbb{R}^{n+1} \qquad f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ f_2^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix} \qquad f_0^{(i)} = 1$$

Todos los ejemplos tendrán 1 característica a 1 (a parte del bias)

Es decir, aplicamos el kernel para cada ejemplo con todos los demás y por tanto mapeamos el ejemplo a un vector m-dimensional

SVM con Kernels

Hipótesis: Dado x, calcular las características $f \in \mathbb{R}^{m+1}$

Predecir "y=1" si $\theta^T f \geq 0 \;\rightarrow \theta_0 f_0 + \theta_1 f_1 + \cdots + \theta_m f_m \quad \theta \in \mathbb{R}^{m+1}$

Entrenamiento:

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} cost_1(\underbrace{\theta^T f^{(i)}}_{\text{En vez de } \theta^T x^{(i)} \rightarrow \theta^T f^{(i)}}) + (1 - y^{(i)}) cost_0(\theta^T f^{(i)}) + \boxed{\frac{1}{2}\sum_{j=1}^{n} \theta_j^2}$$

En este caso $n = m$

θ_0 no se regulariza

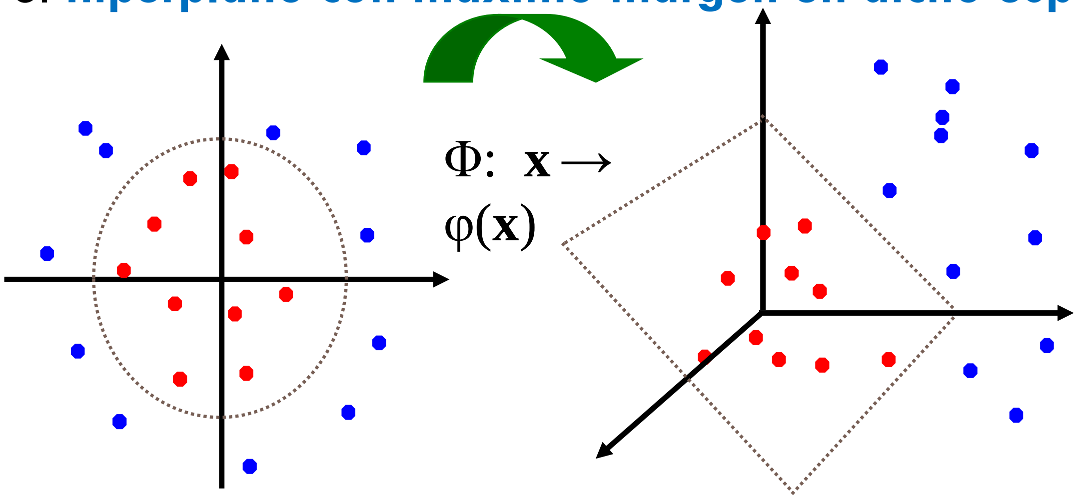
Modificación para mejorar la eficiencia:

- Rescribimos $\sum_{j=1}^{m} \theta_j^2 = \theta^T \theta \;\leftarrow \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix}$ ignorando θ_0
- Se sustituye por $\theta^T M \theta$ donde M es una matriz que depende del kernel y modifica las distancias
- Hace el aprendizaje de la SVM más eficiente (no entraremos en detalles)

No vamos a meternos en cómo minimizar esta función, utilizaremos paquetes que lo hacen

Intuición Kernel

- La idea intuitiva bajo el uso de los kernels es que **mapeamos nuestros ejemplos a un espacio de características mucho mayor**
- Y buscamos el **hiperplano con máximo margen en dicho espacio**



- **$\mathbf{\Phi}$ es la función de transformación**
  - Convierte un ejemplo x en un punto del espacio de características

$$\Phi(x) = [\phi_1(x), \ldots, \phi_m(x)]$$

  - Cada función ϕ_i es una función no lineal
    - m es el número de variables del nuevo espacio creado

Intuición Kernel

- Gracias al uso de las funciones kernel, **evitamos que el coste computacional aumente**
  - Ya que el resultado de la función kernel entre dos ejemplos es el mismo que el de trasladar cada ejemplo por separado a un espacio de características mucho mayor y luego calcular su producto escalar
  - Es decir, **en realidad no necesitamos calcular el nuevo conjunto de características para cada ejemplo**

$$K(\boldsymbol{x}, \boldsymbol{x}') = < \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') >= (\phi_1(\boldsymbol{x})\phi_1(\boldsymbol{x}') + \ldots + \phi_m(\boldsymbol{x})\phi_m(\boldsymbol{x}'))$$

- El kernel entre dos ejemplos es igual al producto escalar entre el mapeo de dichos ejemplos a otro espacio de características
- Cada kernel se corresponde con un mapeo diferente
- El kernel Gaussiano corresponde a un mapeo polinomial a infinitas características

SVM con kernel

- Una vez realizada la transformación se aprende el hiperplano óptimo en el espacio de características

$$D(x) = \theta_1 \phi_1(x) + \cdots + \theta_m \phi_m(x) = < \theta, \Phi(x) >$$

La frontera de decisión lineal aprendida en el espacio de características se convierte en una frontera de decisión no lineal en el espacio original

Índice

Support Vector Regression (SVR)

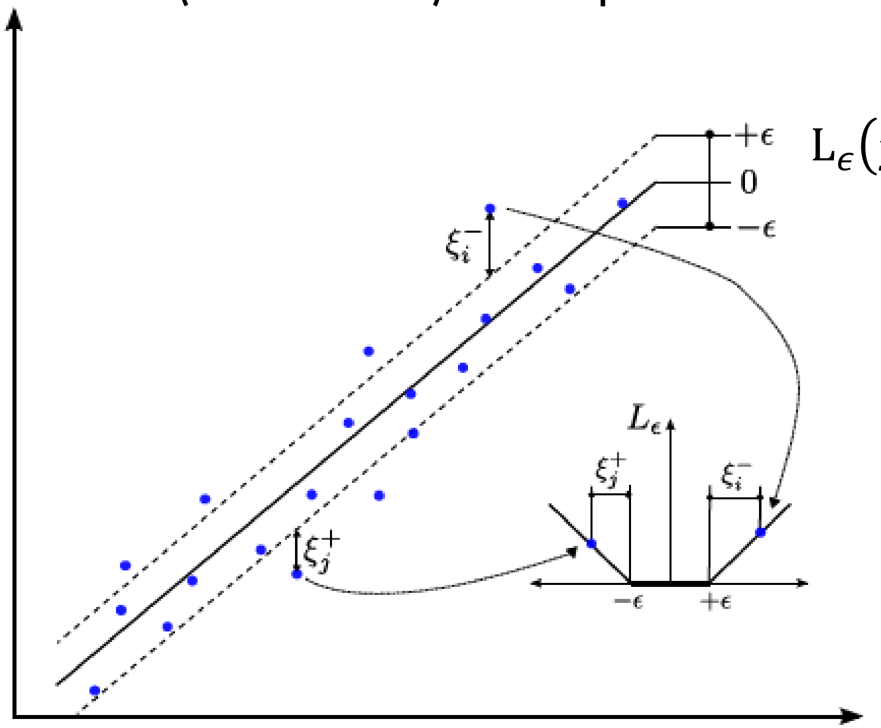
- Dado un conjunto de m ejemplos de entrenamiento
  - $S = ((x_i, y_i), i \in \{1, \dots, m\}, x_i \in \mathbb{R}^n, y_i \in \mathbb{R}$
  - Se asume que todos los y_i se puede ajustar (o casi) mediante una función lineal

- Objetivo: encontrar el vector de parámetros θ que permita definir la función lineal

$$f(x) = \theta_1 x_1 + \dots + \theta_n x_n + b = \theta^T x + b = < \theta, x > + b$$

Support Vector Regression (SVR)

- Para permitir cierto ruido en los ejemplos de entrenamiento se puede relajar la condición de error entre el valor predicho y el real
  - Función de pérdida ϵ-sensible, L_ϵ: función lineal con una zona insensible (anchura 2ϵ) en la que el error es 0



$$L_\epsilon\big(y, f(x)\big) = \begin{cases} 0 & si\ |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon & en\ otro\ caso \end{cases}$$

Support Vector Regression (SVR)

- Es muy difícil que los ejemplos se ajusten al modelo con error 0
  - Se recurre al margen
    - Con dos variables de holgura ξ^+ y ξ^- que cuantifican el error
      - $\xi_i^+ > 0$ cuando $\mathrm{f}(x_i) - y_i > \epsilon$ y $\xi_i^+ = 0$ en otro caso
      - $\xi_i^- > 0$ cuando $\mathrm{y_i} - \mathrm{f}(x_i) > \epsilon$ y $\xi_i^- = 0$ en otro caso
      - $\xi_i^+ * \xi_i^- = 0$
    - La suma de todas las variables de holgura permite cuantificar el coste asociado a los ejemplos con error de predicción no nulo
      - En clasificación tenemos solo una variable de holgura por cada ejemplo: ξ_i

Support Vector Regression (SVR)

- Problema de optimización
  - Igual al de clasificación pero con dos variables de holgura

$$\min_{\theta} \frac{1}{2}||\theta||^2 + C\sum_{i=1}^{m}(\xi_i^+ + \xi_i^-)$$
$$sujeto\ a \quad (\theta^T x_i + b) - y_i - \epsilon - \xi_i^+ \leq 0$$
$$\mathrm{y_i} - (\theta^T x_i + b) - \epsilon - \xi_i^- \leq 0$$
$$\xi_i^+, \xi_i^- \geq 0, i \in \{1, \ldots, m\}$$

- Si los ejemplos no pueden ajustarse por una función lineal
  - Uso de las funciones kernel

Índice

1. Introducción
2. Regresión logística, SVMs y el margen
3. Frontera de decisión en SVM
    1. Frontera dura (hard margin)
    2. Frontera suave (soft margin)
4. Funciones Kernel y fronteras no lineales
5. SVM para problemas de regresión
6. Comentarios finales

Comentarios finales

Utilizaremos **cualquier software de SVMs** para encontrar los parámetros θ (ej., scikit-learn, liblinear, libsvm, …)

Es necesario **especificar**:

- **Elección del parámetro C.**
- **Elección del kernel** (función de similitud):

Ej. Sin kernel ("kernel **lineal**")

Predecir "y = 1" si $\theta^T x \geq 0$

$$\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \geq 0$$

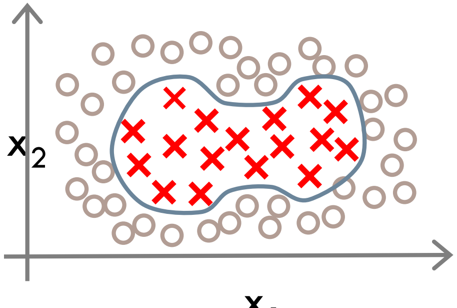
n grande, m pequeño $x \in \mathbb{R}^{n+1}$

Kernel **Gaussiano**:

$$f_i = \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right), \text{ donde } l^{(i)} = x^{(i)}$$

Necesario elegir σ^2

n pequeño y/o m grande



Comentarios finales

- **Sobre las funciones kernel**
  - Es necesario el **escalado antes de usar el kernel Gaussiano**
  - Sino la **distancia Euclídea viene influenciada por la magnitud de las características**

$$||x - l||^2 = \underbrace{(x_1 - l_1)^2}_{m^2} + \underbrace{(x_2 - l_2)^2}_{\text{1-5 habitaciones}} + \cdots + (x_n - l_n)^2$$

Los metros cuadrados tendrán mucha más influencia que el número de habitaciones si no están normalizadas

  - Hay muchas **funciones kernel**
    - Generalmente con la **Gaussiana** es suficiente
    - El **kernel polinomial** también suele usarse
    - Otras: String kernel, chi-square kernel, histogram intersection kernel,…

Comentarios finales

- **Clasificación multiclase**
  - Las SVMs no soportan múltiples clases de manera nativa
  - Veremos más adelante cómo utilizarlas
    - **One-vs-One**
    - **One-vs-All**