Máquinas de vectores soporte (Support Vector Machine)

Dataset: Conjunto de datos clasificados en 2 clases

$$\mathcal{C} = \{ (\vec{x}_n, t_n) \mid n = 1, \_\_, N \}$$

FEATURES: $\vec{x}_n$ — TARGET: t_n — NÚMERO DE INSTANCIAS (FILAS DE LA B.D.): N

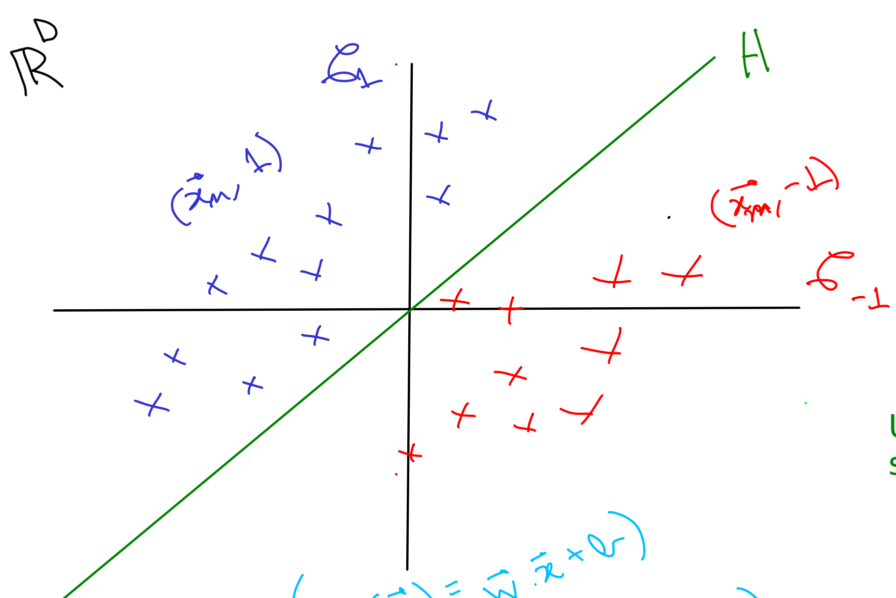
Base de datos:

$$\begin{array}{c} \vec{x}_1 \rightarrow \\ \\ \\ \end{array} \begin{pmatrix} x_{11} & \cdots & x_{1D} & \vdots & t_1 \\ \vdots & & & \vdots & \vdots \\ x_{N1} & \cdots & x_{ND} & \vdots & t_N \end{pmatrix}$$

$\vec{x}_n \in \mathbb{R}^D$

$t_n \in \{1, -1\}$ — 1: CLASE $\mathcal{C}_1$; -1: CLASE $\mathcal{C}_{-1}$

$$\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_{-1}$$



$(y(\vec{x}) = \vec{w} \cdot \vec{x} + b)$

$(H = \{ y(\vec{x}) = 0 \})$

$$H = \{ \underbrace{\vec{w} \cdot \vec{x} + b}_{y(\vec{x})} = 0 \}$$

$\vec{w} \in \mathbb{R}^D$, $\vec{x} \in \mathbb{R}^D$, $b \in \mathbb{R}$

$(w_1 x_1 + \dots + w_n x_n + b = 0)$

Un hiperplano es un hiperplano separador si:

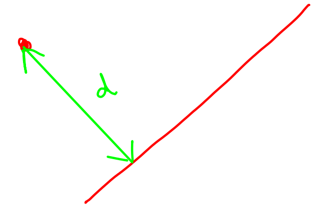
$$\mathcal{C}_1 \in \{ y(\vec{x}) > 0 \}$$

$$\mathcal{C}_{-1} \in \{ y(\vec{x}) < 0 \}$$

es decir,

$$\boxed{t_n \cdot y(\vec{x}_n) > 0} \quad \forall n$$

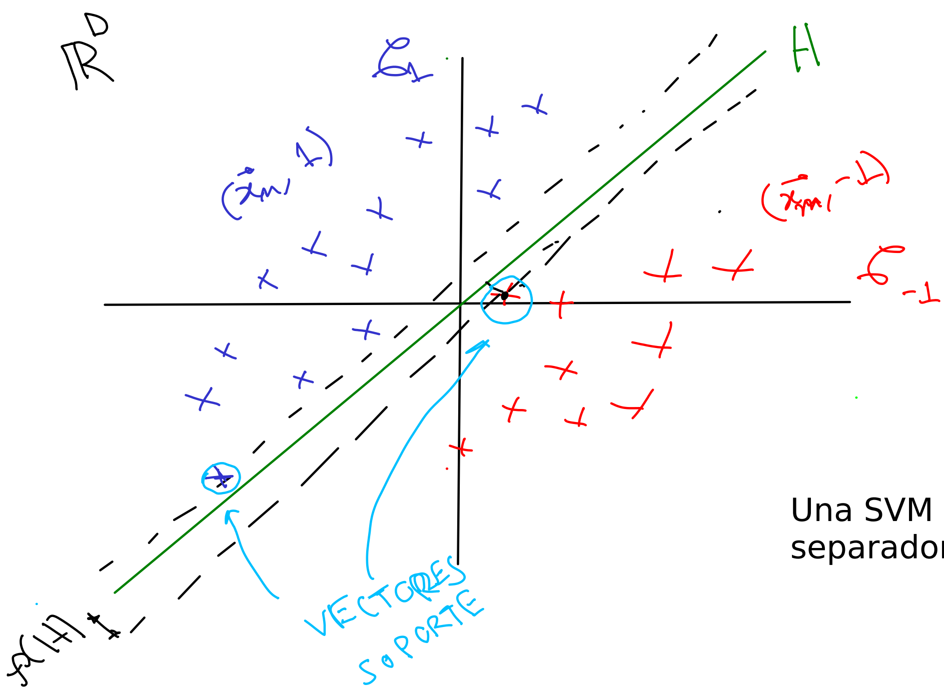
$$\boxed{t_n \cdot y(\vec{x}_n) > 1}$$



El margen de un hiperplano separador H es la distancia del conjunto de datos a H:

$$\rho(H) = \min_{n} d(\vec{x}_n, H)$$

↑ MARGEN



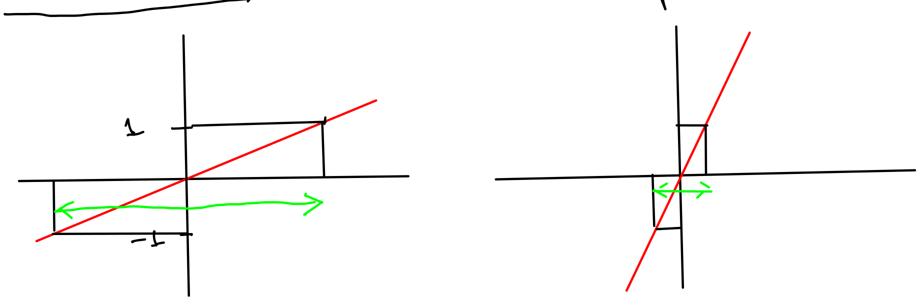
Una SVM calcula un hiperplano separador maximizando su margen.

Es decir, buscamos $\vec{w}$ tal que maximice $\rho(\{\vec{w}\cdot\vec{x}+b=0\})$

sujeto a $\boxed{t_n(\vec{w}\cdot\vec{x}+b) \geq 1}$.

COND. DE HIPERPLANO SEPARADOR

Observación: $\|\vec{w}\|$ da la pendiente del hiperplano



A mayor pendiente, menor margen

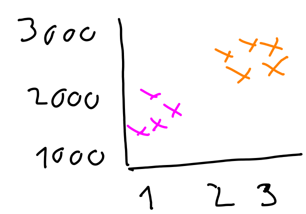
Nuestro problema es equivalente a:

$$
\begin{cases}
\text{minimizar} & \frac{1}{2}\|\vec{w}\|^2 = \frac{1}{2}\vec{w}\cdot\vec{w} = w_1^2 + \dots + w_D^2 \\
\text{sujeto a} & t_n(\vec{w}\cdot\vec{x}_n + b) \geq 1 \quad \forall n
\end{cases}
$$

PROBLEMAS DE LA SVM:

- Sensible a outliers

- Sensible a escalas



- Estamos asumiendo que el conjunto es

<u>LINEALMENTE SEPARABLE</u>

(es decir, que existe un hiperplano separador)

Posibles soluciones para el caso no linealmente separable:

(Para un caso que es "casi" linealmente separable):

$\boxed{\text{SOFT MARGIN}}$

- Introducimos VARIABLES DE HOLGURA ("SLACK")

$h_n \geq 0$.

- Sustituimos las condiciones $t_n y(\vec{x}_n) \geq 1$ $\rightsquigarrow$ $t_n y(\vec{x}_n) \geq 1 - \xi_n$ $\quad n = 1, \dots, N$

- Sustituyo la función de coste $\frac{1}{2}\|\vec{w}\|^2$ $\rightsquigarrow$ $\frac{1}{2}\|\vec{w}\|^2 + C \sum_{n=1}^{N} \xi_n$

$$
\begin{cases}
\min \ \frac{1}{2}\|\vec{w}\|^2 + C \sum_{n=1}^{N} \xi_n \\
\text{s.a.} \quad t_n y(\vec{x}) + \xi_n \geq 1 \\
\qquad\quad \xi_n \geq 0
\end{cases}
$$

C es un HIPERPARÁMETRO que controla la penalización que se impone a las variables de holgura

C alto: Margen pequeño con pocas violaciones
C bajo: Margen ancho con muchas violaciones

SVM NO LINEAL

El problema dual

Asociado al problema (PRIMAL)

$$
\begin{cases}
\min \ \frac{1}{2}\|\vec{w}\|^2 \\
\text{s.a.} \ t_n (\vec{w}\cdot\vec{x}_n + b) \geq 1 \quad \forall n
\end{cases}
$$

hay un problema DUAL equivalente

$$
\begin{cases}
\text{minimizar} \ \frac{1}{2}\sum a_n a_m t_n t_m \boxed{\vec{x}_n \vec{x}_m} - \sum_{n=1}^{N} a_n \\
\text{s.a.} \quad a_n \geq 0 \quad \forall n = 1, \dots, N \\
\qquad\quad \sum_{n=1}^{N} a_n t_n = 0
\end{cases}
$$

$\hat{a}$ óptimos $\rightsquigarrow$

$$
\hat{\vec{w}} = \sum_{n=1}^{N} \hat{a}_n t_n \vec{x}_n
$$

$$
\hat{b} = \frac{1}{\# S} \sum_{n \in S} t_n - \vec{w}\cdot\vec{x}_n
$$

$$
S = \{ n \mid \hat{a}_n > 0 \}
$$

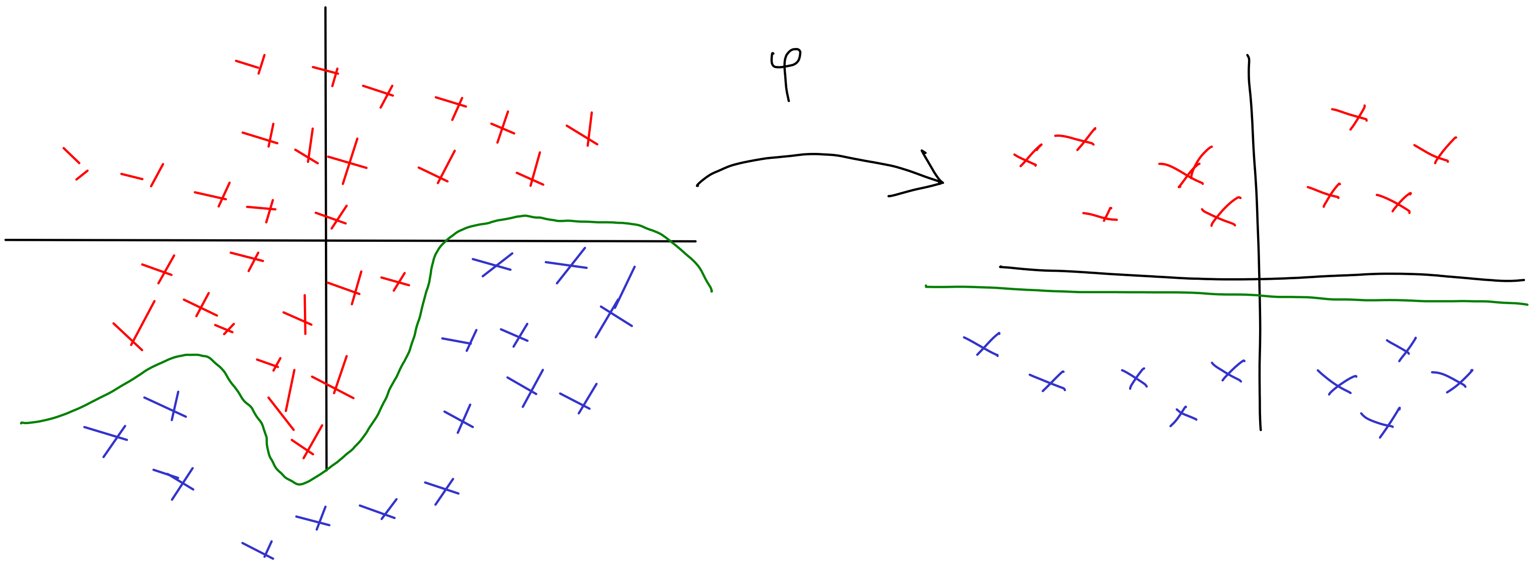
COSTES COMPUTACIONALES

PRIMAL — $O(D^3)$ $\quad D = \#$ FEATURES

DUAL — $O(N^3)$ $\quad N = \#$ INSTANCIAS

Pero para lo que realmente nos interesa el problema dual es para poder usar el "kernel trick".

NÚCLEOS



Supongamos que tenemos un training set $\{\vec{x}_n\} \subset \mathbb{R}^2$

y le aplicamos una transformación polinomial de grado 2

$$\mathbb{R}^2 \xrightarrow{\ \varphi\ } \mathbb{R}^3$$
$$(x, y) \longmapsto (x^2, \sqrt{2}\,xy, y^2)$$

para "linealizar" el conjunto de datos.

Sucede lo siguiente

$$\underset{(a_1,a_2)}{\varphi(\vec{a})} \cdot \underset{(b_1,b_2)}{\varphi(\vec{b})} = (a_1^2, \sqrt{2}a_1a_2, a_2^2)\cdot(b_1^2, \sqrt{2}b_1b_2, b_2^2) = a_1^2b_1^2 + 2a_1a_2b_1b_2 + a_2^2b_2^2 =$$
$$= (a_1b_1 + a_2b_2)^2 = \boxed{(\vec{a}\cdot\vec{b})^2}$$

En el problema dual, basta reemplazar $\vec{x}_n\cdot\vec{x}_m$ por $(\vec{x}_n\cdot\vec{x}_m)^2$

¡no hace falta aplicar la transformación φ!

$K(\vec{a},\vec{b}) = (\vec{a}\cdot\vec{b})^2$ se llama NÚCLEO POLINOMIAL DE GRADO 2 (O CUADRÁTICO)

En general:

$$K:\mathbb{R}^D \times \mathbb{R}^D \longrightarrow \mathbb{R}$$
$$(\vec{x}_n, \vec{x}_m) \longmapsto K(\vec{x}_n, \vec{x}_m)$$

es un **núcleo** si existe una transformación $\varphi: \mathbb{R}^D \longrightarrow \mathbb{R}^E$ tal que

$$K(\vec{x}_n, \vec{x}_m) = \varphi(\vec{x}_n)\cdot\varphi(\vec{x}_m)$$

En resumen, utilizar un núcleo en la función de coste es equivalente a hacer una transformación del dataset.

NÚCLEOS COMUNES:

- **Lineal**: $K(\vec{a},\vec{b}) = \vec{a}\cdot\vec{b}$
- **Polinomial** (de grado d): $K(\vec{a},\vec{b}) = (\gamma\,\vec{a}\cdot\vec{b} + r)^d$
- **Gaussiano o radial**: $K(\vec{a},\vec{b}) = \exp\left(-\gamma\,\|\vec{a}-\vec{b}\|^2\right)$
- **Sigmoide**: $K(\vec{a},\vec{b}) = \tanh(\gamma\,\vec{a}\cdot\vec{b} + r)$

RED NEURONAL

Se usa, pero no es un verdadero núcleo