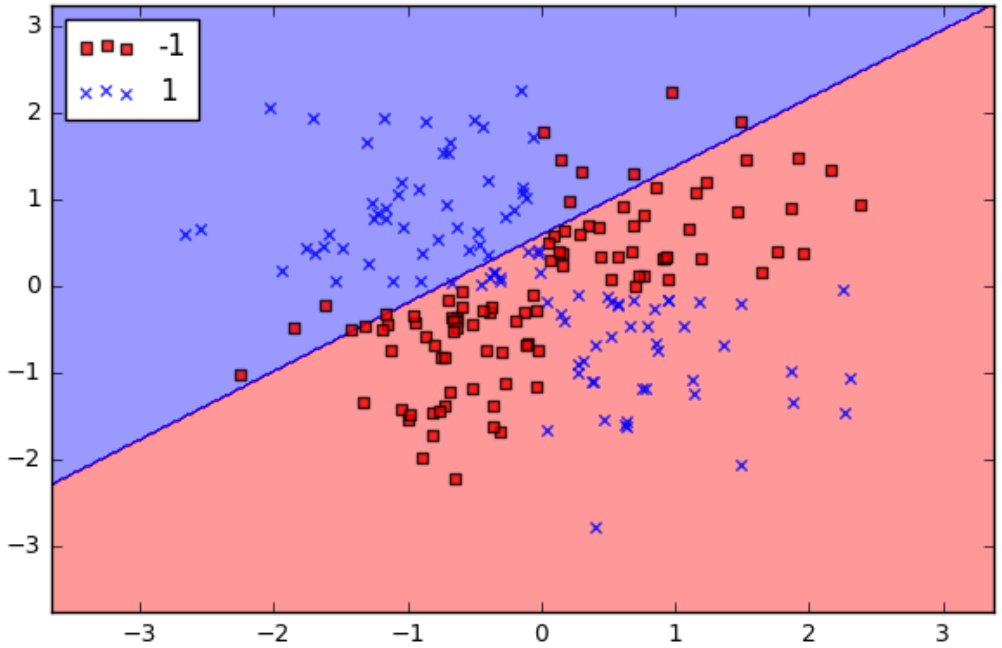


SVM (clasificación-regresión)

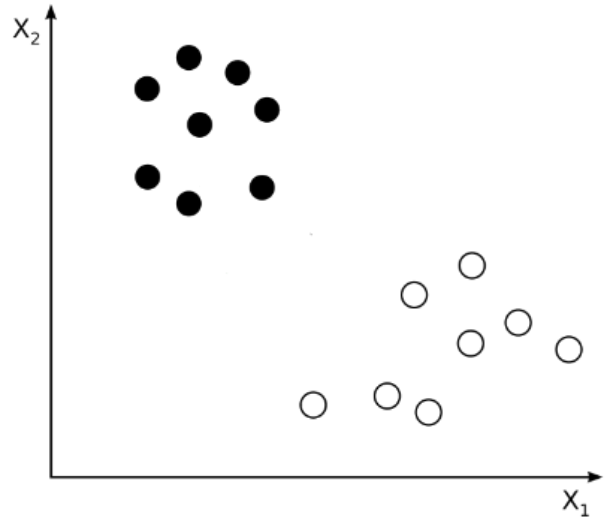
PhD(e). Jonnatan Arias Garcia – jonnatan.arias@utp.edu.co – jariasg@uniquindio.edu.co

PhD. David Cardenas peña - dcardenasp@utp.edu.co

PhD. Hernán Felipe Garcia - hernanf.garcia@udea.edu.co

Support Vector Machines

- (*Máquinas de Vectores de Soporte*) son un conjunto de versátiles y potentes algoritmos desarrollados en los laboratorios AT&T, útiles tanto en escenarios de clasificación, clúster y regresión.
- La idea detrás de SVM Partimos de un conjunto de puntos en un plano que pertenecen a dos clases distintas:



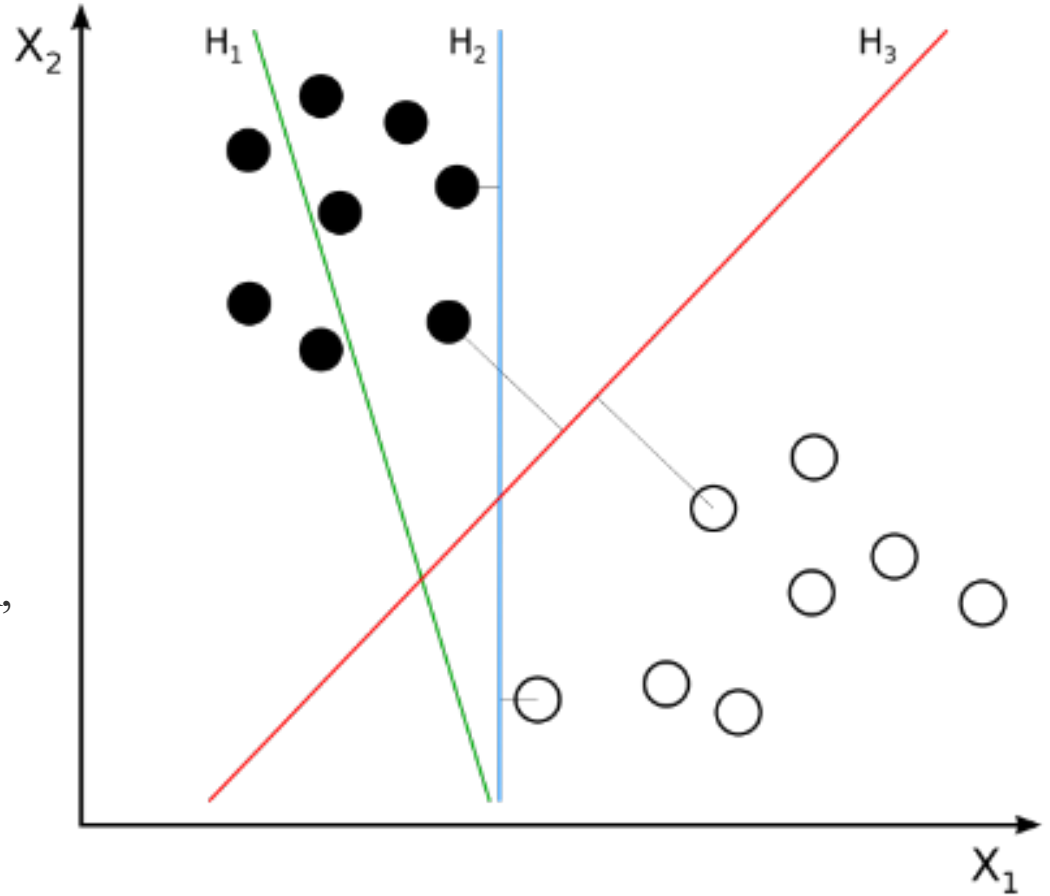
Support Vector Machines

El objetivo es encontrar una recta (un hiperplano, en general) que permita separar ambos bloques de puntos. En el caso mostrado, existe un número infinito de rectas candidatas a resolver el problema:

La recta H1 no divide correctamente los dos bloques

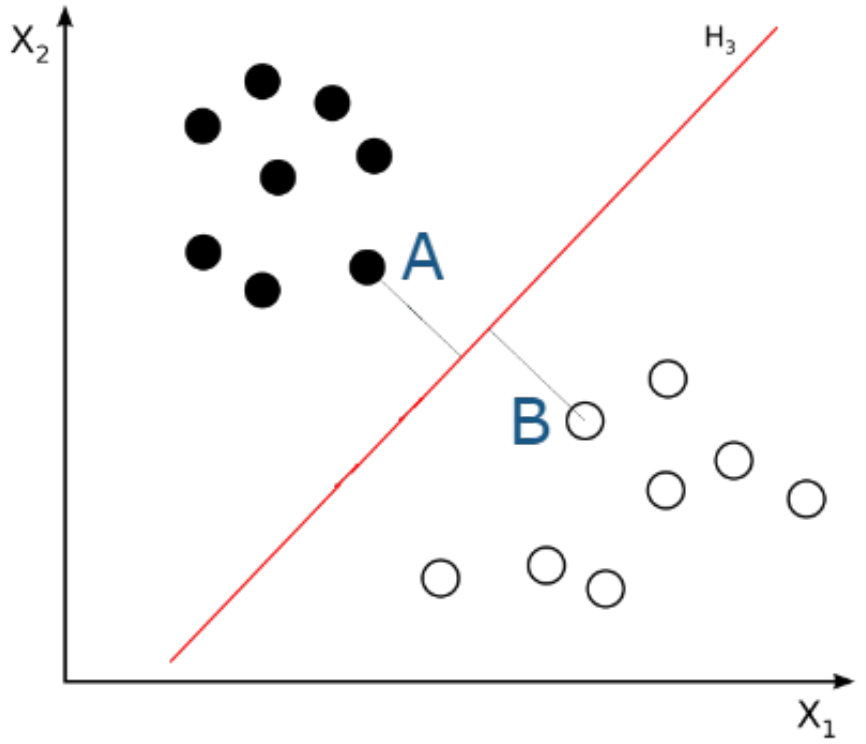
La recta H2 sí lo hace, pero su proximidad a los puntos negros hace que difícilmente pueda generalizarse el resultado (puede existir en el conjunto de puntos sobre los que realizar la predicción alguno que quede cerca de los puntos negros)

La recta H3, desde cierto punto de vista, es la recta ideal, pues maximiza la distancia mínima a los puntos negros y blancos, optimizando la capacidad de generalización del algoritmo:



Support Vector Machines

Los puntos A y B, los más próximos a la recta H3, son los que determinan la posición de la recta. Si esta recta existe (si los puntos son linealmente separables) se denomina *hiperplano de máximo margen* (*maximum-margin hyperplane*), y los puntos A y B se denominan vectores de soporte (*support vectors*). La suma de las distancias que separan los puntos A y B del hiperplano de máximo margen se denomina *margen*.

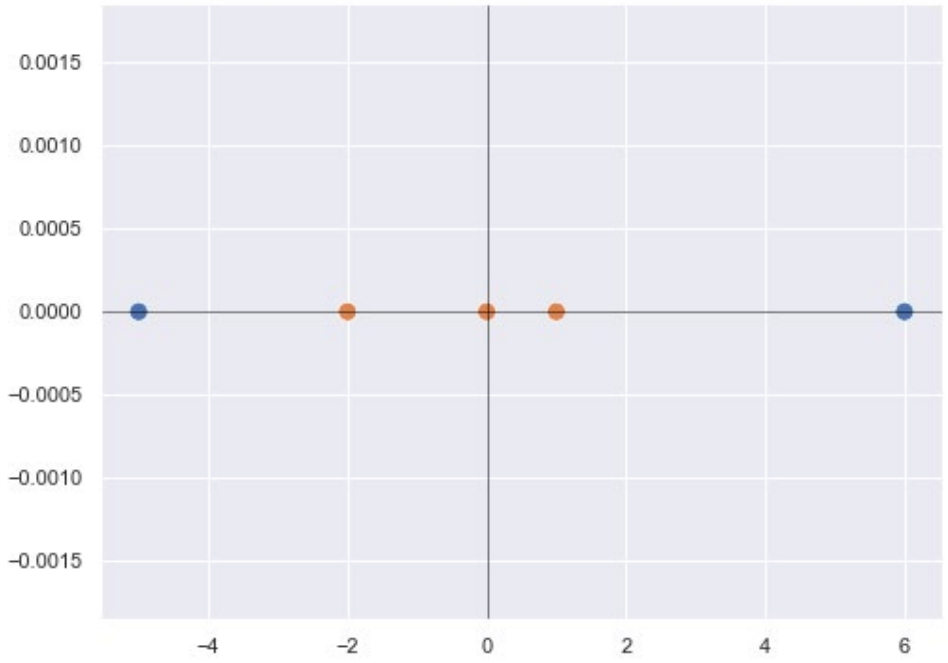


El kernel Trick

No siempre van a ser linealmente separables.

Ejemplo, Existe un plano que separe puntos azules de naranjas?

Solución?



El kernel Trick

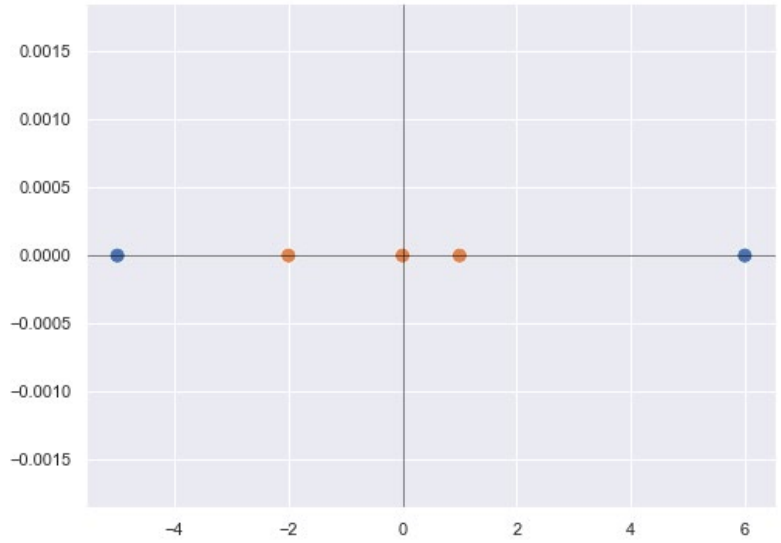
Solución?

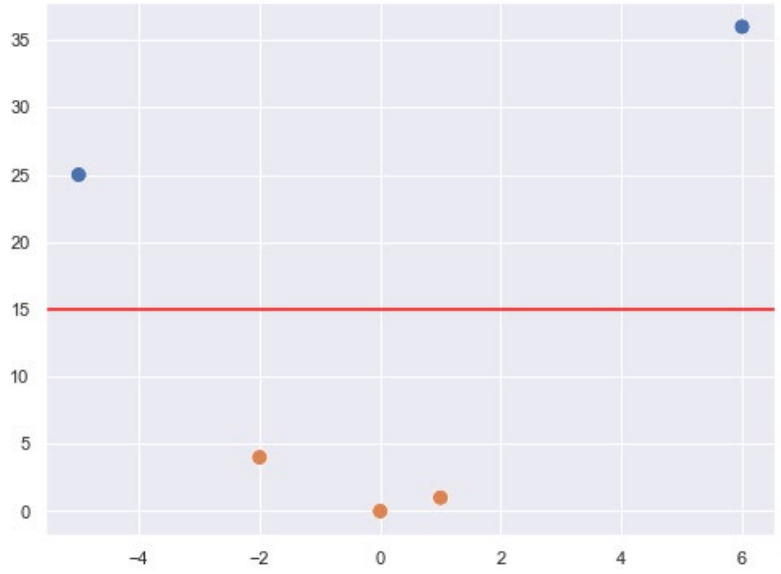
Transformar los puntos a un espacio de mayor dimensionalidad donde sean separables.

Por ejemplo, pasar los punto (dim=1) a un plano (dim=2)

X a XX

-5, -2, 0, 1, 6 -> 25, 4, 0, 1, 36

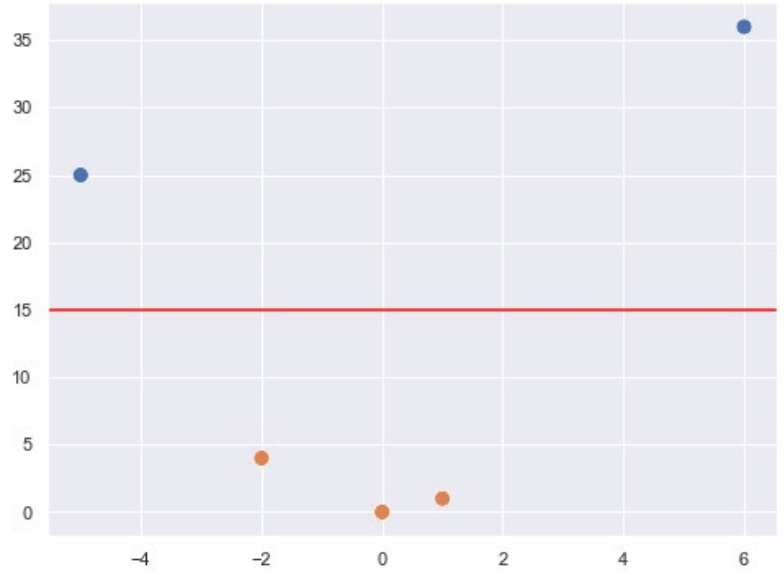




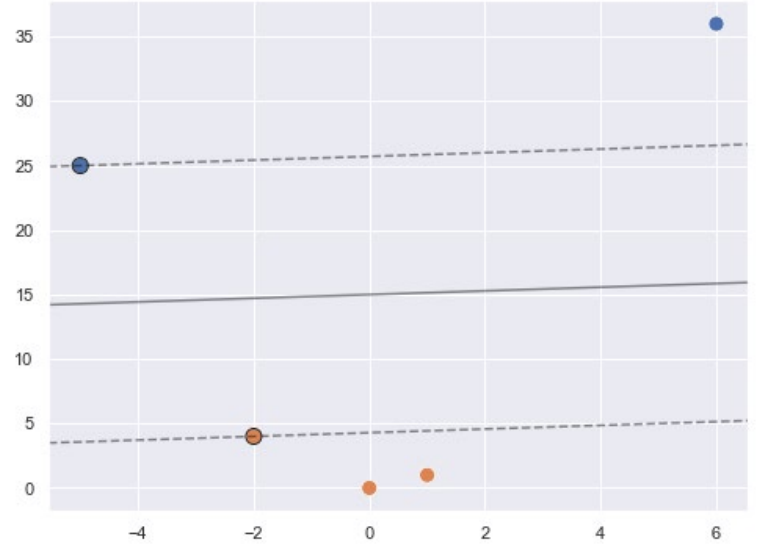
El kernel Trick

En el nuevo espacio, es posible crear una recta que separe ambos grupos de puntos, por el ejemplo la recta (hay un infinito número de ellas, de hecho).

SVM buscaría la recta que maximizase la distancia a ambos grupos de puntos en este nuevo espacio.



Podemos distinguir el hiperplano de máximo margen ?
Y los vectores de soporte ?



Esta transformación de las muestras desde el espacio original hasta una espacio de mayor dimensionalidad es llamada "**kernel trick** "

Márgenes duros y blandos

- Si, tras aplicar el Kernel Trick los puntos que queremos clasificar son linealmente separables. Sin embargo, puede ocurrir que las clases no sean linealmente separables, no será posible encontrar el hiperplano en cuestión.
- "*Margen duro*" hace referencia al escenario en el que no se permiten errores en el entrenamiento: si se encuentra un hiperplano de máximo margen es porque clasifica correctamente todas las muestras. Si las clases no son linealmente separables, resulta más práctico permitir ciertos errores en la clasificación a cambio de poder seguir encontrando el hiperplano de máximo margen. Este segundo enfoque es el que denominamos "*de margen blando*".
- Margen Blando -determinado por la función de coste del algoritmo- está regulado con el parámetro C:
  - Un valor mayor de C implica un coste mayor derivado de las muestras mal clasificadas.
  - Un valor menor implica que las muestras mal clasificadas van a suponer un coste menor, por lo que se tiende a un escenario en el que se permite un mayor número de errores.

Kernels

Función de transformación que permite aumentar dimensionalidad y dar posiblemente un plus en la separación de datos.

Un kernel viene determinado por una matriz cuadrada de dimensión igual al número de muestras siendo analizadas, y el contenido de esta matriz es la resultante de aplicar la función de mapeo al conjunto de entrenamiento.

Linear

La función aplicada es:

$$K(x_i, x_j) = x_i^T . x_j$$

Kernels

Linear

La función aplicada es:

$$K(x_i, x_j) = x_i^T . x_j$$

Polynomial

En el kernel polinómico la función aplicada es:

$$K(x_i, x_j) = (\gamma x_i^T . X_j + r)^d$$

...donde γ viene dada por el parámetro **gamma** de la función SVC, r por **coef0** y d por el parámetro **degree**.

Kernels

Radial Basis Function

La función aplicada es:

$$K(x_i, x_j) = e^{-\gamma \|x_i x_j\|^2}$$

$\|X_i - X_j\|$ representa una matriz cuadrada con la distancia euclídea entre cada par de puntos de X. Esta matriz puede conseguirse con la función sklearn.metrics.pairwise.euclidean_distances. Al igual que en el caso del kernel polinómico, γ viene dado por el parámetro **gamma** de SVC.

Sigmoid

En este último caso la función aplicada es:

$$K(x_i, x_j) = \tanh(\gamma x_i^T . x_j + r)$$

...donde γ y r vienen dados por los parámetros **gamma** y **coef0**, respectivamente.

Matemática del método

Máxima margen

- Modelos

$$y(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\varphi}(\boldsymbol{x}) + b$$

Donde la decisión es

$$t^* = sign\big(y(\boldsymbol{x})\big), \boldsymbol{w}, b$$

La margen es el valor $y(\boldsymbol{x_n})$

La distancia de un punto a un (hiper)plano:

$$d(\boldsymbol{x_n}, \boldsymbol{w}) = \frac{t_n y_n}{||\boldsymbol{w}||} = \frac{t_n(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\varphi}(\boldsymbol{x}) + b\ )}{||\boldsymbol{w}||}$$

donde t_n se incluye para indicar si la muestra está bien clasificada. De los modelos lineales sabemos que una muestra bien clasificada cumple $y(\boldsymbol{x}_n)t_n > 0$ porque los signos coinciden.

- Para que el error sea mínimo, la margen debe ser máxima, es decir, el punto **más cercano** a la frontera debe estar **lo más lejos posible.** El punto más cercano a la frontera (sin importar la clase) estará a distancia:

$$\min_n d(\boldsymbol{x}_n, \boldsymbol{w}) = \min_n \frac{t_n(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_n) + b)}{\|\boldsymbol{w}\|}$$

- Como queremos que esta distancia sea lo más grande posible, una *proto*función de costo para maximizar la margen es:

$$\max_{\boldsymbol{w},b} \min_n d(\boldsymbol{x}_n, \boldsymbol{w}) = \max_{\boldsymbol{w},b} \left\{ \min_n \frac{t_n(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_n) + b)}{\|\boldsymbol{w}\|} \right\}$$

$$\max_{\boldsymbol{w},b} \min_n d(\boldsymbol{x}_n, \boldsymbol{w}) = \max_{\boldsymbol{w},b} \left\{ \frac{1}{\|\boldsymbol{w}\|} \min_n t_n(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_n) + b) \right\}$$

- Como escalar $\boldsymbol{w}, b$ no afecta la distancia, $d(\boldsymbol{x}_n, k\boldsymbol{w}, kb) = d(\boldsymbol{x}_n, \boldsymbol{w}, b)$, facilitamos la solución del problema imponiendo una restricción para aquellas muestras que esten sobre la margen:

$$t_n y_n = t_n(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_n) + b) = 1$$

- Con esta restricción, la *proto*función se vuelve el problema primal de optimización:

$$\max_{\boldsymbol{w},b} \min_{n} d(\boldsymbol{x}_n, \boldsymbol{w}) = \max_{\boldsymbol{w},b} \left\{ \frac{1}{\|\boldsymbol{w}\|} \min_{n} 1 \right\} = \max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|} = \min_{\boldsymbol{w},b} \frac{1}{2} \|\boldsymbol{w}^2\|$$

$$\min_{\boldsymbol{w},b} \frac{1}{2} \|\boldsymbol{w}\|^2$$

$$\text{s.t. } t_n(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_n) + b) \geq 1 \; ; n = 1, \dots, N$$

- Para optimizar con restricciones usamos el Lagrangiano:

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2} \|\boldsymbol{w}^2\| - \sum_{n=1}^{N} a_n \{t_n y(\boldsymbol{x}_n) - 1\}$$

donde $\boldsymbol{a} \in \mathbb{R}^N$ es el vector de los N multiplicadores de Lagrange a_n (uno por cada restricción).

- Para optimizar con restricciones usamos el Lagrangiano:

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{w}^2\| - \sum_{n=1}^{N} a_n\{t_n y(\boldsymbol{x}_n) - 1\}$$

donde $\boldsymbol{a} \in \mathbb{R}^N$ es el vector de los N multiplicadores de Lagrange a_n (uno por cada restricción).

- Tomamos derivada respecto al vector de pesos:

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial \boldsymbol{w}} = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^\top \boldsymbol{w} - \frac{\partial}{\partial \boldsymbol{w}}\sum_{n=1}^{N} a_n\{t_n(\boldsymbol{w}^\top \boldsymbol{\varphi}_n + b) - 1\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{w}}\{a_n t_n \boldsymbol{w}^\top \boldsymbol{\varphi}_n + t_n a_n b - a_n\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{n=1}^{N} a_n t_n \boldsymbol{\varphi}_n = 0$$

$$\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\varphi}_n \quad (1)$$

- Tomamos derivada respecto al intercepto:

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial b} = \frac{\partial}{\partial b} \sum_{n=1}^{N} a_n \{t_n(\boldsymbol{w}^\top \boldsymbol{\varphi}_n + b) - 1\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial b} = \sum_{n=1}^{N} \frac{\partial}{\partial b} \{a_n t_n \boldsymbol{w}^\top \boldsymbol{\varphi}_n + a_n t_n b - a_n\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial b} = \sum_{n=1}^{N} a_n t_n = 0 \quad (2)$$

- Reemplazamos estos dos resultados en el Lagrangiano:

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n \{t_n y(\boldsymbol{x}_n) - 1\}$$

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n t_n y(\boldsymbol{x}_n) + \sum_{n=1}^{N} a_n \quad (5)$$

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} - \sum_{n=1}^{N} a_n t_n (\boldsymbol{w}^\top \boldsymbol{\varphi}_n + b) + \sum_{n=1}^{N} a_n$$

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} - \sum_{n=1}^{N} a_n t_n \boldsymbol{w}^\top \boldsymbol{\varphi}_n - \sum_{n=1}^{N} a_n t_n b + \sum_{n=1}^{N} a_n \quad (0)$$

- El tercer término se elimina usando la condición de *equilibrio*:

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} - \sum_{n=1}^{N} a_n t_n \boldsymbol{w}^\top \boldsymbol{\varphi}_n + \sum_{n=1}^{N} a_n \quad (1)$$

- Reemplazamos la ecuación de $\boldsymbol{w}$ en términos de $\boldsymbol{a}$:

$$L(\boldsymbol{a}) = \frac{1}{2}\left(\sum_{n=1}^{N} a_n t_n \boldsymbol{\varphi}_n\right)^{\top}\left(\sum_{m=1}^{N} a_m t_m \boldsymbol{\varphi}_m\right) - \sum_{n=1}^{N} a_n t_n \left(\sum_{m=1}^{N} a_m t_m \boldsymbol{\varphi}_m\right)^{\top} \boldsymbol{\varphi}_n + \sum_{n=1}^{N} a_n$$

$$L(\boldsymbol{a}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n t_n t_m \boldsymbol{\varphi}_n^{\top}\boldsymbol{\varphi}_m a_m - \sum_{n=1}^{N}\sum_{m=1}^{N} a_n t_n t_m \boldsymbol{\varphi}_n^{\top}\boldsymbol{\varphi}_m a_m + \sum_{n=1}^{N} a_n$$

$$L(\boldsymbol{a}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n t_n t_m \boldsymbol{\varphi}_n^{\top}\boldsymbol{\varphi}_m a_m + \sum_{n=1}^{N} a_n$$

- Hemos llegado al problema dual, restringido a las dos ecuaciones que reemplazamos en el primal:

$$L(\boldsymbol{a}) = -\frac{1}{2}\boldsymbol{a}^{\top}(\boldsymbol{T} \circ \boldsymbol{K})\boldsymbol{a} + \mathbf{1}_N^{\top}\boldsymbol{a}$$
$$s.t. \quad \boldsymbol{a} \geq \mathbf{0}$$
$$\boldsymbol{t}^{\top}\boldsymbol{a} = 0$$

quadprog
linear restrictions.
$K_{ij} = x_i^{\top} x_j$; $K_{ij} = K(x_i, x_j)$

correspondiente a un problema de optimización cuadrática con restricciones de igualdad y desigualdad lineales donde:

- $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ es una matriz con todos los productos internos entre parejas de muestras de la base de datos de entrenamiento en el RHKS $K_{nm} = \boldsymbol{\varphi}_n^{\top}\boldsymbol{\varphi}_m = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$, también conocida como *matriz Gram*.
- $\boldsymbol{T} \in \{+1, -1\}^{N \times N}$ es la matriz objetivo con elementos $T_{nm} = t_n t_m$ comparando las etiquetas de las muestras.
- $\boldsymbol{T} \circ \boldsymbol{K}$ es el producto de Haddamard (elemento a elemento).
- $\mathbf{1}_N \in \mathbb{R}^N$ es un vector de unos.

- La interpretación gráfica del resultado es:

- Además, reemplazando la ecuación del vector que define el hiperplano en el modelo:

$$y(\boldsymbol{x}^*) = \boldsymbol{w}^\top \varphi(\boldsymbol{x}^*) + b$$ modelo primal

(1)

$$y(\boldsymbol{x}^*) = \sum_{n=1}^{N} a_n t_n \boldsymbol{\varphi}_n{}^\top \boldsymbol{\varphi}^* + b$$

$$y(\boldsymbol{x}^*) = \sum_{n=1}^{N} a_n t_n \, k(\boldsymbol{x}_n, \boldsymbol{x}^*) + b$$ modelo dual

- Según las condiciones de optimalidad:
  - $a_n \geq 0$
  - $t_n y(\boldsymbol{x}_n) - 1 \geq 0$
  - $a_n\{t_n y(\boldsymbol{x}_n) - 1\} = 0 \quad \forall n$

- Pero, sin que a_n y $t_n y(\boldsymbol{x}_n) - 1$ sean 0 simultáneamente. Por lo tanto:
  - Si $a_n = 0$, entonces $t_n y(\boldsymbol{x}_n) > 1$, es decir, $\boldsymbol{x}_n$ está fuera de la margen y NO suma para las predicciones.
  - Si $a_n > 0$, entonces $t_n y(\boldsymbol{x}_n) = 1$, es decir, $\boldsymbol{x}_n$ está en la margen y SI suma para las predicciones. $\boldsymbol{x}_n$ se vuelve un soporte para construir el vector $\boldsymbol{w}$, **un vector de soporte** para la máquina:

$$y(\boldsymbol{x}^*) = \sum_{n \in SV} a_n t_n \, k(\boldsymbol{x}_n, \boldsymbol{x}^*) + b$$ modelo dual final

- Finalmente, el intercepto b se puede calcular sabiendo que todo vector de soporte vive sobre la margen y cumple $y(\boldsymbol{x}_n)t_n = 1$:

$$t_m \left(\sum_{n \in SV} a_n t_n \, k(\boldsymbol{x}_n, \boldsymbol{x}_m) + b \right) t_m = 1 \cdot t_m \quad \forall \boldsymbol{x}_m \in SV$$

(y_m)

$$y_n = \begin{cases} 1 & \boldsymbol{x}_n \in SV, \; t_n = 1 \\ -1 & \boldsymbol{x}_n \in SV, \; t_n = -1 \end{cases}$$

- Multiplicamos por t_m y sumamos para todos los SV:

$$\sum_{m \in SV} \left(\sum_{n \in SV} a_n t_n \, k(\boldsymbol{x}_n, \boldsymbol{x}_m) + b \right) = \sum_{m \in SV} t_m$$

$$\sum_{m \in SV} \sum_{n \in SV} a_n t_n \, k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \sum_{m \in SV} b = \sum_{m \in SV} t_m$$

$$b N_{SV} = \sum_{m \in SV} \left\{ t_m - \sum_{n \in SV} a_n t_n \, k(\boldsymbol{x}_n, \boldsymbol{x}_m) \right\}$$

$$b = \frac{1}{N_{SV}} \sum_{m \in SV} \left\{ t_m - \sum_{n \in SV} a_n t_n \, k(\boldsymbol{x}_n, \boldsymbol{x}_m) \right\}$$

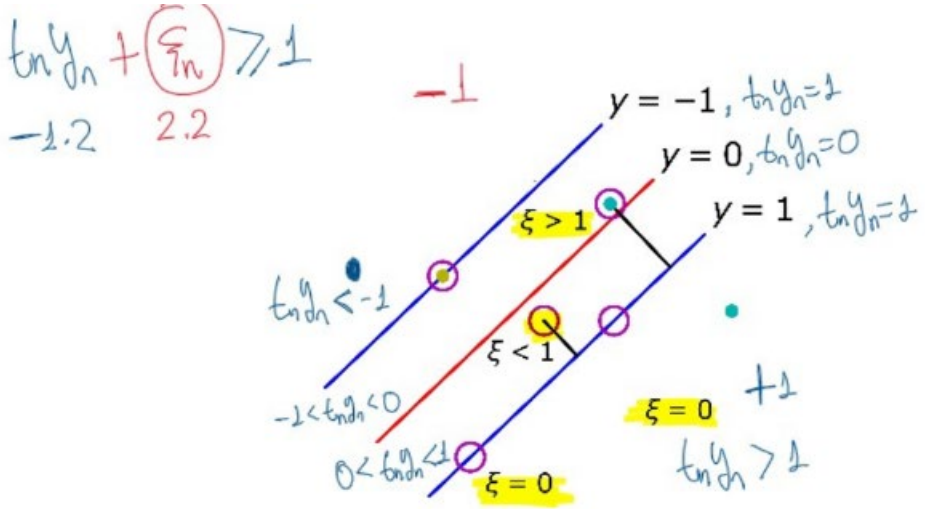
Margen Suave

- Si existe una separación lineal de las muestras en el RKHS, la SVM de margen dura la encontrará, así se vea no-lineal en el espacio de entrada.

- Sin embargo, si NO existe separación lineal en el RKHS, el problema de optimización NO tendrá solución.

- Además, si existe traslape en las distribuciones condicionales de clase en el espacio de entrada, en el RKHS se forzará la separación lineal, generando sobre-ajustes.

- Para evitar el sobre-ajuste, se permite que existan muestras dentro de la margen **con una penalidad**, haciendo que la margen sea suave.



- Agregamos una **variable de holgura** (*slack variable*), $\xi_n \geq 0$, para cada muestra tal que complete lo que le falta a la muestra para cumplir la restricción del problema primal de la SVM con margen dura:

$$t_n y(\boldsymbol{x}_n) + \xi_n \geq 1$$

$$\xi_n \geq 0$$

$\xi_n = 0$	Para muestras en el lado correcto de la margen.
$0 < \xi_n < 1$	Para muestras entre la frontera y la margen de la clase correcta.
$\xi_n = 1$	Para muestras sobre la frontera porque $y(\boldsymbol{x}_n) = 0$.
$\xi_n > 1$	Para muestras entre la frontera y la margen de la clase incorrecta.

- Puesto que las penalizaciones deben ser las mínimas, el nuevo problema primal es:

$$\min_{\boldsymbol{w},b} C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2$$

s.t. $t_n y_n + \xi_n \geq 1 \quad \forall n$

$\xi_n \geq 0 \quad \forall n$

donde $C > 0$ actúa como parámetro de balance/*trade-off*.

- El nuevo Lagrangiano será:

$$L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu}) = C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n\{t_n y(\boldsymbol{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n$$

- Tomamos derivada respecto al vector de pesos:

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \frac{\partial}{\partial \boldsymbol{w}}\sum_{n=1}^{N} a_n\{t_n(\boldsymbol{w}^\top\boldsymbol{\varphi}_n + b) - 1 + \xi_n\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{n=1}^{N}\frac{\partial}{\partial \boldsymbol{w}}\{a_n t_n \boldsymbol{w}^\top\boldsymbol{\varphi}_n + t_n a_n b - a_n + a_n\xi_n\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{n=1}^{N} a_n t_n \boldsymbol{\varphi}_n = 0$$

$$\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\varphi}_n \quad (1)$$

- Tomamos derivada respecto al intercepto:

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial b} = \frac{\partial}{\partial b}\sum_{n=1}^{N} a_n\{t_n(\boldsymbol{w}^\top\boldsymbol{\varphi}_n + b) - 1 + \xi_n\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial b} = \sum_{n=1}^{N}\frac{\partial}{\partial b}\{a_n t_n \boldsymbol{w}^\top\boldsymbol{\varphi}_n + a_n t_n b - a_n + a_n\xi_n\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial b} = \sum_{n=1}^{N} a_n t_n = 0 \quad (2)$$

- Tomamos derivada respecto a las holguras:

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial \xi_m} = \frac{\partial}{\partial \xi_m}\left\{ C\sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n\{t_n y(\boldsymbol{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N} \mu_n \xi_n \right\}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial \xi_m} = C - a_m - \mu_m = 0$$

$$a_m \leq C \qquad a_m + \mu_m = C$$

$$\rightarrow \; 0 \leq a_m \leq C$$

- Reemplazamos estos tres resultados en el Lagrangiano:

$$L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu}) = \sum_{n=1}^{N} C\xi_n - \sum_{n=1}^{N} \mu_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n\{t_n y(\boldsymbol{x}_n) - 1 + \xi_n\}$$

$$L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu}) = \sum_{n=1}^{N} (C - \mu_n)\xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n t_n y(\boldsymbol{x}_n) + \sum_{n=1}^{N} a_n - \sum_{n=1}^{N} a_n \xi_n$$

$$L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}, \boldsymbol{\mu}) = \sum_{n=1}^{N} (C - \mu_n - a_n)\xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n t_n y(\boldsymbol{x}_n) + \sum_{n=1}^{N} a_n$$

$$L(\boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n t_n y(\boldsymbol{x}_n) + \sum_{n=1}^{N} a_n$$

- Volvemos al problema dual de la SVM con margen dura pero incluyendo una restricción más:

$$L(\boldsymbol{a}) = -\frac{1}{2}\boldsymbol{a}^{\mathrm{T}}(\boldsymbol{T} \circ \boldsymbol{K})\boldsymbol{a} + \boldsymbol{1}^{\mathrm{T}}\boldsymbol{a}$$

$$s.t. \quad 0 \leq a_n \leq C$$

$$\boldsymbol{t}^{\mathrm{T}}\boldsymbol{a} = 0$$

- Conclusión: Puesto que las muestras mal clasificadas tienen $\xi_n > 1$, la suma de las holguras es una cota superior al error de clasificación. Entonces:
  - Si C aumenta, se le da más peso al término de holguras.
  - Si $C \to \infty$, la máquina elimina por completo las holguras y vuelve a la solución de margen dura a costa de un modelo más complejo.
  - Si C disminuye, las holguras no son tan importantes. Pesa más la simpleza del modelo a costa de muchos "préstamos".
  - Por lo tanto, C se comporta como un parámetro de regularización inverso.