SVM

Support Vector Machine

SVM

Las **máquinas de soporte vectorial (Support Vector Machine - SVM)** tienen su origen en los años 90 por Vapnik y sus colaboradores.

Las SVM originalmente son clasificadores lineales, ya que crean separadores lineales o hiperplanos, tanto en el espacio original como en el espacio de características.

- Separadores lineales en el espacio original si el problema es linealmente separable.

- Separadores lineales en el espacio de características si el problema no es linealmente separable.

SVM

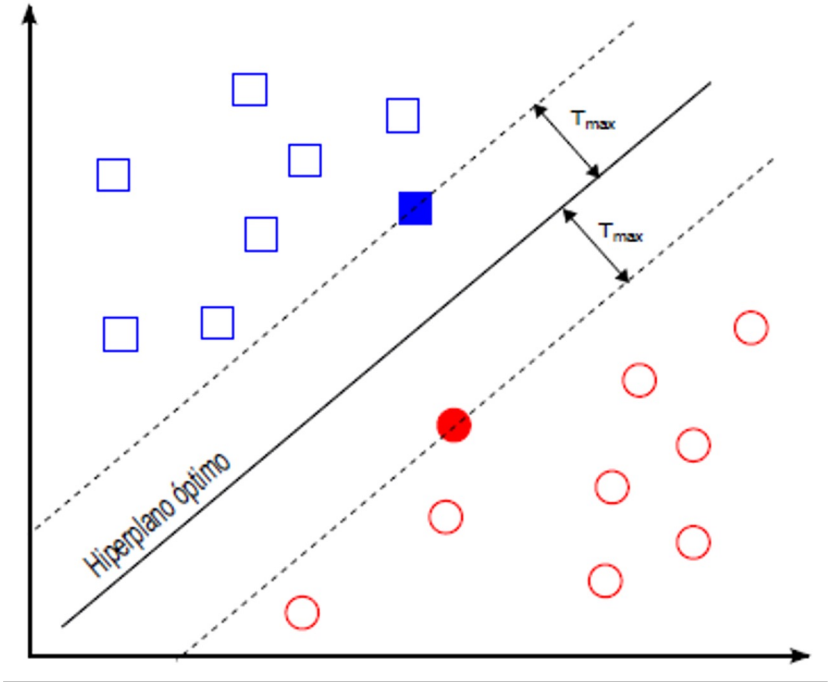
La mayoría de los métodos de aprendizaje se centra en la minimización de los errores empíricos (error de entrenamiento).

En cambio, las SVM se centra en la minimización del riesgo estructural:

- Elegir un hiperplano separador que equidista de los ejemplos más cercanos de cada clase.
- Maximizar la distancia del hiperplano a estos ejemplos más cercanos por ambos lados (clasificación binaria).
- Estos ejemplos son los llamados: vectores soporte

SVM

El encontrar el hiperplano se reduce a un problema de optimización cuadrática con restricciones lineales que debe cumplir con la condición de convexidad.



SVM

En esta presentación se verá 3 formulaciones de la SVM para clasificación binaria:

- SVM para ejemplos linealmente separables.
- SVM para ejemplos cuasi-linealmente separables.
- SVM para ejemplos no separables linealmente.

SVM para ejemplos linealmente separables.

SVM

Dado un conjunto de ejemplos $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ donde $x_i \in \mathbb{R}^D$ y $y_i \in \{-1, 1\}$.

Se puede definir un hiperplano separador como una función lineal capaz de separar dicho conjunto de ejemplos sin error:

$$D(x) = (w_1 x_1 + ... + w_D x_D) + b = < w, x > + b$$

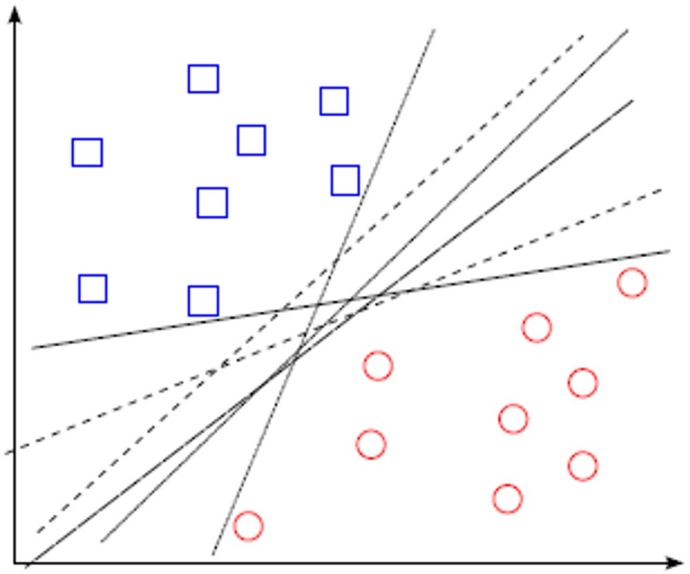
donde w y b son coeficientes reales y el hiperplano debe cumplir con:

$$< w, x > + b \geq 0 \quad si \quad y_i = 1$$

$$< w, x > + b \leq 0 \quad si \quad y_i = -1 \quad i = 1...n$$

SVM

Pero, el modelo anterior puede generar infinitos hiperplanos separadores.

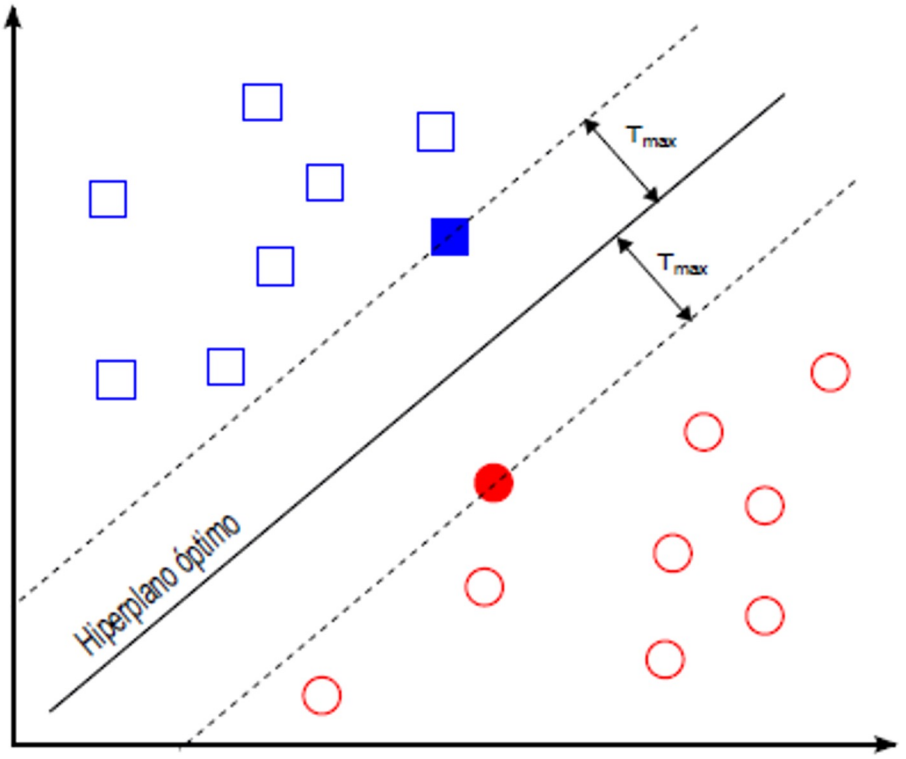


Entonces, se debe aplicar un criterio que permita encontrar el mejor hiperplano.

SVM

Este criterio es el que máxima el margen: distancia entre el hiperplano y los vectores soporte.

Volviendo a la primera imagen:



SVM

Por geometría, la distancia a un hiperplano y un punto x' es:

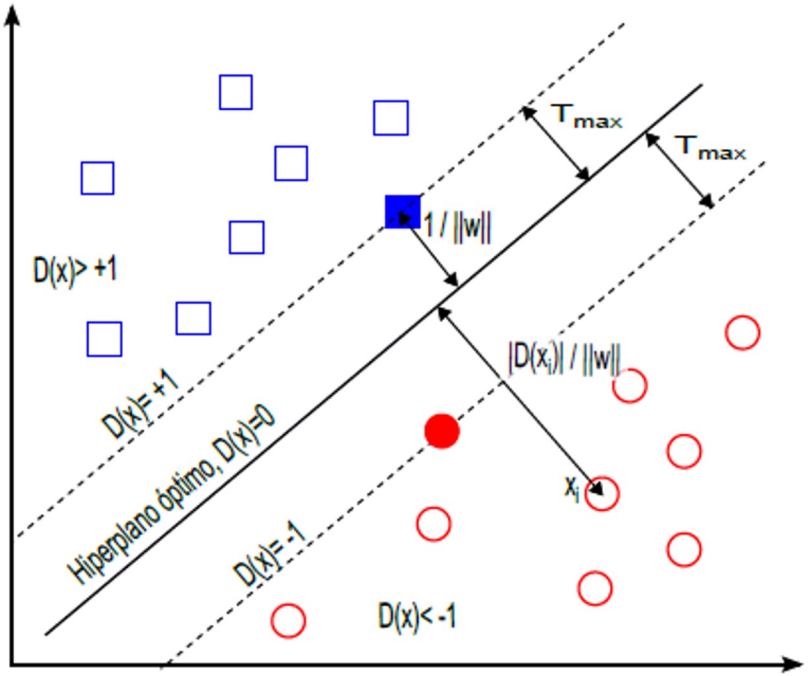
$$\frac{y_i D(x_i)}{\| w \|} \geq \tau, \quad i = 1, ..., n$$

Y haciendo que la norma de w, $\| w \| = 1$ podemos tener un hiperplano único.

Se busca encontrar los w y b (parámetros del modelo) que maximice T_{max}.

Pero el hiperplano óptimo será cuando llwll sea mínimo.

¿Se ve?



SVM para ejemplos cuasi-linealmente separables.

SVM

La situación planteada aquí es mucho más realista.

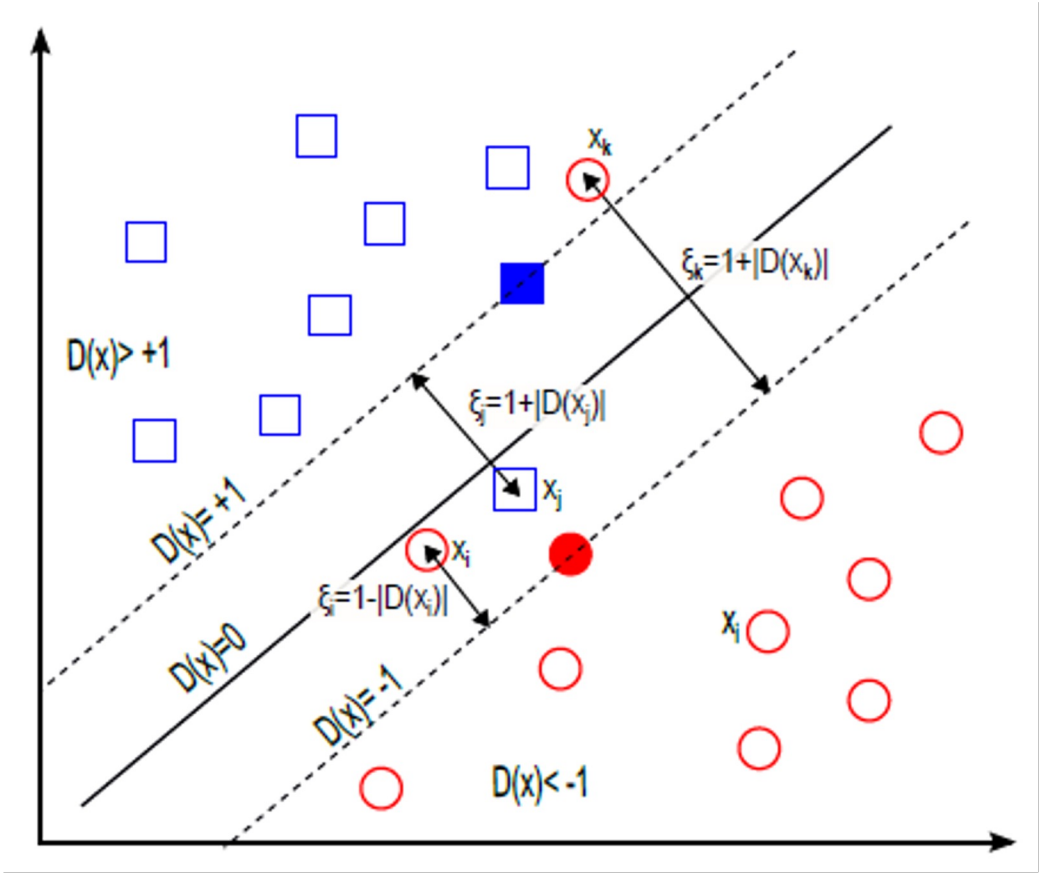
No se puede suponer que en la vida real un problema de clasificación sea linealmente separable.

Aquí, se relaja el supuesto de que no existen errores y efectivamente se permitirá que existan errores en el grupo de entrenamiento.

Sin embargo, el objetivo sigue siendo encontrar el hiperplano separador óptimo al resto de los ejemplos.

SVM

La situación de este caso se puede observar mediante la siguiente figura:



SVM

La modificación que se incluye para este caso es la de incluir un conjunto de variables de holgura a la ecuación del hiperplano, que cuantificarán el número de ejemplos no separables que se estará dispuesto a admitir:

$$y_i(< w, x_i > +b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, ..., n$$

De esta manera, la variable de holgura ξ valdrá 0 para los casos separables, mayores que 0 para los casos no separables y mayores que 1 a los casos no separables y mal clasificados.

En otras palabras, ahora se permiten clasificar ejemplos de manera incorrecta, pero se castigan con un valor. Generalmente, este castigo tiene la variable C.

Este hiperplano se conoce como: *hiperplano de separación de margen blando (soft margen).* El proceso de optimización es similar al caso anterior.

SVM para ejemplos
no separables linealmente.

SVM

Hasta el momento solo se ha asumido trabajar en el espacio original.

Ya que en el espacio original el problema puede ser muy difícil separar las clases, existe la posibilidad de que en un nuevo espacio sea más sencillo.

Aquí, utilizaremos funciones no lineales, para transformar los ejemplos del espacio original al llamado espacio de características

Este espacio, será mayor o mucho mayor que el espacio original.

SVM

Sea $\Phi : \mathbb{X} \longrightarrow F$ la función de transformación que hace corresponder cada ejemplo del espacio original al espacio de características.

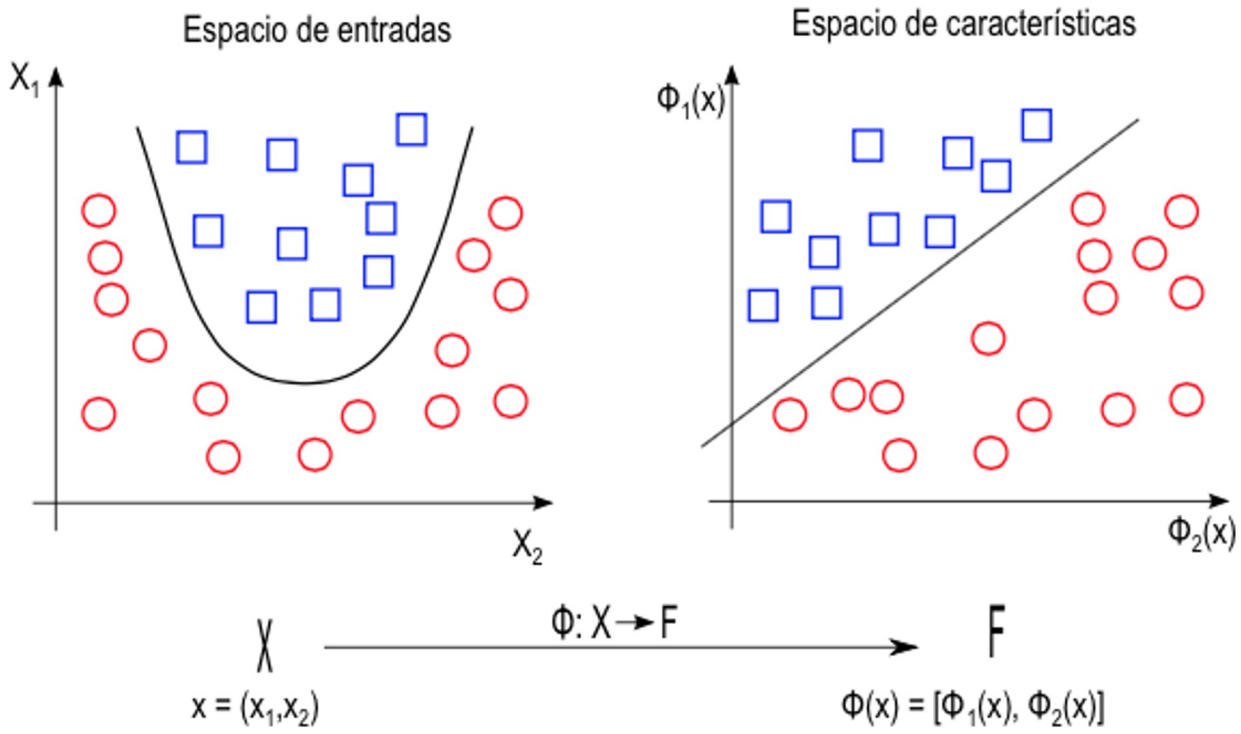
Entonces, sea $\Phi(x) = [\phi_1(x), ..., \phi_n(x)]$ una función no lineal que permitirá encontrar un hiperplano separador lineal en el nuevo espacio, pero qué será no-lineal en el espacio original.

Entonces, la función de decisión estará dada por:

$$D(x) = (w_1\phi_1(x) + ... + w_n) =< w, \phi(x) >$$

SVM

Visualización de la transformación del espacio original al espacio de características:



SVM

Pero en este caso, se trabaja en su forma dual: $D(x) = \sum_{i=1}^{n} \alpha_i^* y_i K(x, x_i)$

donde $K(x, x^*)$ se denomina función de kernel y α^* son coeficientes reales que nacen de aplicar multiplicadores de Lagrange.

No nos meteremos aqu

Una función de Kernel es una función $K = \mathbb{X} \times \mathbb{X} \longrightarrow \mathbb{R}$ que asigna a cada par de ejemplos del espacio de entrada un valor correspondiente al producto escalar de las imágenes de dichos ejemplos en un nuevo espacio.

$$K(x, x^*) =< \Phi(x), \Phi(x^*) >= (\phi_1(x)\phi_1(x^*) + ... + \phi_n(x)\phi_n(x^*))$$

SVM

Algunos ejemplos de funciones de kernel son:

- Kernel Lineal: $K(x, x^*) =< x, x^* >$
- Kernel polinomial (grado p): $K(x, x^*) = [\gamma < x, x^* > +\gamma]^p$
- Kernel Gaussiano: $K(x, x^*) = exp(-\gamma \parallel x - x^* \parallel^2), \quad \gamma > 0$

SVM

Aquí pueden encontrar un tutorial en español que profundiza la explicación vista en esta presentación.

Enrique Carmona. Tutorial sobre Máquinas de Vectores Soporte. 2013 (Muy buen material en español).

En scikit-learn, podremos usar este algoritmo usando la siguiente función:

sklearn.svm.SVC

sklearn.svm tiene varias versiones del algoritmo. Se recomienda revisar la documentación.