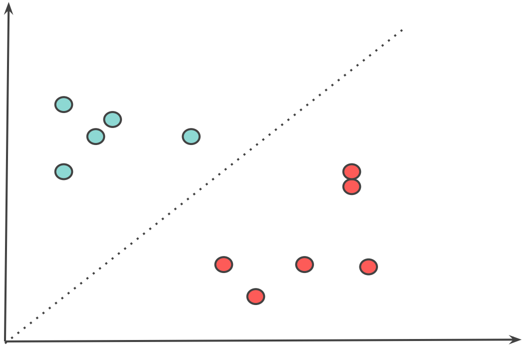
Support Vector Machine (SVM)

SVM es uno de los modelos supervisados más populares y más versátiles de machine learning Es usado tanto para clasificación como para regresión. Pero trataremos en la teoría el caso de clasificación.

¨El objetivo principal de SVM es encontrar el hiperplano óptimo que separa linealmente los puntos de datos en dos componentes maximizando el margen.¨

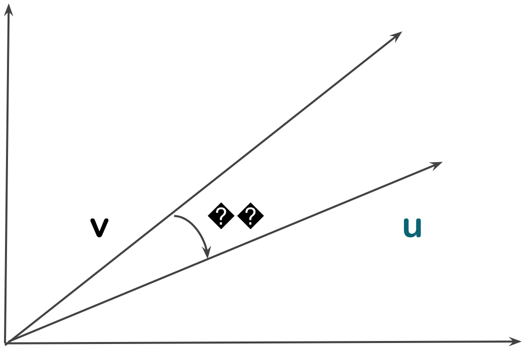


La línea punteada es el hiperplano que separa las dos diferentes clases.

Support Vector Machine (SVM)

Repaso álgebra lineal:

- Los vectores son cantidades matemáticas que tienen magnitud y dirección.
- La longitud de los vectores también se denomina norma. Indica la distancia de los vectores al origen.



Longitud: $||x|| = \sqrt{x_1^2+x_2^2+x_3^2}$

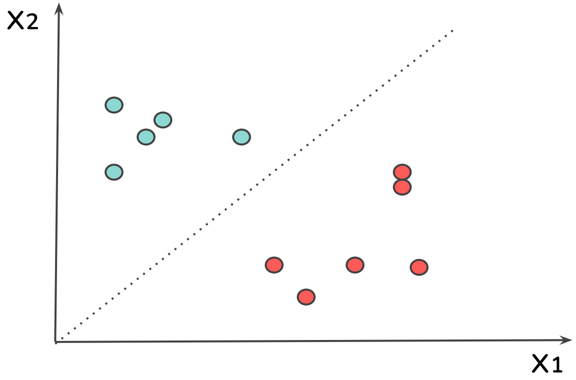
Direccion: $= \{ x_1/||x|| , x_2/||x||, x_3/||x|| \}$

$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}|| \ ||\mathbf{v}|| \, \mathrm{Cos}(\theta)$

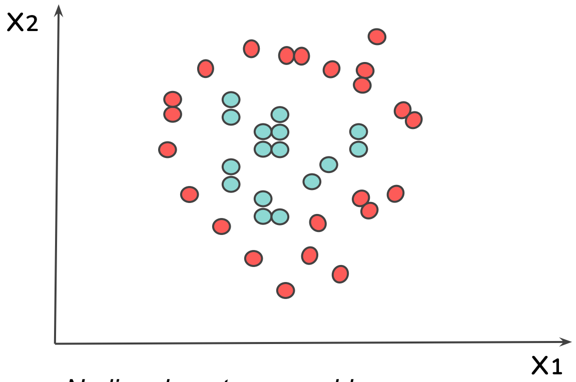
Support Vector Machine (SVM)

Hiperplano:

- Es un plano que divide linealmente los puntos de datos n-dimensionales en dos componentes. En el caso de 2D, el hiperplano es una línea, en el caso de 3D es un plano.



Linealmente separable



No linealmente separable

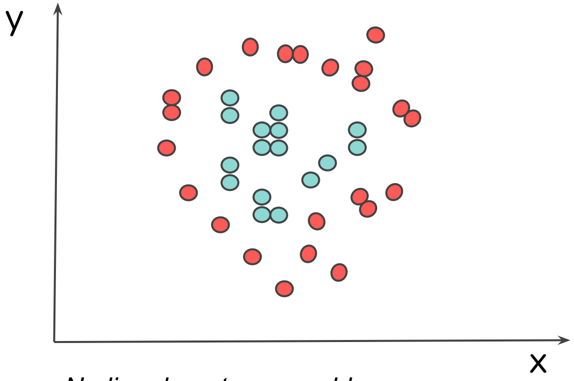
Si podemos separar los datos mediante un hiperplano podemos obtener la ecuación que los describe como:

W.X+b= 0 definiendo los vectores X = (x,y) W= (a,-1)

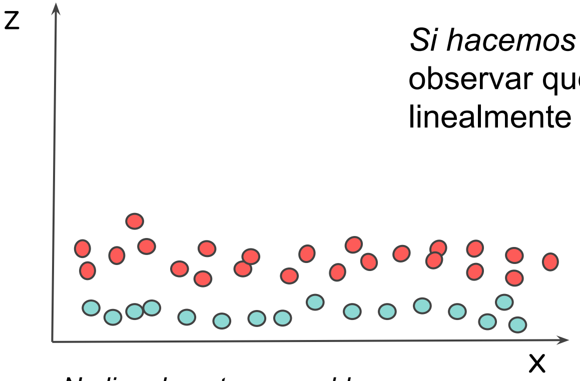
Support Vector Machine (SVM)

¿ Qué ocurre si los puntos no son linealmente separables ?:

- Esto es una situación que se da muy a menudo en problemas de aprendizaje automático ya que los datos que utilizamos normalmente son siempre linealmente separables.
- Podemos solucionarlo añadiendo una dimensión extra a los puntos de datos para que sean separables.



No linealmente separable



No linealmente separable

Si hacemos $z = x^2+y^2$ podemos observar que los datos son linealmente separables.

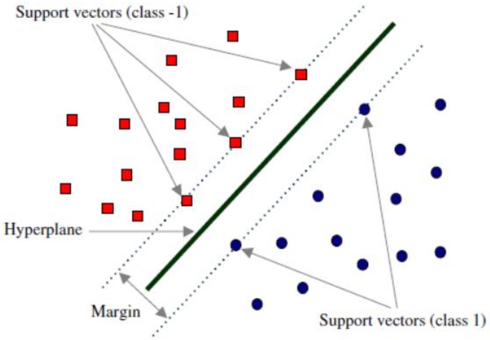
El proceso anterior de convertir un conjunto de datos no separable linealmente en uno separable linealmente también se conoce como Kernel Trick.

Support Vector Machine (SVM)

Hiperplano optimo

Pero definir un hiper plano no es fácil ya que hay muchos hiperplanos que pueden separar los puntos de datos en dos componentes. Así que el hiperplano óptimo es uno que divide los puntos de datos muy bien. Así que la pregunta es ¿por qué es necesario elegir el hiperplano óptimo?

Si se escoge un hiperplano sub-óptimo, no hay duda de que después de un número de iteraciones de entrenamiento, el error de entrenamiento disminuirá, pero durante la prueba, cuando una instancia no vista venga, resultará en un alto error de testeo. En ese caso, es necesario elegir un plano óptimo para obtener una buena precisión.



Support Vector Machine (SVM)

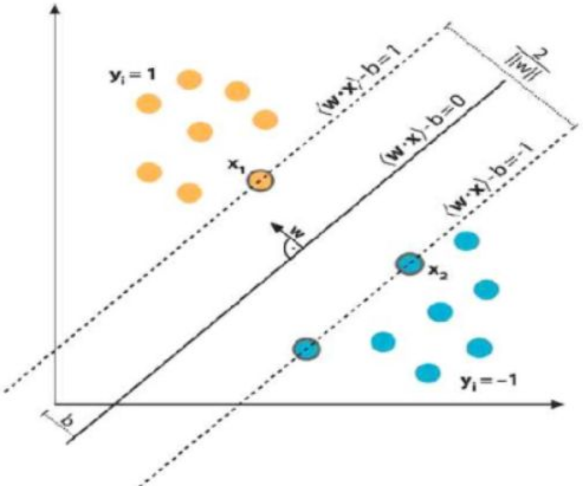
Margen (margin) y Support Vectors

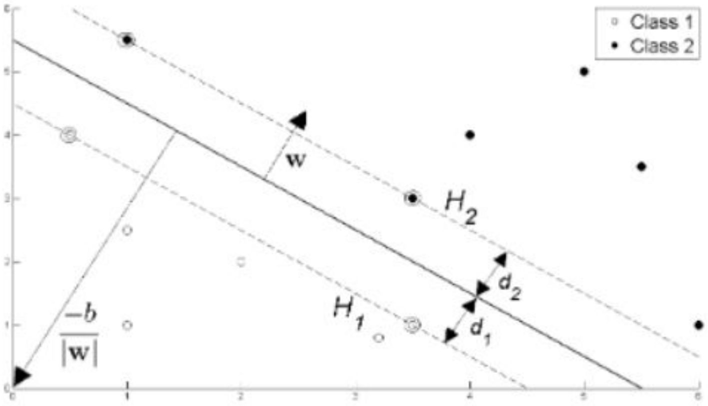
Supongamos que la línea negra sólida de la figura anterior es el hiperplano óptimo y que las dos líneas punteadas son algunos hiperplanos que pasan por los puntos de datos más cercanos al hiperplano óptimo. Entonces la distancia entre el hiperplano y el hiperplano óptimo se conoce como margen, y los puntos de datos más cercanos se conocen como vectores de soporte. El margen es un área que no contiene ningún punto de datos. Habrá algunos casos en los que tengamos puntos de datos en el área del margen, pero en este momento dejemos eso de lado.

Por lo tanto, al elegir el hiperplano óptimo, seleccionaremos uno entre el conjunto de hiperplanos que esté a la mayor distancia de los puntos de datos más cercanos. Si el hiperplano óptimo está muy cerca de los puntos de datos, entonces el margen será muy pequeño y generalizará bien para los datos de entrenamiento, pero cuando lleguen datos no vistos, fallará en generalizar bien como se explicó anteriormente. Así que nuestro objetivo es maximizar el margen para que nuestro clasificador sea capaz de generalizar bien para los casos no vistos.

Por lo tanto, en SVM nuestro objetivo es elegir un hiperplano óptimo que maximice el margen.

Interpretación matemática del hiperplano óptimo





Los ejemplos de los vectores de apoyo son los más cercanos al hiperplano óptimo y el objetivo de la SVM es orientar este hiperplano lo más lejos posible del miembro más cercano de las dos clases.

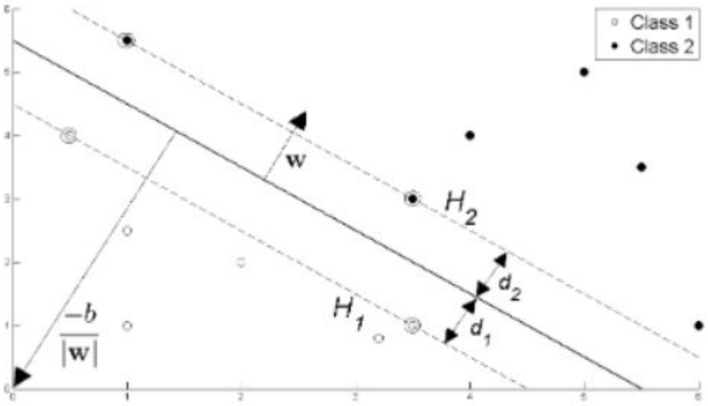
Support Vector Machine (SVM)

El problema SVM se puede formular como:

$w.x_i+b \geq 1$ para $y_i=1$
$w.x_i+b \leq -1$ para $y_i=-1$

combinando estas ec.
$y_i\ (w.x_i+b\ )-1 \geq 0$ para $y_i=1,-1$

De la fig de la derecha tenemos dos hiperplanos H1 y H2 que pasan por los vectores soporte de la clase +1 y -1 respectivamente.



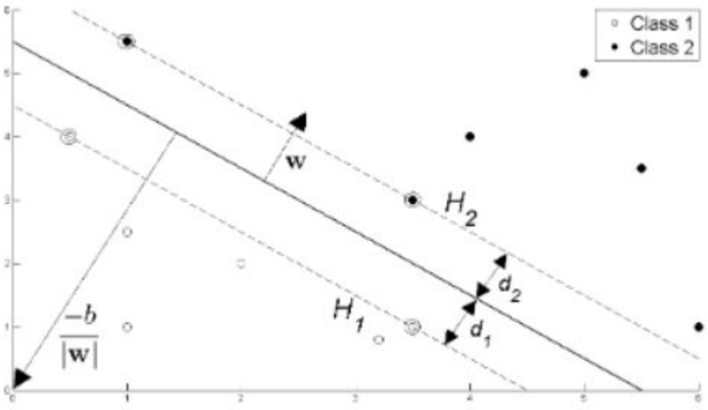
$w.x+b = -1$ para H1
$w.x+b = 1$ para H2

Support Vector Machine (SVM)

El problema SVM se puede formular como:

La distancia entre el hiperplano H1 y el origen es (-1-b)/|w| y la distancia entre el hiperplano H2 y el origen es (1-b)/|w|. Por lo tanto, el margen puede ser dado como

M = (1-b)/|w| - (-1-b)/|w|
M = 2/|w|



Donde M no es más que el doble del margen. Así que el margen se puede escribir como 1/|w|. Como el hiperplano óptimo maximiza el margen, el objetivo de la SVM se reduce a maximizar el término 1/|w|,

max 1/|w|, que puede escribirse como min |w|.

Support Vector Machine (SVM)

El problema a optimizar:

La meta es obtener el valor óptimo valor para w y b tal de poder minimizar ||**w**||, para ellos podemos utilizar los multiplicadores de lagrange.

$l = ||w||^2/2 - \sum \lambda_i (y_i(w.x_i+b)-1)$ para i=1,..,n la cantidad de puntos **(1)**

$\partial l/\partial w = w - \sum \lambda_i y_i.x_i=0$ **(2)** $w = \sum \lambda_i y_i.x_i$

$\partial l/\partial \lambda = \sum y_i(w.x_i+b)-1=0$ **(3)** $b=(1-\sum y_i w.x_i)/\sum y_i = (1-\sum y_i w.x_i)/\sum y_i$

$\partial l/\partial b = \sum \lambda_i y_i = 0$ **(4)**

A partir de la formulación anterior, sólo podemos encontrar el valor óptimo de w y eso depende de λ, por lo que necesitamos encontrar el valor óptimo de λ también. Y para encontrar el valor óptimo de b necesitamos tanto w como λ. Así que encontrar el valor de λ será lo más importante para nosotros.

Entonces, ¿cómo encontramos el valor de λ?

Support Vector Machine (SVM)

El problema a optimizar:

La meta es obtener el valor óptimo valor para w y b tal de poder minimizar ||**w**||, para ellos podemos utilizar los multiplicadores de lagrange.

$l = ||w||^2/2 - \sum \lambda_i \,(y_i(w.x_i+b)-1)$ para i=1,..,n la cantidad de puntos **(1)**

$l = ||\sum \lambda_\beta\, y_\beta.x_\beta||^2/2 - \sum \lambda_i\, y_i(\sum \lambda_\beta\, y_\beta.x_\beta.x_i+b)+ \sum \lambda_i$

$l = ||\sum \lambda_\beta\, y_\beta.x_\beta||^2/2 - \sum\sum \lambda_i\, y_i\, \lambda_\beta\, y_\beta.x_\beta.x_i - b\sum \lambda_i\, y_i + \sum \lambda_i$

$l = \sum \lambda_i - \sum\sum \lambda_i\, y_i\, \lambda_\beta\, y_\beta.x_\beta.x_i/2$

Definiendo $K(i,\square)=\lambda_i\, y_i\, \lambda_\beta\, y_\beta.x_\beta.x_i$ en forma de vector $K=y^t\, y\, .\, x^t\, x$

max: $l = \sum \lambda_i - \lambda^t\, \lambda\, K\, /2$

Support Vector Machine (SVM)

El problema a optimizar:

Así que este es el problema de optimización final, para encontrar el valor máximo de λ.

max: $l = \sum \lambda_i - \lambda t \, \lambda \, K / 2$

¿Ahora cómo resolvemos el problema anterior?

La operación de maximización anterior puede ser resuelta con los algoritmos SMO (optimización de minimización secuencial). También hay el apoyo de varias bibliotecas en línea para esta optimización. Una vez que obtenemos el valor de λ podemos obtener w y por último el valor de b.

Como ahora tenemos el valor tanto para w como para b , entonces el hiperplano óptimo que puede separar los puntos de datos puede ser escrito como,

$$\mathbf{w.x+b=0}$$

SVM: Ventajas

- *SVM sirve tanto para clasificación como para regresión.*
- *SVM es eficaz en espacios de alta dimensión.*
- *SVM funciona relativamente bien cuando hay un claro margen de separación entre las clases.*
- *SVM es eficaz en los casos en que el número de dimensiones es mayor que el número de muestras.*
- *Puede gestionar datos no lineales mediante el truco del núcleo*
- *SVM es relativamente eficiente en cuanto a la memoria.*

SVM : Desventajas

- *SVM no es adecuado para grandes conjuntos de datos.*
- *SVM no funciona muy bien cuando el conjunto de datos tiene más ruido, es decir, cuando las clases objetivo se solapan.*
- *Sensible a los valores atípicos*
- *Alta complejidad algorítmica*
- *El entrenamiento requiere mucho tiempo.*

Bibliografía

- [Sklearn SVM]
- [Kernel Trick]