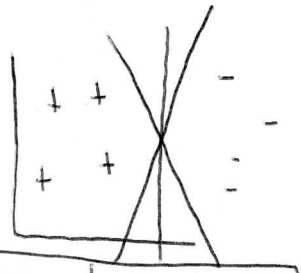
Support Vector Machines

Felipe Bravo-Marquez

- Aprendizaje Estadístico
- Funciona bien con alta dimensionalidad (evita maldición de la dimensionalidad)

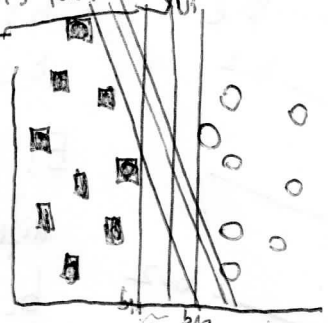
- Hiperplanos de Máximo margen o. Hiperplanos separadores



- Idea: Datos a un lador del Hiperplano son clasificados de una clase y al otro lado de la otra.

Hipótesis: Los datos son linealmente Separables

¿Cuál es el "mejor" hiperplano? Es el que maximiza la distancia (B1) entre ambos conjuntos separador



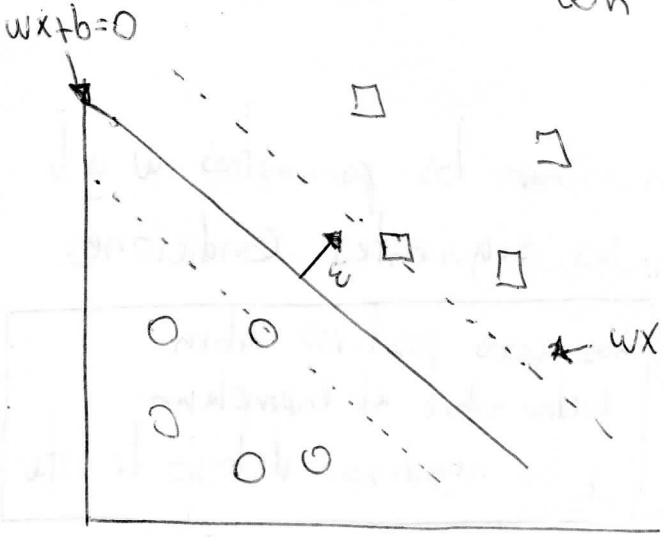
- Entre mayor sea el margen, mejor poder de generalización

- SVM Lineales

- Se tiene un problema de clasificación binaria con N ejemplos de entrenamiento.
- Cada ejemplo se denota por la tupla (x_i, y_i) $i \in [1, 2, \dots N)$
  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$. Luego $y_i \in \{-1, 1\}$

- El limite de decisión del clasificador:

$$w \cdot x + b = 0$$ con w y b como parametros del modelo



Cualquier punto x_a, x_b ubicado en el límite de decisión cumple

$$w \cdot x_a + b = 0$$
$$w \cdot x_b + b = 0 \Rightarrow w \cdot (x_b - x_a) = 0$$

Como $x_b - x_a$ es paralelo al límite de decisión, sabemos que w es perpendicular al hiperplano

w es el vector perpendicular al plano!

Para cualquier punto X_s que esté sobre el límite de decisión se cumple que:

$$w \cdot X_s + b = k \quad \text{con } k > 0$$

Para cada cisne X_c se cumple que

$$w \cdot X_c + b = k' \quad \text{con } k' < 0$$

Un ejemplo de test z se puede clasificar como

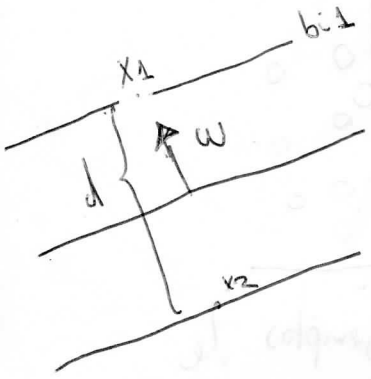
$$y = \begin{cases} 1 & \text{si } w \cdot z + b > 0 \\ -1 & \text{si } w \cdot z + b < 0 \end{cases}$$

Margen de un Clasificador Lineal

Como w y b son parámetros del modelo podemos re-escalarlos de forma que:

$b_{i1}: w \cdot X + b = 1$ *

$b_{i2}: w \cdot X + b = -1$ **



El margen del límite de decisión es la diferencia entre los dos hiperplanos paralelos b_{i1} y b_{i2}, Sea X_1 un punto en b_{i1} y X_2 un punto en b_{i2}

Entonces reemplazamos X_1 y X_2 en * y ** luego sustraemos ** a * nos queda

$$w \cdot (X_1 - X_2) = 2$$

Del álgebra lineal sabemos que $a \cdot b = \|a\| \|b\| \cos\theta$

Como $X_1 - X_2$ es paralelo a w, $\theta = 0 \Rightarrow \cos(\theta) = 1$

$$\Rightarrow \|w\| \times d = 2 \quad \Rightarrow \quad d = \frac{2}{\|w\|}$$

Entrenando una SVM Lineal

El proceso de entrenamiento consiste en encontrar los parámetros w y b que maximicen el margen y se cumplan las siguientes condiciones

$$w \cdot x_i + b \geq 1 \quad \text{si } y_i = 1$$

$$w \cdot x_i + b \leq -1 \quad \text{si } y_i = -1$$

Los casos positivos deben quedar sobre el hiperplano y los negativos debajo de éste

$w^T w = \|w\|^2$

Las últimas restricciones pueden resumirse como:

$$y_i(w \cdot x_i + b) \geq 1 \quad , \quad i = 1, 2, \ldots, N$$

SVM impone que el margen de decisión sea máximo

Maximizar el margen, es equivalente a minimizar $f(w) = \frac{\|w\|^2}{2}$

Formalizamos el problema de optimización como:

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

La restricción me dice que todos los datos estén bien clasificados

sujeto a $y_i(w \cdot x_i + b) \geq 1, \quad i = 1, 2, \ldots, N$

El problema de optimización es convexo y de restricciones lineales puede resolverse con optimización cuadrática

- Se resuelve el problema usando multiplicadores de Lagrange

Queremos minimizar esta nueva función

$$L_p = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda_i \left(y_i(w \cdot x_i + b) - 1 \right)$$

- Los parámetros λ_i se conocen como multiplicadores de Lagrange
- Si no le pusieramos restricciones al problema, el optimo sería $w = 0$ que viola las restricciones del problema. ✓ infactibles
- Las soluciones para w y b son no-satisfaibles si violan alguna de las restricciones ej: $y_i(w \cdot x_i + b) - 1 < 0$
- Asumiendo que $\lambda_i \geq 0$ cualquier solución no valida sólo "incrementa" el valor del Lagrangeano.

Para minimizar el Lagrangiano tenemos que derivar L_p respecto a w y b y setearlas a cero

$\frac{\partial L_P}{\partial w} = 0 \Rightarrow$ w es un vector realmente hacemos $\frac{\partial L_P}{\partial w_i}$ para todo i

Mantenemos la notación vectorial para simplificar los cálculos.

$$\frac{\partial L_P}{\partial w} = w - \sum_{i=1}^{N} \lambda_i y_i x_i \Rightarrow w = \sum_{i=1}^{N} \lambda_i y_i x_i \quad (1)$$

w es una combinación lineal de los vectores x_i

$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i y_i = 0 \quad (2)$$

Como los multiplicadores de Lagrange son desconocidos aún no podemos encontrar los w^*, b^* óptimos

Si las restricciones fueran de igualdad en vez de desigualdad podríamos resolver w^*, b^*, λ_i^* óptimos haciendo $\frac{\partial L_P}{\partial w} = 0$, $\frac{\partial L_P}{\partial b} = 0$, $\frac{\partial L_P}{\partial \lambda_i} = 0$

Notamos que λ_i son valores libres que pueden tomar cualquier valor.

Para resolver problemas de optimización con restricciones de desigualdad se transforma en un problema con restricciones de igualdad y se verifican las condiciones Karush-Kuhn-Tucker

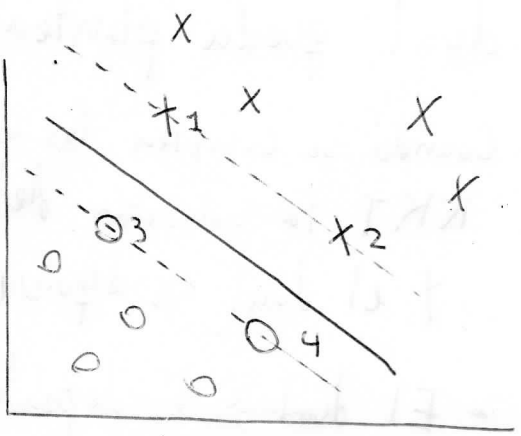
$$\lambda_i \geq 0 \quad (3)$$

KKT son las condiciones necesarias y suficientes para que la solución de un problema no lineal sea óptima

$$\lambda_i [y_i (w \cdot x_i + b) - 1] = 0 \quad (4)$$ "Regla de holgura complementaria"

- Pareciera que se tienen tantos multiplicadores de Lagrange como instancias de entrenamiento, pero la restricción 4 hace que todos los multiplicadores valgan 0 a menos que x_i satifaga la ecuación $y_i(w \cdot x_i + b) = 1$. Esos x_i donde $\lambda_i > 0$ caen en

los hiperplanos b_{i1} o b_{i2} y se conocen como "vectores de soporte". Los datos que no caen en esos hiperplanos tendrán $\lambda_i = 0$



1, 2, 3 y 4 son los vectores de soporte.

Recordemos que

$$w = \sum_{i=1}^{N} \lambda_i y_i x_i \quad (1)$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0 \quad (2)$$

} Dependan sólo de los vectores de soporte.

El problema de optimización sigue siendo muy complejo, pues requiere encontrar el valor óptimo de muchos parámetros: w, b y λ_i.

El Lagrangiano L_P se transforma en el dual L_D que depende solamente de λ_i. El dual es un problema de maximización en vez de un problema de minimización como L_P (primal). Tenemos que sustituir la ecuación (1) y (2) en L_P.

$$\Rightarrow L_P = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i \left(y_i (w \cdot x_i + b) - 1 \right)$$

$$= \frac{1}{2} \left(\sum_i \lambda_i y_i x_i \right)^T \left(\sum_j \lambda_j y_j x_j \right) - \sum_{i=1}^{N} \lambda_i \left(y_i \left(\sum_j \lambda_j y_j x_j \right) x_i + b \right) - 1$$

Reordenando.

$$= \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j - b \sum_{i=1}^{N} \lambda_i y_i$$

$\sum \lambda_i y_i = 0$ por 2

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j$$

El problema de optimización dual queda planteado como

$$\underset{\lambda_i}{\text{Max}} \; L_D$$

Sujeto a $\lambda_i \geq 0 \quad i = 1, \ldots, N$

$$\sum_{i=1}^{N} \lambda_i y_i = 0$$

- Cuando se cumplen las condiones KKT la solución del primal y el dual es equivalente
- El dual sólo depende de λ_i y los datos de entrenamiento

El problema dual se puede optimizar usando métodos numéricos de programación cuadrática.

- Una vez encontrados los valores óptimos del dual λ_i^*

Usamos $w^* = \sum_{i=1}^{N} \lambda_i y_i x_i$ y la regla de holgura complementaria $\lambda_i [y_i (w x_i + b) - 1] = 0$ para encontrar b^*

El límite de decisión queda expresado por:

$$\left(\sum_{i=1}^{N} \lambda_i y_i x_i \cdot x \right) + b = 0$$

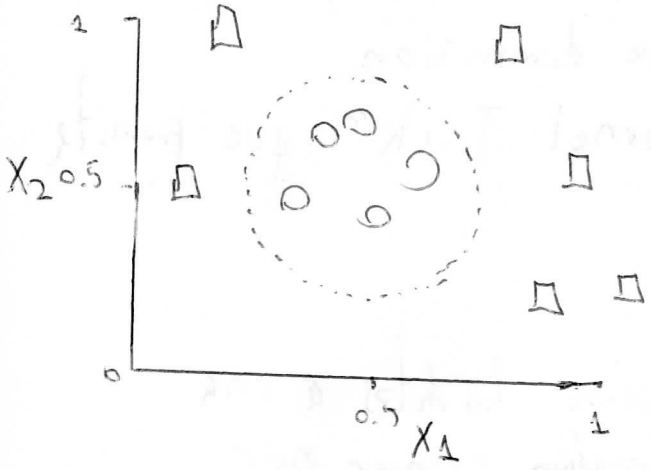
Entonces para clasificar un dato nuevo x basta con calcular su producto interno con todos los datos de entrenamiento que sean vectores de soporte.

- b se obtiene resolviendo la ecuación de holgura complementaria para los vectores de soporte.
- Como los λ_i se obtienen numericamente pueden haber errores numéricos y b no sea único
- Se promedian los distintos valores de b

Generalmente los vectores de soporte son pocos.

SVM no lineal

La formulación de SVM anterior encuentra un límite de decisión lineal entre las clases. Ahora veremos como aplicar SVM a datasets que tienen límites de decisión no lineales.



Sea este dataset donde los cuadrados ($y=1$) y los círculos ($y=-1$)

Los círculos se agrupan en el centro

Las instancias del dataset se clasifican mediante la siguiente ecuación

$$y(X_1, X_2) = \begin{cases} 1 & \text{si } \sqrt{(X_1-0.5)^2+(X_2-0.5)^2} > 0.2 \\ -1 & \sim \end{cases}$$

El límite de decisión quedaría como: $\sqrt{(X_1-0.5)^2+(X_2-0.5)^2} = 0.2 \iff X_1^2 - X_1 + X_2^2 - X_2 = -0.46$

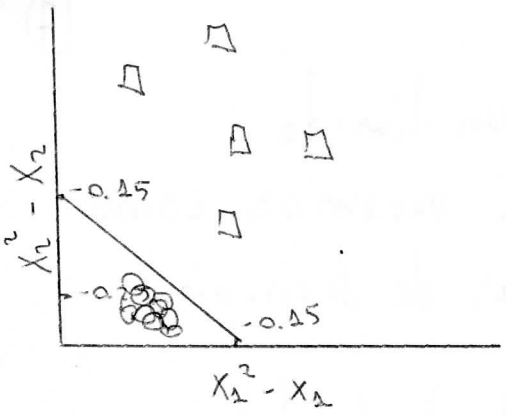
Necesitamos una transformación no-lineal ϕ para mapear los datos desde su espacio de atributos a un espacio de atributos superior donde el límite de decisión se vuelva lineal

Por ejemplo, supongamos la transformación:

$$\phi : (X_1, X_2) \rightarrow (X_1^2, X_2^2, \sqrt{2}X_1, \sqrt{2}X_2, 1)$$

En el espacio transformado podríamos encontrar los parámetros $W = (w_0, w_1, \dots, w_4)$ tal que:

$$w_4 X_1^2 + w_3 X_2^2 + w_2 \sqrt{2} X_1 + w_1 \sqrt{2} X_2 + w_0 = 0$$



Idea: En este nuevo espacio si es posible encontrar un límite de decisión lineal

- Un posible problema de este enfoque, es que podría sufrir de maldición de dimensionalidad al trabajar con datos de alta dimensión.
- Veremos un truco llamado "Kernel Trick" que permite evitar este problema.

Entrenando una SVM no lineal

Necesitamos encontrar funciones ϕ que mapeen los datos a una dimensión superior para que los datos se puedan separar por un hiperplano.

- La tranformación puede ser costosa
- El nuevo límite de decisión tendrá la forma $w \cdot \phi(x) + b = 0$

El dual de la SVM no lineal quedaría como:

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \phi(x_i) \cdot \phi(x_j)$$

Una vez que encontramos los λ_i's óptimos usando técnicas de programación cuadrática w^*, b^* podrían ser calculados usando:

$$w = \sum_i \lambda_i y_i \phi(x_i)$$

$$\lambda_i \left(y_i \left(\sum_j \lambda_j y_j \phi(x_j) \cdot \phi(x_i) + b \right) - 1 \right) = 0$$

Una nueva instancia z puede clasificarse usando la ecuación.

$$f(z) = \text{sign}(w \cdot \phi(z) + b) = \text{sign}\left(\sum_{i=1}^{n} \lambda_i y_i \phi(x_i) \cdot \phi(z) + b\right)$$

Tanto el valor de b^* como $f(z)$ depende de productos internos (similitud) de pares de vectores en el espacio transformado. Esto puede ser costoso y puede sufrir de maldición de la dimensionalidad. La idea es reemplazar $\phi(x_i) \cdot \phi(x_j)$ por una función Kernel $K(x_i, x_j)$ que pueda calcular el producto punto de ambos vectores transformados sin necesidad de calcular la transformación. Esto se llama "Kernel Trick"

Kernel Trick

El producto punto se puede relacionar con una medida de similitud entre dos vectores (pensar en la similitud coseno)

Entonces $\phi(\vec{x_i}) \cdot \phi(\vec{x_j})$ puede entenderse como una similitud en el espacio transformado

- El Kernel trick permite computar el producto punto en el espacio transformado haciendo el cálculo en el espacio Original

Ejemplo:

Sea $\phi(\vec{x_i}) = (x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, 1)$ con $x_i \in \mathbb{R}^2$

$$\phi(\vec{x_i}) \cdot (\vec{x_j}) = (x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, 1) \cdot (x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}, \sqrt{2}x_{j2}, 1)$$

$$= x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 1$$

$$= (x_i \cdot x_j + 1)^2$$

Podríamos reemplazar todos los $\phi(x_i) \cdot \phi(x_j)$ por $K(x_i, x_j)$ con $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) = (x_i \cdot x_j + 1)^2$ (10)

K es una función de Kernel que es mucho más barata de calcular que $\phi(x_i) \cdot \phi(x_j)$. Entonces si reemplazamos los productos puntos de vectores por funciones de Kernel estaremos trabajando "implícitamente" en un espacio de mayor dimensión

Ejemplos de Kernel

$K(x, y) = (x \cdot y + 1)^p$ Kernel polinomial

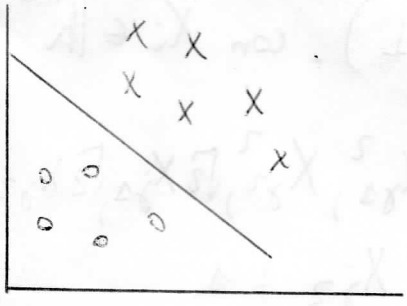
$K(x, y) = e^{-\|x-y\|^2/(2\sigma^2)}$ Gaussiano o RBF (En este Kernel ϕ tiene dimensión infinita)

$K(x, y) = \tanh(kx \cdot y - \delta)$

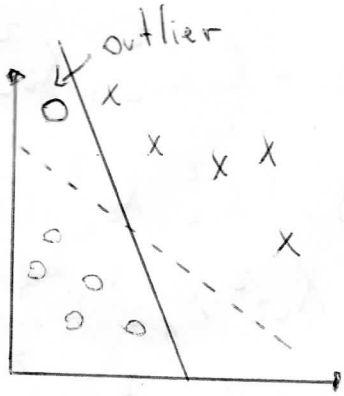
Una función de Kernel deben ser funciones definidas positivas.

SVM de Margen Suave

Sea el siguiente caso:



Ahora le agregamos un outlier



Un outlier puede reducir mucho el tamaño del margen

Le agregamos variables de holgura al problema para permitirle al hiperplano que clasifique mal algunos puntos. A esto se le llama "Soft margin".

El problema primal queda como

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$$

sujeto a $y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad i = 1, \ldots, N$

$$\xi_i \geq 0, \quad i = 1, \ldots, N.$$

Entonces si nos equivocamos con un dato de entrenamiento $\xi > 0$ y pagaremos un costo $C \cdot \xi$.

Entonces C es un parámetro del modelo

Para entrenar una SVM se hace una búsqueda de grill (grid-search) sobre valores de C y algún parámetro del Kernel. γ para Gaussiano, p para polinomial
↑ gamma

- Generalmente se usan valores de potencia de dos ej: $C \in [2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3]$
- Se usa validación cruzada y uno se queda con los parámetros de menor error.