Máquinas de vectores soporte (SVM)

UCO ONLINE

Máquinas de vectores soporte

1. Introducción

Las máquinas de soporte vectorial (más conocidas por sus siglas en inglés: SVM, *Support Vector Machine*), son métodos para aprendizaje supervisado basados en teoría de aprendizaje estadística. Estos métodos, por lo general, tratan de escoger una función de entre un conjunto de funciones, de modo que minimice un cierto riesgo; de este modo, un algoritmo de aprendizaje debe buscar el mejor conjunto de funciones (determinado por su complejidad), y la mejor función de dicho conjunto.

Supongamos en primer lugar, que el conjunto de datos de entrenamiento es separable por un hiperplano (línea en 2D, plano en 3D). La complejidad de un hiperplano puede medirse por el margen. El margen se define como la mínima distancia entre un ejemplo y la superficie o hiperplano de decisión. Por lo que, conociendo o limitando el margen de una clase de función por abajo, podemos controlar su complejidad.

El aprendizaje basado en vectores soporte se basa en que *el riesgo se minimiza cuando el margen se maximiza*. SVM escoge un hiperplano con máximo margen (*máximum-margin hyperplane*) en un espacio de entrada transformado, de forma que divida los patrones de entrenamiento a la vez que maximiza la distancia a los patrones más cercanos a dicha superficie de decisión. Los parámetros del hiperplano solución se derivan de un problema de optimización de programación cuadrática.

Consideremos como ejemplo un problema de clasificación simple donde los patrones son separables, en un espacio multi-dimensional. Dados los patrones de dos clases, cualquier hiperplano de clasificación razonable pasaría entra ambos grupos de patrones, separando las clases. Un posible hiperplano es aquel que asigna un nuevo punto a la clase cuya media/centroide esté más cercana a él. Esta frontera de decisión es equivalente a calcular la clase de un nuevo punto como el ángulo entre dos vectores: el vector que conecta los centroides de ambos grupos de patrones, y el vector que conecta el punto medio de dicha línea con el nuevo punto. Este ángulo se puede formular por medio de un producto escalar entre vectores. Por tanto, la superficie de decisión se define en términos de similaridad entre cada nuevo punto y el centroide del grupo – hablamos de una función *kernel*. Este clasificador simple es lineal en el espacio de características, mientras que en el dominio de entrada se representa como una función *kernel* en términos de los ejemplos de entrenamiento. Como se observará más adelante, los *kernels* no suelen tener en cuenta todos los patrones, sino que los ejemplos tendrán pesos para la generación de la frontera de decisión final. Por ejemplo, se pretende eliminar la influencia de patrones que están lejos de la frontera de decisión, ya que no va a influir en el error de generalización del método, de modo que también se reduzca la complejidad del modelo. Por tanto, la frontera de decisión dependerá únicamente de un subconjunto de patrones, llamados vectores soporte (*support vectors*).

2. Clasificación basada en hiperplanos

La tarea más básica de clasificación es buscar una regla, basada en observaciones externas, que asigne un objeto a una de varias clases. En el caso más simple, solo existen dos clases, por lo que podemos formular el problema como la estimación de una función $f: \mathbb{R}^N \rightarrow \{-1, +1\}$, utilizando las entradas y salidas de patrones de entrenamiento, y suponiendo que los valores de salida para ambas clases son -1 (clase negativa) y +1 (clase positiva). Nótese que en otras formulaciones del problema podíamos utilizar 0 y 1 como valores de salida, mientras que en este caso, por conveniencia, utilizamos -1 y 1.

La mejor función f es aquella que minimiza el error (o riesgo) esperado, que sería la integral de una cierta función de error l de acuerdo a la distribución de probabilidad desconocida de los datos $P(\mathbf{x}, y)$. Para problemas de clasificación, l se suele llamar *0/1 loss*: $l(f(\mathbf{x}), y) = \theta(-y\, f(\mathbf{x}))$, donde $\theta(z) = 0$ si $z < 0$, y $\theta(z) = 1$ en caso contrario.

Desafortunadamente, el riesgo no puede minimizarse directamente, ya que la distribución $P(\mathbf{x}, y)$ del problema concreto no se conoce. Por tanto, debemos tratar de estimar una función que se acerque a la óptima con la información disponible, es decir, los patrones de entrenamiento. La base de estos métodos es que, un clasificador simple o lineal, con una función que explique la mayoría de los datos, es preferible a una función compleja (principio de la *Navaja de Ockham*)

2.1. Clasificador lineal

Volvamos a asumir, de momento, que los patrones de entrenamiento son separables por un hiperplano, y escogemos funciones de la forma $(\mathbf{w} \cdot \mathbf{x}) + b = 0$ (donde $\mathbf{w}$ es un conjunto de pesos, todos ellos números reales; $\mathbf{x}$ son los valores para los distintos atributos de un patrón; y b es también un número real llamado *bias* por lo general) para la frontera de decisión. La función de decisión vendría dada por:

$$f(x) = signo\big((\mathbf{w} \cdot \mathbf{x}) + b\big).$$

Como se comentaba anteriormente, el margen sería la mínima distancia de un patrón a la frontera de decisión. Como asumimos que los patrones de entrenamiento son separables, podemos escalar $\mathbf{w}$ y b de tal manera que los puntos más cercanos a la frontera de decisión cumplan que $|(\mathbf{w} \cdot \mathbf{x}_i) + b| = 1$; a esto lo llamamos obtener la representación canónica del hiperplano. Ahora consideremos dos ejemplos $\mathbf{x}_1$ y $\mathbf{x}_2$, de diferentes clases, con $|(\mathbf{w} \cdot \mathbf{x}_1) + b| = 1$ y $|(\mathbf{w} \cdot \mathbf{x}_2) + b| = 1$, respectivamente. En este caso, el margen viene dado por la distancia de esos dos puntos, medidos

de forma perpendicular al hiperplano, es decir:

$$\left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot (\mathbf{x}_1 - \mathbf{x}_2)\right) = \frac{2}{\|\mathbf{w}\|}$$

De entre todos los hiperplanos que separen los datos, existe uno que obtiene un máximo margen de separación entre ambas clases:

$$\max_{\{\mathbf{w},b\}} \min \{\|\mathbf{x} - \mathbf{x}_i\|: \ \mathbf{x} \in \mathbb{R}^N, (\mathbf{w} \cdot \mathbf{x}) + b = 0, i = 1, \dots, n\}$$

Para construir dicho hiperplano óptimo, habrá que solucionar el siguiente problema de optimización:

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

$$\text{Sujeto a: } y_i \cdot \big((\mathbf{w} \cdot \mathbf{x}_i) + b\big) \geq 1, i = 1, \dots, n$$

Esta restricción en el problema de optimización puede resolverse utilizando los multiplicadores de Lagrange $\alpha_i \geq 0$ y la función de Lagrange:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \big(y_i \cdot \big((\mathbf{w} \cdot \mathbf{x}_i) + b\big) - 1\big)$$

Al minimizar esta función de Lagrange, obtendríamos que:

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \ ; \ \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Sustituyendo, obtenemos la optimización de los valores de α como:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\text{Sujeto a: } \alpha_i \geq 0 \ ; \ i = 1, \dots, n \ ; \ \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0$$

El hiperplano de decisión se puede especificar ahora como:

$$f(x) = signo\left(\sum_{i=1}^{n} \alpha_i y_i (\mathbf{x} \cdot \mathbf{x}_i) + b\right)$$

y b se calcula del conjunto de vectores soporte $\mathbf{x}_i, i \in I \equiv \{i: \alpha_i \neq 0\}$:

$$b = \frac{1}{|I|} \sum_{i \in I} \left(y_i - \sum_{j=1}^{n} \alpha_j y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \right)$$

Aquellos patrones que resulten tener un valor de α_i distinto de 0, son los llamados vectores soporte, ya que contribuyen a los pesos de la frontera de decisión (y sus correspondientes valores de α_i son llamados valores soporte). Por tanto, todos los patrones que no sean vectores soporte, son irrelevantes a la hora de obtener el hiperplano o clasificador solución; es decir, el hiperplano está definido únicamente por los patrones cercanos.

En la Figura 1 se muestra un ejemplo de problema linealmente separable, donde existen múltiples posibles separaciones entre ambas clases (imagen de la izquierda). Sin embargo, existe un único hiperplano (en este caso, una línea dado que son 2 dimensiones) que separa ambas clases con un margen máximo (imagen de la derecha). Además, en la imagen de la derecha podemos observar como los vectores soporte son los ejemplos con el menor margen o distancia al hiperplano (patrones con relleno sólido).

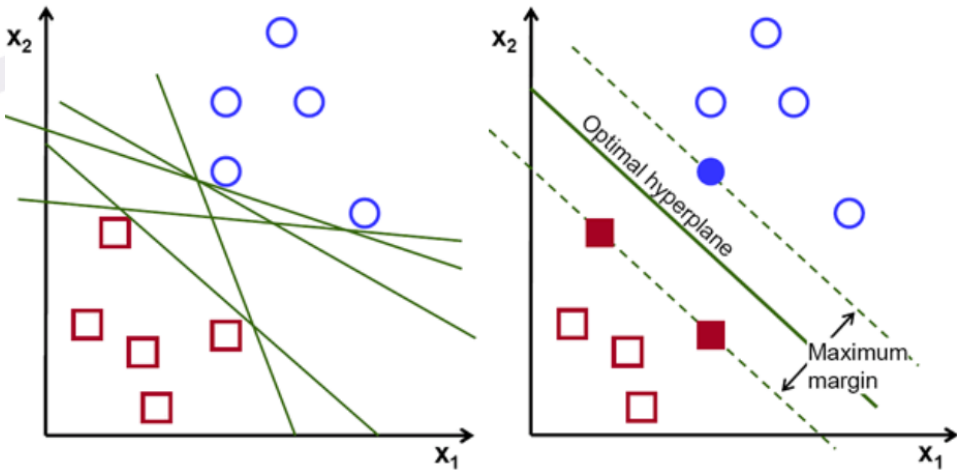

Figura 1. Posibles separaciones lineales entre dos clases (izquierda), y separación de máximo margen en SVM (derecha).[1]

[1] Fuente: https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c

2.2. El truco del kernel

La elección de funciones de clasificación lineal parece estar muy limitada, y es bastante probable que no aprenda correctamente la distribución de los datos. Afortunadamente, es posible tener tanto modelos lineales como un conjunto amplio de funciones de decisión no lineales gracias al truco del *kernel* con hiperplanos de margen máximo.

Usar *kernels* en SVM hace que el hiperplano de margen máximo se obtenga en un espacio de características F, que es una transformación no lineal del espacio de entrada original $\Phi: R^N \rightarrow F$. Por lo general, el nuevo espacio de características tiene una dimensionalidad mucho más alta que el espacio de entrada original. Con dicho truco del *kernel*, el mismo clasificador lineal se utiliza para aprender a partir de los datos transformados.

En la Figura 2 se observa un ejemplo de un problema con 2 dimensiones que gracias a una función de *kernel*, pasa a tener 3 dimensiones. Mientras que en la parte de la izquierda (original), los datos no son linealmente separables, en la parte derecha (transformado), es posible obtener un plano que separe los datos de una clase de la otra. Es decir, hemos conseguido una transformación de los datos que haga que sean linealmente separables.

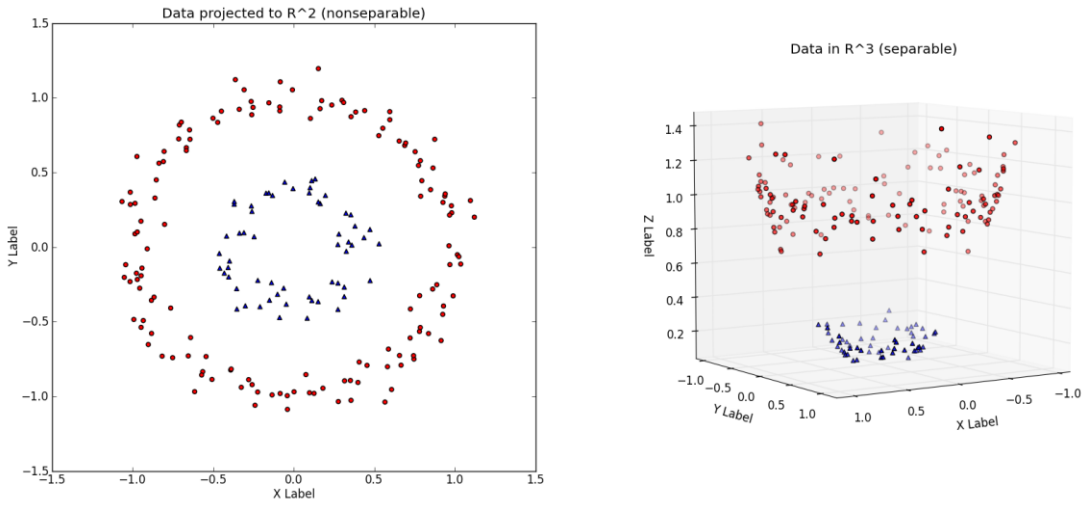

Figura 2. Efecto de utilizar un kernel que incremente la dimensionalidad de los datos originales.[2]

La elección de uno u otro *kernel* es ya un trabajo experimental, no existiendo uno que sea mejor que otro para todos los casos. Algunos *kernels* comunes se muestran en la Tabla 1. Cabe destacar que el *kernel* de funciones de base radial (RBF, *Radial Basis Function*) es uno de los más extendidos.

[2] Fuente: https://turing.iimas.unam.mx/~ivanvladimir/page/curso_aprendizaje_automatico_s6/

Tabla 1. Funciones de kernel utilizadas comunmente.

Kernel	$K(\mathbf{x}, \mathbf{x}_i)$
Funciones de base radial (RBF)	$\exp(-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2),\ \gamma > 0$
Multicuadrática inversa	$1/\sqrt{\|\mathbf{x} - \mathbf{x}_i\| + \eta}$
Polinómico de grado d	$\left((\mathbf{x}^T \cdot \mathbf{x}_i) + \eta\right)^d$
Sigmoidal	$\tanh(\gamma(\mathbf{x}^T \cdot \mathbf{x}_i) + \eta),\ \gamma > 0$
Lineal	$\mathbf{x}^T \cdot \mathbf{x}_i$

Utilizando *kernel*, en las ecuaciones descritas anteriormente habría que sustituir cada producto escalar por el *kernel* específico. Así, obtendríamos la fórmula general para la decisión como:

$$f(\mathbf{x}) = signo\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right)$$

Y el problema de optimización cuadrática como:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}, \mathbf{x}_i)$$

$$\text{Sujeto a: } \alpha_i \geq 0\ ;\ i = 1, \ldots, n\ ;\ \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0$$

En la Figura 3 s muestra un ejemplo de la frontera de decisión encontrada con un SVM utilizando *kernel* RBF. En la figura se muestran los datos de ambas clases (negros y transparentes); mientras que aquellos puntos con doble circunferencia indican los vectores soporte. La línea central separando ambas clases sería la frontera de decisión, y las líneas exteriores indican el margen de cada clase respecto a la frontera.

Una de las ventajas de SVM es que es muy disperso en cuanto a los valores de α (vectores soporte), ya que muchos patrones tienen un valor de $\alpha_i = 0$. Gracias a esta propiedad, SVM es aun práctico para conjuntos de datos grandes.

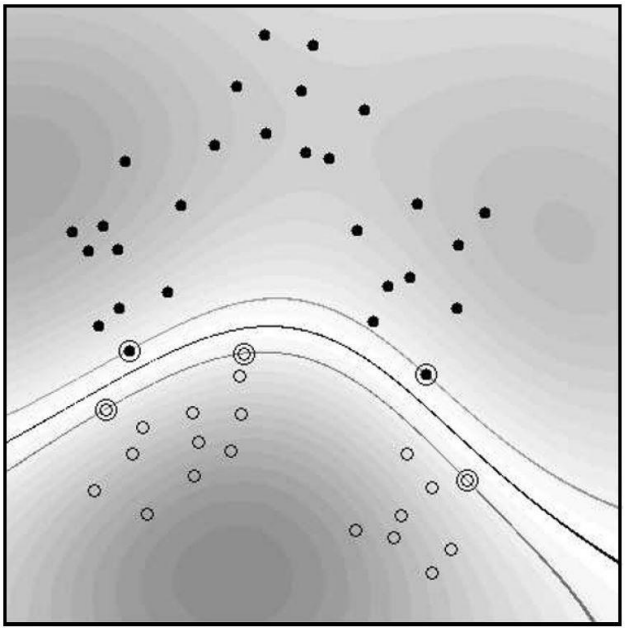

Figura 3. Ejemplo de SVM utilizando kernel RBF.

3. SVM en datos no separables

Hasta el momento, hemos considerado que el problema era linealmente separable. Sin embargo, en la práctica, es raro encontrar datos donde un hiperplano pueda separar de forma correcta todos los datos de entrenamiento.

Para poder trabajar con dichos problemas, se propone el método de *soft-margin SVM*, donde se permite que algunos patrones incumplan la restricción de que todos los $\alpha_i > 0$. Para ello, se definen variables de holgura (*slack variables*) ξ_i que relajan las restricciones del margen vistas anteriormente; de modo que:

$$y_i \cdot \left(\left(\mathbf{w} \cdot \Phi(\mathbf{x}_i) \right) + b \right) \geq 1 - \xi_i \, , \xi_i \geq 0, i = 1, \ldots, n$$

Un clasificador que generaliza bien se obtiene gracias a controlar tanto la capacidad del clasificador (via $\|\mathbf{w}\|$), y la suma de las variables de holgura $\sum_{i=1}^{n} \xi_i$, es decir, el número de errores. A partir de estas consideraciones, el método llamado C-SVM implementa una minimización del *soft margin* en base a la siguiente función objetivo:

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

La constante de regularización $C > 0$ determina la compensación entre el error empírico y la complejidad del modelo. Incorporándolo a los multiplicadores de Lagrange, lleva al siguiente problema:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x},\mathbf{x}_i)$$

$$\text{Sujeto a: } 0 \le \alpha_i \le C\ ;\ i = 1,\dots,n\ ;\ \textstyle\sum_{i=1}^{n}\alpha_i y_i = 0$$

La única diferencia con el caso de datos separables es el límite C en los multiplicadores de Lagrange α_i.

Otra posible versión del SVM con *soft margin*, llamado ν-SVM, reemplaza la constante C para regularización (que puede no ser muy intuitiva), por otro valor $\nu \in [0,1]$, quedando la formulación del problema que se muestra a continuación:

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x},\mathbf{x}_i)$$

$$\text{Sujeto a: } 0 \le \alpha_i \le \tfrac{1}{n}\ ;\ i = 1,\dots,n\ ;\ \textstyle\sum_{i=1}^{n}\alpha_i y_i = 0\ ;\ \sum_{i=1}^{n}\alpha_i \ge \nu$$

Bajo una correcta elección de parámetros, ambos C-SVM y ν-SVM obtienen exactamente los mismos resultados. La principal diferencia es que en el caso de ν-SVM, ν es una cota superior a la fracción de errores en el margen, y también ν es una cota inferior a la fracción de vectores soporte. Por tanto, controlar ν influye a la compensación o solución intermedia entre la precisión del modelo y su complejidad.

En la Figura 4 se muestra una comparación entre SVM *hard margin* y *soft margin*. En el segundo caso, hay patrones que pueden estar entre los márgenes, o incluso al otro lado del margen. De este modo, se busca obtener un modelo final más robusto sobre nuevos datos de test.

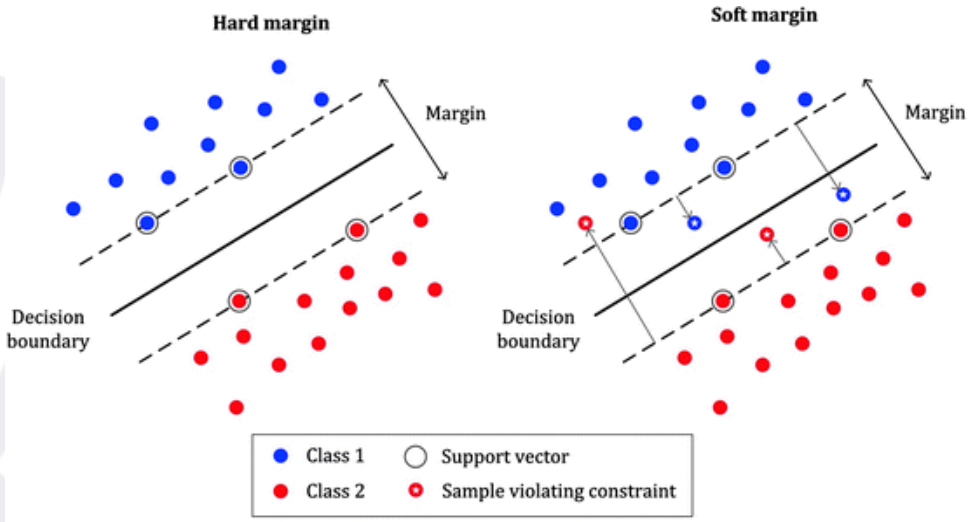

Figura 4. Comparación entre SVM clásico (hard margin) y soft margin.

Referencias

[Mai10] Maimon, O., & Rokach, L. (Eds.). (2010). Data mining and knowledge discovery handbook, 2nd edition. *Springer*.