

Máquinas de Vectores de Soporte

Luis Norberto Zúñiga Morales

11 de enero de 2022

Índice

1. Motivación	1
2. Máquina de Vectores de Soporte Lineal	3
2.1. Teoría	3
2.2. Implementación Práctica	7
3. Máquina de Vectores de Soporte y el Margen Suave	8
3.1. Teoría	8
3.2. Implementación Práctica	9
4. Máquinas de Vectores de Soporte para Casos No Lineales	10
4.1. Implementación Práctica	12
5. Máquinas de Vectores de Soporte para el Caso Multiclase	13
6. Ejercicios	14

Resumen

La Máquina de Vectores de Soporte (MVS) es un poderoso modelo de aprendizaje automático que es utilizado en aplicaciones que cubren un amplio espectro de áreas de estudio, tales como medicina, lingüística computacional, cómputo financiero, psicología, entre muchas otras. La MVS [3] es un algoritmo de clasificación binaria (i.e., separa objetos de dos clases distintas) que puede adaptarse a un problema de clasificación multiclase (más de dos clases) y que, gracias al truco del kernel, es posible salir de la idea inicial de separar clases por medio de un hiperplano y utilizar funciones con formas más complejas para realizar dicha separación. Lo anterior las vuelve atractivas para trabajar con problemas de clasificación cuyo conjunto de datos se encuentren entre un rango de tamaño bajo y medio.

1. Motivación

La idea básica que surge al considerar el problema de separar elementos de dos clases distintas de datos, los cuales se pueden representar como vectores en \mathbb{R}^n , es la forma de realizar

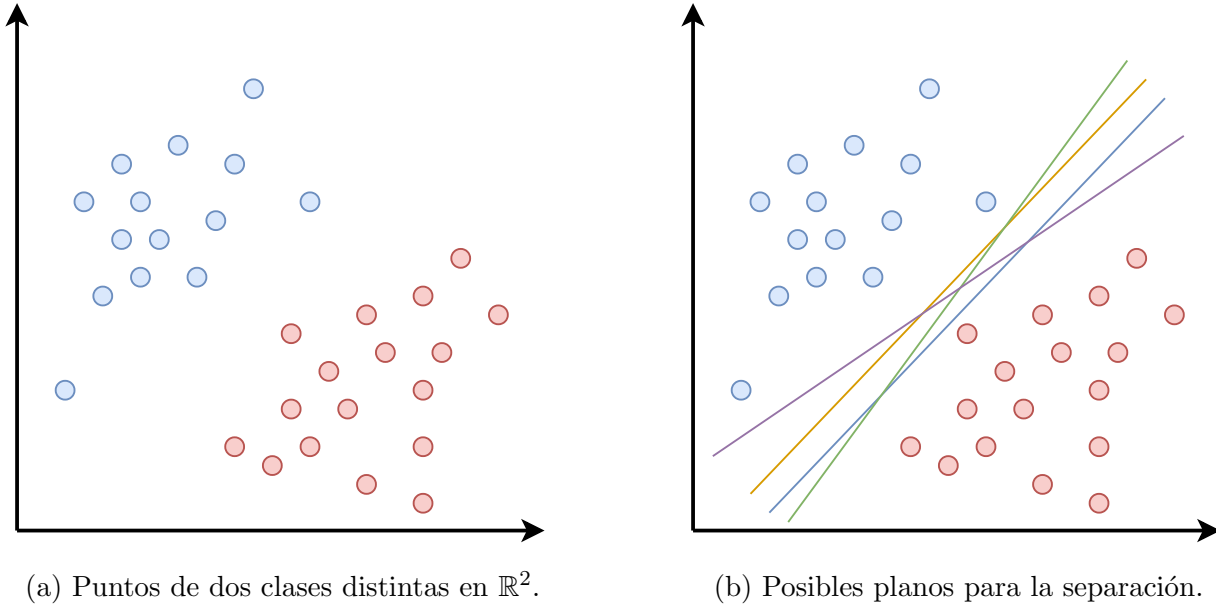


Figura 1: Ejemplo sencillo del problema de separación de clases. ¿Cuántos hiperplanos se pueden construir? ¿Cuál de ellos se debe elegir?

tal separación. Lo más sencillo es la construcción de un hiperplano que separe los puntos de tal manera que de un lado se encuentren aquellos de la clase A y del otro aquellos de la clase B ¹. Para ejemplificar el problema anterior, consideremos el problema de clasificación de la Fig. (1a) que muestra un caso a modo para nosotros, cosa que rara vez se observa en práctica. Al momento de separar las clases, ¿de cuántas maneras es posible construir un hiperplano que separe puntos de ambas clases? La Fig. (1b) muestra algunos hiperplanos propuestos que resuelven perfectamente el problema planteado. ¿Cuál de ellos se debe elegir? ¿Existe alguno que se pueda denominar «mejor»?

Después de esta breve discusión, es natural preguntarse si existe una manera ordenada que permita construir un plano único que separe, de alguna manera, óptimamente los puntos que representan a cada miembro de una clase en particular. Por supuesto, faltaría definir con qué se relaciona ese óptimo y condiciones adicionales para encontrarlo. Las Máquinas de Vectores de Soporte abordan esta pregunta planteando un problema que busca al hiperplano que maximiza la distancia entre este y ciertos puntos del conjunto de datos de una manera elegante y relativamente sencilla.

¹Para este modelo, hasta que se diga lo contrario, se considerará el caso de clasificación binaria, el cual permitirá formular el problema básico a atacar, y que posteriormente se expandirá a problemas de clasificación multiclase.

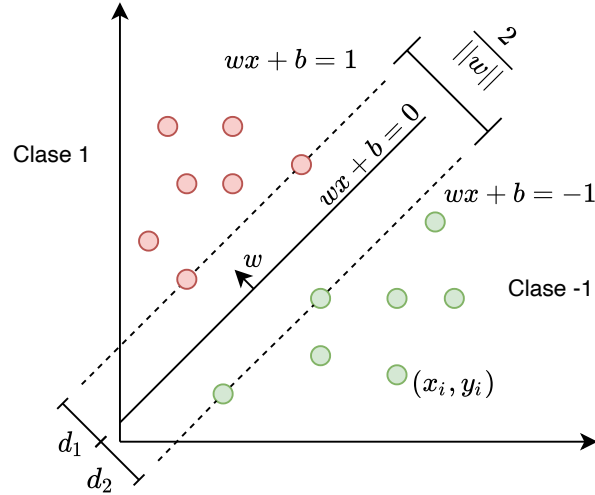


Figura 2: Diagrama que muestra la idea básica de la Máquina de Vectores de Soporte.

2. Máquina de Vectores de Soporte Lineal

2.1. Teoría

Para formular la idea básica de la MVS es útil comenzar con el caso más sencillo, el cual se presenta al considerar funciones lineales para construir la función que separa los puntos de ambas clases. A lo anterior se le conoce como un problema de **clasificación binaria linealmente separable**.

Supongamos que se tienen L puntos de entrenamiento $\{\mathbf{x}_i, y_i\}$, donde $\mathbf{x}_i \in \mathbb{R}^D$ representa un vector de entrada de dimensión D (el vector de atributos o características) el cual se asocia a una de dos clases posibles: $y_i = +1$ o $y_i = -1$. Al asumir un problema de clasificación separable linealmente, la función que separa los puntos de ambas clases es una recta cuando $D = 2$, y un hiperplano cuando $D > 2$. Para no perder generalidad, dicho hiperplano se puede definir de la siguiente manera:

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0 \quad (1)$$

donde:

- \mathbf{w} es un vector normal al hiperplano
- $\frac{b}{\|\mathbf{w}\|}$ es la distancia perpendicular del hiperplano al origen

Para determinar el hiperplano separador, es posible utilizar a los puntos \mathbf{x}_i que se encuentren más cercanos a este, de tal forma que la separación se encuentre lo más lejos posible de los puntos más cercanos de ambas clases. La Fig. (2) muestra la idea anterior, donde los hiperplanos formados por los vectores de soporte (punteado) forman un margen donde se ubicará el hiperplano separador (sólido), idealmente en medio de este.

Para encontrar el hiperplano separador mediante la MVS, se necesita encontrar aquellos \mathbf{w} y b tales que

$$\mathbf{w}^T \cdot \mathbf{x}_i + b \geq +1 \quad \text{para } y_i = +1 \quad (2)$$

$$\mathbf{w}^T \cdot \mathbf{x}_i + b \leq -1 \quad \text{para } y_i = -1 \quad (3)$$

Las ecuaciones (2) y (3) se pueden combinar convenientemente en una sola ecuación:

$$y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i \quad (4)$$

Las ecuaciones de los hiperplanos H_1 y H_2 de soporte (uno para cada clase) se encuentran dadas por²:

$$\mathbf{w}^T \cdot \mathbf{x} + b = +1 \quad \text{para } H_1 \quad (5)$$

$$\mathbf{w}^T \cdot \mathbf{x} + b = -1 \quad \text{para } H_2 \quad (6)$$

En la Fig. (2) se muestra que el hiperplano separador se encuentra en medio del espacio formado por H_1 y H_2 , el cual se llama **margen de la MVS**, donde d_1 y d_2 denotan la distancia de H_1 y H_2 al hiperplano separador, respectivamente. Este margen debe ser lo más amplio posible, por lo que se presenta un problema de maximización. Sin embargo, ¿cuánto mide el margen de la MVS?

Sea \mathbf{x}_j un punto en el hiperplano $\mathbf{w}^T \cdot \mathbf{x} + b = -1$. Para medir la distancia entre los hiperplanos $\mathbf{w}^T \cdot \mathbf{x} + b = +1$ y $\mathbf{w}^T \cdot \mathbf{x} + b = -1$ es necesario calcular la longitud del segmento perpendicular desde \mathbf{x}_j hasta el plano $\mathbf{w}^T \cdot \mathbf{x} + b = +1$. Denotemos como z dicha cantidad. La clave aquí es denotar z en términos de \mathbf{w} , lo cual se logra al considerar el vector unitario $\frac{\mathbf{w}}{\|\mathbf{w}\|}$ del hiperplano $\mathbf{w}^T \cdot \mathbf{x} + b = +1$, y multiplicarlo por z . Por lo tanto, se tiene que

$$\mathbf{w}^T(\mathbf{x}_j + z \frac{\mathbf{w}}{\|\mathbf{w}\|}) + b = 1 \quad (7)$$

ya que $\mathbf{x}_j + z \frac{\mathbf{w}}{\|\mathbf{w}\|}$ es un punto que yace en $\mathbf{w}^T \cdot \mathbf{x} + b = +1$. En otras palabras, si al punto \mathbf{x}_j , que por definición se encuentra en el hiperplano $\mathbf{w}^T \cdot \mathbf{x} + b = -1$, le sumamos la distancia z , terminamos con un punto en el hiperplano $\mathbf{w}^T \cdot \mathbf{x} + b = +1$. Expandiendo la ecuación (7):

²Noten que en las ecuaciones (5) y (6) se utiliza una igualdad ya que se tocan elementos de cada clase, mientras que el hiperplano separador solo las divide en dos regiones.

$$\begin{aligned}
\mathbf{w}^T(\mathbf{x}_j + z \frac{\mathbf{w}}{\|\mathbf{w}\|}) + b &= 1 \\
\mathbf{w}^T \cdot \mathbf{x}_j + z \frac{\mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} + b &= 1 \\
\mathbf{w}^T \cdot \mathbf{x}_j + z \frac{\|\mathbf{w}\|^2}{\|\mathbf{w}\|} + b &= 1 \\
\mathbf{w}^T \cdot \mathbf{x}_j + z \|\mathbf{w}\| + b &= 1 \\
\mathbf{w}^T \cdot \mathbf{x}_j + b &= 1 - z \|\mathbf{w}\| \\
-1 &= 1 - z \|\mathbf{w}\| \\
2 &= z \|\mathbf{w}\| \\
z &= \frac{2}{\|\mathbf{w}\|}
\end{aligned} \tag{8}$$

Para maximizar $\frac{2}{\|\mathbf{w}\|}$, la distancia entre los hiperplanos H_1 y H_2 , se debe minimizar³ $\|\mathbf{w}\|$. Por lo tanto, el problema de optimización resultante es el siguiente:

$$\begin{aligned}
&\text{mín} \quad \|\mathbf{w}\| \\
&\text{sujeto a} \quad y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1
\end{aligned} \tag{9}$$

Al minimizar $\|\mathbf{w}\|$, notemos que esta no es diferenciable en 0, por lo que en su lugar⁴ se minimiza $\frac{1}{2}\|\mathbf{w}\|^2$. Oportunamente, de esta forma es posible solucionar el problema de optimización cuadrática más adelante. En consecuencia, es necesario encontrar la solución a:

$$\begin{aligned}
&\text{mín} \quad \frac{1}{2}\|\mathbf{w}\|^2 \\
&\text{sujeto a} \quad y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1
\end{aligned} \tag{10}$$

El siguiente paso es resolver el problema de minimización, por lo que vamos a utilizar multiplicadores de Lagrange α , donde $\alpha_i \geq 0 \forall i$.

$$\mathbb{L}_P = \frac{1}{2}\|\mathbf{w}\|^2 - \alpha[y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) - 1] \quad \forall i \tag{11}$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^L \alpha_i[y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) - 1] \tag{12}$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^L \alpha_i y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) + \sum_{i=1}^L \alpha_i \tag{13}$$

Como se mencionó anteriormente, se debe encontrar \mathbf{w} y b tales que minimicen (10) y los $\alpha_i \geq 0$ que maximicen (13). Para lograrlo, diferenciamos \mathbb{L}_P con respecto de \mathbf{w} y b , igualando las derivadas parciales a cero:⁵

³Entre más pequeño sea el denominador en una fracción, mayor es el resultado final.

⁴Los algoritmos de optimización suelen funcionar mejor cuando se aplican en funciones diferenciables en cualquier punto.

⁵Es decir, el clásico problema de optimizar mediante derivadas.

$$\frac{\partial \mathbb{L}_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^L \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^L \alpha_i y_i \mathbf{x}_i \quad (14)$$

$$\frac{\partial \mathbb{L}_P}{\partial b} = \sum_{i=1}^L \alpha_i y_i = 0 \quad (15)$$

Al sustituir las ecuaciones (14) y (15) en la ecuación (13), se obtiene una nueva ecuación que depende de $\boldsymbol{\alpha}$, por lo que hay que maximizar:

$$\mathbb{L}_D = \sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j, \quad \alpha_i \geq 0, \quad \sum_{i=1}^L \alpha_i y_i = 0 \quad (16)$$

$$= \sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i H_{ij} \alpha_j, \quad H_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (17)$$

$$= \sum_{i=1}^L \alpha_i - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha}, \quad \alpha_i \geq 0, \quad \sum_{i=1}^L \alpha_i y_i = 0 \quad (18)$$

La ecuación (18) se le conoce como el **dual** del **primal** \mathbb{L}_P ⁶ En este caso, el dual tiene la característica de requerir únicamente el producto punto de los vectores de entrada \mathbf{x}_i , el cual será útil más adelante cuando se introduzca el truco del kernel.

Durante este proceso, se partió de minimizar \mathbb{L}_P a maximizar \mathbb{L}_D . Es decir, el problema de optimización luce de la siguiente forma:

$$\begin{aligned} & \underset{\boldsymbol{\alpha}}{\text{máx}} \quad \sum_{i=1}^L \alpha_i - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} \\ & \text{sujeto a} \quad \alpha_i \geq 0 \\ & \quad \sum_{i=1}^L \alpha_i y_i = 0 \end{aligned} \quad (19)$$

Al problema descrito en (19) se le conoce como un problema de **optimización cuadrático convexo**, el cual se puede resolver (convenientemente) por medio de algoritmos o paquetes de software específicos. En resumen, al resolver el problema de optimización cuadrático, se obtendrá $\boldsymbol{\alpha}$, y de la ecuación (14) se puede obtener \mathbf{w} . Solo faltaría calcular b .

¿Recuerdan que durante al planteamiento de la idea se mencionó que los márgenes se construyen mediante vectores de apoyo del conjunto de datos? Cualquier punto \mathbf{x}_s que funcione como vector de soporte satisface la ecuación

$$y_s(\mathbf{x}_s \cdot \mathbf{w} + b) = 1 \quad (20)$$

Sustituyendo en (14):

$$y_s \left(\sum_{k \in S} \alpha_k y_k \mathbf{x}_k \cdot \mathbf{x}_s + b \right) = 1 \quad (21)$$

⁶De ahí los subíndices P y D en \mathbb{L} .

donde S denota el conjunto de índices de los vectores de soporte, el cual se construye al encontrar los índices i tales que $\alpha_i > 0$. Multiplicando por y_s , observando que $y_s^2 = 1$ de (2) y (3):

$$\begin{aligned} y_s^2 \left(\sum_{k \in S} \alpha_k y_k \mathbf{x}_k \cdot \mathbf{x}_s + b \right) &= y_s \\ \Rightarrow b &= y_s - \sum_{k \in S} \alpha_k y_k \mathbf{x}_k \cdot \mathbf{x}_s \end{aligned} \quad (22)$$

Ahora, surge la pregunta sobre cuál o cuáles vectores de soporte \mathbf{x}_s utilizar para calcular b . En la práctica, se utiliza un promedio de todos los vectores en S :

$$b = \frac{1}{N_S} \sum_{s \in S} \left(y_s - \sum_{k \in S} \alpha_k y_k \mathbf{x}_k \cdot \mathbf{x}_s \right) \quad (23)$$

Finalmente se tiene \mathbf{w} y b para construir el hiperplano óptimo que separe los datos y, en consecuencia, la Máquina de Vectores de Soporte.

2.2. Implementación Práctica

1. Crear \mathbf{H} , donde $H_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$.
2. Encontrar las α_i que maximicen

$$\sum_{i=1}^L \alpha_i - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha}$$

sueto a las restricciones

$$\alpha_i \geq 0, \quad \sum_{i=1}^L \alpha_i y_i = 0.$$

En práctica, se utilizan librerías en distintos lenguajes de programación que resuelvan el problema de optimización cuadrática, como [qpsovers](#) en Python.

3. Calcular $\mathbf{w} = \sum_{i=1}^L \alpha_i y_i \mathbf{x}_i$.
4. Determinar el conjunto de vectores de soporte S mediante la identificación de los índices i tales que $\alpha_i > 0$.
5. Calcular el valor de b mediante la ecuación

$$b = \frac{1}{N_S} \sum_{s \in S} \left(y_s - \sum_{k \in S} \alpha_k y_k \mathbf{x}_k \cdot \mathbf{x}_s \right).$$

6. Cada elemento del conjunto de prueba x_t se clasifica evaluando

$$y_t = \text{sgn}(\mathbf{w}^T \cdot \mathbf{x}_t + b).$$

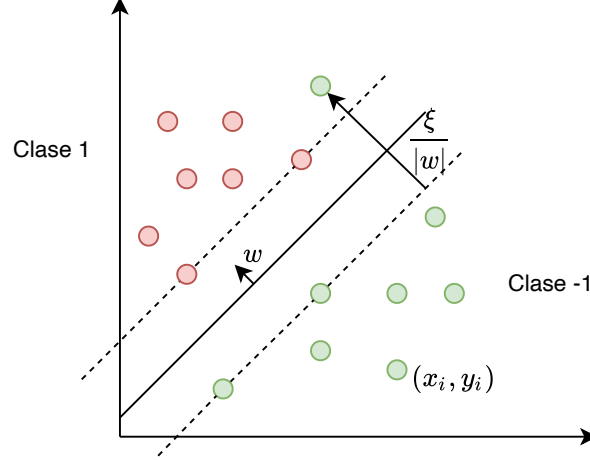


Figura 3: Idea de la MVS con penalización.

3. Máquina de Vectores de Soporte y el Margen Suave

3.1. Teoría

El caso de clasificación de datos linealmente separable permite crear los fundamentos de la MVS. Sin embargo, el caso hecho a la medida rara vez se aprecia en entornos prácticos. El científico de datos frecuentemente se enfrenta a casos donde los datos no son totalmente linealmente separables. El caso lineal permite construir **márgenes duros**, los cuales no consideran errores al momento de clasificar. Con este problema en mente, se introduce el concepto del **margen suave** de la MVS.

La idea es relajar las limitantes dadas en (2) y (3) para permitir puntos mal clasificados. Lo anterior se logra introduciendo un variable de holgura positiva ξ_i , $i = 1, \dots, L$:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 - \xi_i \quad y_i = +1 \quad (24)$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 + \xi_i \quad y_i = -1 \quad (25)$$

$$\xi_i \geq 0 \quad \forall i \quad (26)$$

Similar a la ecuación (4), las ecuaciones (24), (25) y (26) se pueden combinar en una sola:

$$y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{donde} \quad \xi_i \geq 0 \quad \forall i \quad (27)$$

El margen suave penaliza los puntos que se encuentren en el lado equivocado del hiperplano de separación. Dicha penalización incrementa conforme aumenta la distancia entre ellos. De esta forma, se busca reducir el número de clasificaciones incorrectas. Para incluir esta idea en el modelo de la MVS, se adapta la ecuación (10) de la siguiente manera:

$$\begin{aligned} \text{mín} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^L \xi_i \\ \text{sujeto a} \quad & y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i \end{aligned} \quad (28)$$

donde el parámetro C controla la razón de intercambio entre la penalización y el tamaño del margen.

De manera similar a la MVS Lineal, es momento de calcular el Lagrangiano, el cual debe ser minimizado con respecto a \mathbf{w} , b y ahora ξ_i ; y maximizado con respecto a α y μ :

$$\mathbb{L}_P = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^L \xi_i - \sum_{i=1}^L \alpha_i [y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^L \mu_i \xi_i \quad (29)$$

Diferenciando con respecto a \mathbf{w} , b y ξ_i e igualando a cero:

$$\frac{\partial \mathbb{L}_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^L \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^L \alpha_i y_i \mathbf{x}_i \quad (30)$$

$$\frac{\partial \mathbb{L}_P}{\partial b} = \sum_{i=1}^L \alpha_i y_i = 0 \quad (31)$$

$$\frac{\partial \mathbb{L}_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \Rightarrow C = \alpha_i + \mu_i \quad (32)$$

Sustituyendo en (29), \mathbb{L}_D tiene la misma forma que la ecuación 16 en el caso lineal. Sin embargo, al considerar la ecuación 32 y el hecho que $\mu_i \geq 0 \forall i$ ⁷, se sigue que $\alpha > C$. Por lo tanto, se debe encontrar:

$$\begin{aligned} & \underset{\alpha}{\text{máx}} \quad \sum_{i=1}^L \alpha_i - \frac{1}{2} \alpha^T \mathbf{H} \alpha \\ & \text{sujeto a} \quad 0 \leq \alpha_i \leq C \quad \forall_i \\ & \quad \sum_{i=1}^L \alpha_i y_i = 0 \end{aligned} \quad (33)$$

Para calcular b , se sigue la misma idea que en el caso lineal dado que la ecuación (30) y (14) son la misma. En este caso, los vectores de soporte son aquellos cuyos índices i cumplan la desigualdad $0 \leq \alpha_i \leq C$.

3.2. Implementación Práctica

1. Crear \mathbf{H} , donde $H_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$.
2. Elegir el valor del parámetro C , el cual permitirá penalizar clasificaciones erróneas.
3. Encontrar las α_i que maximicen

$$\sum_{i=1}^L \alpha_i - \frac{1}{2} \alpha^T \mathbf{H} \alpha$$

⁷Ya que las μ_i son multiplicadores de Lagrange que se añaden al considerar las variables de holgura.

sujeito a las restricciones

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^L \alpha_i y_i = 0.$$

mediante un programa para resolver problemas de optimización cuadrática.

4. Calcular $\mathbf{w} = \sum_{i=1}^L \alpha_i y_i \mathbf{x}_i$.
5. Determinar el conjunto de vectores de soporte S mediante la identificación de los índices i tales que $0 \leq \alpha_i \leq C$.
6. Calcular el valor de b mediante la ecuación

$$b = \frac{1}{N_S} \sum_{s \in S} (y_s - \sum_{k \in S} \alpha_k y_k \mathbf{x}_k \cdot \mathbf{x}_s).$$

7. Cada elemento del conjunto de prueba x_t se clasifica evaluando

$$y_t = \text{sgn}(\mathbf{w}^T \cdot \mathbf{x}_t + b).$$

4. Máquinas de Vectores de Soporte para Casos No Lineales

El caso de separación lineal permitió sentar las bases para el modelo de MVS. Desafortunadamente, en la práctica, esta clase de problemas rara vez se ve en algún conjunto de datos. El caso más común es aquel donde los datos no pueden ser separados fácilmente con un hiperplano y requieren funciones con formas más complejas, las cuales suelen ser no lineales. Durante la construcción del clasificador, en la ecuación (17) se identificó la matriz

$$H_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (34)$$

El producto punto de los vectores de entradas en la matriz anterior se puede representar como una función:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j = \mathbf{x}_i^T \mathbf{x}_j \quad (35)$$

La función $k(\mathbf{x}_i, \mathbf{x}_j)$ es un ejemplo de una familia de funciones llamadas **kernels**, las cuales se basan en el cálculo de productos puntos de los vectores de entrada del conjunto de datos. La finalidad de los kernels es que son funciones $x \mapsto \phi(x)$ que permiten mapear los datos a diferentes dimensiones sin la necesidad de determinar la función ϕ que realiza el mapeo, ya que esta se encuentra dada por nosotros, y únicamente se necesita determinar el producto punto de los vectores.

El **truco del kernel** permite atacar problemas que no son linealmente separables en el espacio en turno, y al realizar el mapeo por medio del kernel, es posible que en otro espacio sí sea separable. La Fig. (4) muestra un ejemplo de un problema de separación donde los datos de ambas clases no pueden separarse por medio de una MVS lineal, por lo que se recurre a una MVS con el truco del kernel.

Los kernels más comunes en la práctica son:

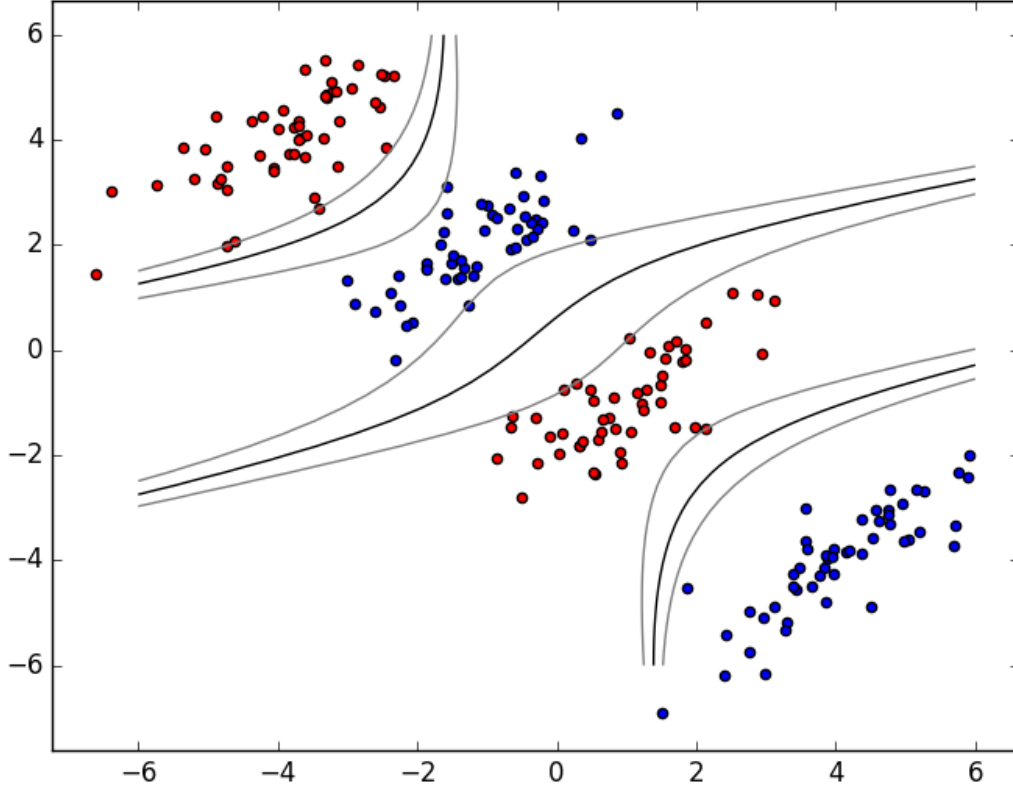


Figura 4: Ejemplo de un conjunto de datos que no es linealmente separable y una MVS que se ajusta para separar los puntos de una forma no lineal.

- Kernel Lineal: $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
- Kernel Polinomial: $[\gamma(\mathbf{x}_i \cdot \mathbf{x}_j) + r]^d$
- Kernel Función de Base Radial: $\exp(-\gamma \cdot \|\mathbf{x}_i - \mathbf{x}_j\|^2)$
- Kernel Sigmoide: $\tanh(\mathbf{x}_i \cdot \mathbf{x}_j + r)$

donde $\gamma > 0$ y $r, d \in \mathbb{R}$.

Expresado en la formulación del clasificador en la ecuación (28):

$$\begin{aligned} \text{mín} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^L \xi_i \\ \text{sujeto a} \quad & y_i(\mathbf{w}^T \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \quad \forall i \end{aligned} \tag{36}$$

y su dual en la ecuación (33)

$$\begin{aligned}
& \underset{\alpha}{\text{máx}} \quad \sum_{i=1}^L \alpha_i - \frac{1}{2} \alpha^T \mathbf{H}_k \alpha \\
& \text{sujeto a} \quad 0 \leq \alpha_i \leq C \quad \forall_i \\
& \quad \sum_{i=1}^L \alpha_i y_i = 0
\end{aligned} \tag{37}$$

donde

$$\mathbf{H}_k = y_i y_j k(\mathbf{x}_i, \mathbf{x}_j). \tag{38}$$

4.1. Implementación Práctica

1. Elegir de antemano cual es el kernel que se aplicará en la MVS y la función de mapeo $\phi(\mathbf{x})$. En práctica, el kernel de función de base radial funciona mejor.
2. Crear \mathbf{H}_k , donde $H_{ij} = y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$.
3. Elegir el valor del parámetro C , el cual permitirá penalizar clasificaciones erróneas.
4. Encontrar las α_i que maximicen

$$\sum_{i=1}^L \alpha_i - \frac{1}{2} \alpha^T \mathbf{H}_k \alpha$$

sujeto a las restricciones

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^L \alpha_i y_i = 0.$$

mediante un programa para resolver problemas de optimización cuadrática.

5. Calcular $\mathbf{w} = \sum_{i=1}^L \alpha_i y_i \phi(\mathbf{x}_i)$.
6. Determinar el conjunto de vectores de soporte S mediante la identificación de los índices i tales que $0 \leq \alpha_i \leq C$.
7. Calcular el valor de b mediante la ecuación

$$b = \frac{1}{N_S} \sum_{s \in S} (y_s - \sum_{k \in S} \alpha_k y_k k(\mathbf{x}_k, \mathbf{x}_s)).$$

8. Cada elemento del conjunto de prueba x_t se clasifica evaluando

$$y_t = \text{sgn}(\mathbf{w}^T \cdot \phi(\mathbf{x}_t) + b).$$

5. Máquinas de Vectores de Soporte para el Caso Multiclase

Originalmente las máquinas de vectores de soporte se plantearon considerando un modelo de clasificación binaria, es decir, solo separan dos clases. Múltiples enfoques se han considerado para extender el algoritmo para el caso de clasificación multiclase. Se consideran dos formas de atacar el problema: uno contra todos y uno contra uno.

El enfoque **uno contra todos** [2] construye n modelos de MVS, uno para cada clase considerada. El i -ésimo modelo considera la j -ésima clase y sus elementos como la clase positiva y las $n - 1$ clases restantes se consideran como las negativas. Dado el conjunto de datos $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_L, y_L)$, la j -ésima MVS resuelve el siguiente problema:

$$\begin{aligned} \text{mín} \quad & \frac{1}{2} \|\mathbf{w}^j\|^2 + C \sum_{i=1}^L \xi_i^j \\ \text{sujeto a} \quad & (\mathbf{w}^j)^T \cdot \phi(\mathbf{x}_i) + b^j \geq 1 - \xi_i^j \quad \text{si } y_j = i \\ & (\mathbf{w}^j)^T \cdot \phi(\mathbf{x}_i) + b^j \leq -1 + \xi_i^j \quad \text{si } y_j \neq i \\ & \xi_i^j \geq 0 \end{aligned} \tag{39}$$

Al resolver la ecuación (39) se tienen n funciones de decisión:

$$\begin{aligned} & (\mathbf{w}^1)^T \phi(\mathbf{x}) + b^1 \\ & \vdots \\ & (\mathbf{w}^n)^T \phi(\mathbf{x}) + b^n \end{aligned}$$

\mathbf{x}_i pertenece a la clase con el mayor valor en la función de decisión:

$$y = \underset{j \in \{1, \dots, n\}}{\operatorname{argmax}} (\mathbf{w}^j)^T \phi(\mathbf{x}_i) + b^j. \tag{40}$$

El segundo método es llamado **uno contra uno**. Dicho enfoque construye $n(n - 1)/2$ clasificadores donde cada uno se entrena con información de dos clases. Considerando los datos de la clase j y la clase k se resuelve el siguiente problema:

$$\begin{aligned} \text{mín} \quad & \frac{1}{2} \|\mathbf{w}^{jk}\|^2 + C \sum_{i=1}^L \xi_i^{jk} \\ \text{sujeto a} \quad & (\mathbf{w}^{jk})^T \cdot \phi(\mathbf{x}_i) + b^{jk} \geq 1 - \xi_i^{jk} \quad \text{si } y_i = j \\ & (\mathbf{w}^{jk})^T \cdot \phi(\mathbf{x}_i) + b^{jk} \leq -1 + \xi_i^{jk} \quad \text{si } y_i = k \\ & \xi_i^{jk} \geq 0 \end{aligned} \tag{41}$$

Hsu y Lin [2] deciden usar la siguiente estrategia basada en votos: si $\operatorname{sign}((\mathbf{w}^{jk})^T \phi(\mathbf{x}) + b^{jk})$ dice que \mathbf{x} pertenece a la j -ésima clase, se suma un voto a esa clase. Si dice que pertenece a la clase k , entonces se da el voto a la clase j . La clase que se elige para \mathbf{x} es aquella que tenga más votos.

6. Ejercicios

1. Para calcular la distancia entre los hiperplanos de apoyo H_1 y H_2 se llegó al resultado expuesto en la ecuación (8) mediante la interpretación de r en términos de \mathbf{w} y un vector normal al hiperplano de separación. Existe otro argumento para calcular r mediante la proyección de un vector en otro. Determinar r por medio de esta idea.
2. Una tarea que se deja a todo estudiante de ciencia de datos es la implementación del algoritmo de la MVS por cuenta propia. Debemos mantener viva esa bonita tradición. Implementar su propia versión de MVS. **Nota:** El paso más complejo es resolver el problema de optimización cuadrática. Utilicen cualquier librería que pueda facilitarles la resolución de ese paso.
3. Investigar qué es el teorema de Mercer y cuál es su relación con la MVS.
4. Consideren un kernel polinomial en un conjunto de datos arbitrario. ¿Qué pasa cuando varían el parámetro d ? Graficar los márgenes que se crean al considerar distintos valores de $d \in [1, 5]$. **Nota:** Armen un conjunto de puntos aleatorio que sea linealmente separable y construyan los márgenes de separación en este punto. Como idea, consideren [este tutorial](#) en la sección *Polinomial Kernel*.

Referencias

- [1] W. Chao. A tutorial for support vector machine. Disponible en: <http://disp.ee.ntu.edu.tw/~pujols/Support%20Vector%20Machine.pdf>.
- [2] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.
- [3] V. Vapnik and C. Cortes. Support-vector networks. *Machine Learning*, 20:273–297, 1995.