Universidad del Valle

Facultad de Ciencias Naturales y Exactas
Universidad del Valle

REVISTA DE CIENCIAS

# AL INTERIOR DE UNA MÁQUINA DE SOPORTE VECTORIAL

**Leonardo Jiménez Moscovitz**
Fundación Universitaria Konrad Lorenz

**Pervys Rengifo Rengifo**
Fundación Universitaria Konrad Lorenz

Recibido: marzo 16, 2010 Aceptado: octubre 19, 2010

Pág. 73-85

**Resumen**

En este artículo se exponen aspectos teóricos generales y algorítmicos al interior de una Máquina de Soporte Vectorial (MSV), la función kernel asociada y el algoritmo SMO, vitales en la actual solución eficiente de problemas de clasificación. Se describe de manera detallada la aplicación de los algoritmos en la solución de un problema simple, complementando la extensa literatura teórica existente.

**Palabras claves:** vectores de soporte, aprendizaje supervisado, clasificación de datos, kernel, hiperplano separador, multiplicadores de Lagrange, algoritmo SMO.

**Abstract**

This article addresses theoretical and algorithmic issues related to Support Vector Machines (SVMs), kernel functions and the SMO algorithm, all important tools in the solution of classification problems. Detailed operation of these algorithms applied to the solution of a simple problem is described, trying to fill a gap in the extensive theoretical literature about SVM.

**Keywords:** support vectors, supervised learning, data classification, kernel, separating hyperplane, Lagrange multipliers, SMO algorithm.

## 1 Introducción

Una *Máquina de Soporte Vectorial* (MSV) es un sistema de aprendizaje automático que permiten resolver problemas de clasificación y regresión de manera muy eficiente y que se ha posicionado por encima de otras técnicas, tales como las redes neuronales. Las MSV est án siendo utilizadas con éxito en diversas áreas de la inform ática e inteligencia artificial.

Las MSV se basan en la Teoría del Aprendizaje Estadístico [1] desarrollada por *V. Vapnik* y *A. Chervonenkis*, hacia el año 1992 cuando proponen un modelo matemático para la resolución de problemas de clasificación y regresión al cual llamaron Modelo MSV [1][2]. En pocos años se crearon aplicaciones para la solución de problemas reales, destacándose como una herramienta robusta en dominios complejos, ruidosos y con escasos datos.

El éxito de las MSV radica en tres ventajas fundamentales: la primera consiste en que poseen una sólida fundamentación matemática. La segunda consiste en que se basan en el concepto de la *minimización del riesgo estructural* [4][3], esto es, minimizar la probabilidad de una clasificación errónea sobre nuevos ejemplos, particularmente importante cuando se disponen de pocos datos de entrenamiento. La tercera ventaja radica en que disponen de potentes herramientas y algoritmos para hallar la solución de manera rápida y eficiente.

Este artículo tiene como principal objetivo presentar la mecánica interior en las MSV y surge dentro de un proyecto de investigación en la FUKL, cuyos aportes se orientaron hacia el exámen algorítmico de esta técnica y la implementación sencilla de estas técnicas tanto en lenguaje C como en Excel, con orientación didáctica. Para ello se introducen algunos aspectos teóricos sobre las MSV como clasificadores lineales, y dado que usualmente no hay un clasificador lineal para los datos originales de un problema real, se describen las funciones kernel, que solucionan este problema eficientemente. Se finaliza con una visión al interior de una MSV en la solución de un problema juguete, haciendo lo más explícito posible cada uno de los pasos principales necesarios hasta llegar a la solución utilizando el algoritmo SMO (*sequential minimal optimization*) desarrollado por Platt [8].

## 2 Problema linealmente separable

Para solucionar un problema de clasificación, la MSV debe aprender una superficie de decisión adecuada, basándose en el conjunto de datos de entrenamiento. La superficie de decisión es un hiperplano que separa los patrones de entrenamiento en dos clases, según se encuentran a uno u otro lado del mismo.

Se dispone entonces de un conjunto de $N$ datos de entrenamiento llamados *patrones*, de la forma $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$..., $(\mathbf{x}_N, y_N)$ donde $\mathbf{x} \in \mathbb{R}^n$. Cada escalar $y_i$ (llamado *etiqueta)* corresponderá a una de dos clases que se identificarán como $+1$ y $-1$. Se llamará vector de etiquetas al vector $\mathbf{y} = (y_1, y_2, ...y_N)$.

En un problema linealmente separable existen muchos hiperplanos que pueden clasificar los datos. Pero las MSV no hallan uno cualquiera de estos hiperplanos, sino el único que maximiza la distancia entre él y el dato más cercano de cada clase (figura 1). Esta distancia es llamada *margen*, y al hiperplano que la maximiza se le llama Hiperplano de Máximo Margen o de Separación Óptima (HSO)[6].

El hiperplano separador está dado de manera general por

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \,\mathtt{donde}\, \mathbf{w}, \mathbf{x} \in \mathbb{R}^n, b \in \mathbb{R} \qquad (1)$$

donde el trabajo consiste en hallar el vector $\mathbf{w}$ de *pesos* que contiene la ponderación de cada atributo, indicando qué tanto aportan en el proceso de clasificación, en tanto que $b$ define el umbral de decisión, llamado usualmente *bias* en inglés. La función discriminante (distancia) $d$ será la función

$$d(\mathbf{x},\mathbf{w},b)=\frac{|(\mathbf{w}\cdot\mathbf{x})+b|}{\|\mathbf{w}\|}\ \texttt{con}\ \|\mathbf{w}\|=\sqrt{(\mathbf{w}\cdot\mathbf{w})}$$

donde $\|\mathbf{w}\|$ es la norma asociada al producto escalar en $\mathbb{R}^n$. Como se trata de patrones linealmente separables, se puede reescalar $\mathbf{w}$ y $b$, de tal manera que:

$$d(\mathbf{x},\mathbf{w},b)=\frac{1}{\|\mathbf{w}\|}\implies|\mathbf{w}\cdot\mathbf{x}_i+b|=1 \tag{2}$$

Así se obtiene el HSO canónico en el cual los patrones de entrenamiento más cercanos al plano tienen distancia normalizada $d(\mathbf{x},\mathbf{w},b)=1$, con $d(\mathbf{x},\mathbf{w},b)\geq 1$ para los demás patrones.

Hallar el mejor hiperplano separador es un clásico problema de maximización con restricciones de la ecuación (2), el cual se puede transformar en un problema más sencillo utilizando el principio de dualidad [7][4], quedando como

$$min\frac{\|\mathbf{w}\|^2}{2}\ \texttt{sujeto a}: y_i\left[(\mathbf{w}\cdot\mathbf{x}_i)+b\right]\geq 1\ \texttt{para}\ i=1,2,...,N \tag{3}$$

Aplicando los multiplicadores de Lagrange ($\alpha_i$) la ecuación (3) se reformula como [4]:

$$L\left(\mathbf{w},b,\alpha\right)=\frac{\|\mathbf{w}\|^2}{2}-\sum_{i=1}^{N}\alpha_i\left[y_i\left[(\mathbf{w}\cdot\mathbf{x}_i)+b\right]-1\right] \tag{4}$$

Para hallar el punto de silla $(\mathbf{w}_0,b_0,\alpha_0)$, se debe minimizar $L\left(\mathbf{w},b,\alpha\right)$ con respecto a $\mathbf{w}$ y $b$ y maximizar con respecto a $\alpha\geq 0$, la cual representa una solución en el espacio primal. El problema puede solucionar en el espacio dual (ver [3] y [4]), para lo cual se replantea como

$$min\ :\ L_d\left(\alpha\right)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i^T\cdot\mathbf{x}_j) \tag{5}$$

$$sujeto\ a\ :\ \sum_{i=1}^{N}\alpha_i y_i=0\ \ \mathbf{y}\ \ \alpha_i\geq 0, i\in\{1,...,N\} \tag{6}$$

donde $L_d$ es la forma dual de $L$, la cual depende de los multiplicadores de Lagrange. Esta representación es preferida, ya que $L_d$ depende únicamente del *producto escalar* de los patrones de entrada $\mathbf{x}$ lo cual simplifica los cálculos, como se verá en la sección 3.

Esta formulación del problema satisface las condiciones de Karush-Kuhn-Tucker (KKT) y por tanto se tienen las condiciones necesarias y suficientes para que un valor extremo exista [7]. Además, la solución siempre conduce *al mismo vector normal* al hiperplano separador. Tomando la representación matricial [3] de las ecuación (5) y (6) se obtiene

$$\texttt{min:}L_d(\alpha)=\frac{1}{2}\alpha^T H\alpha-\mathbf{f}^T\alpha\ \texttt{s.a.:}\ \ \mathbf{y}^T\alpha=0,\alpha\geq 0 \tag{7}$$

donde se utiliza el vector unitario $\mathbf{f} = [11 \cdots 1]^T$ y mediante cualquier método de optimización se halla el vector de multiplicadores $\alpha_0 = (\alpha_1^0, \alpha_2^0, ..., \alpha_N^0)$ para determinar finalmente el vector normal $\mathbf{w}_0$ y el *bias* $b_0$ del HSO así

$$\mathbf{w}_0 = \sum_{i=1}^{N} y_i \alpha_i^0 \mathbf{x}_i \quad \mathbf{y} \quad b_0 = -\frac{1}{2}\mathbf{w}_0 \cdot [\mathbf{x}_r + \mathbf{x}_s] \tag{8}$$

Esto indica que $\mathbf{w}_0$ se puede expresar como combinación lineal de los $N$ vectores de entrada. De las condiciones de KKT se desprende que gran parte de los $\alpha_i^0$ son 0, y solo una menor cantidad $N_{SV}$ de vectores del conjunto de entrada participan en la combinación lineal que origina a $\mathbf{w}_0$: son los vectores de soporte (SV). En la ecuación (8) $\mathbf{x}_r$ y $\mathbf{x}_s$ son un par de vectores de soporte, uno de cada clase.

El clasificador buscado se puede construir finalmente como

$$f(x) = sign\left(\mathbf{w}_0 \cdot \mathbf{x} + b_0\right) = sign\left(\sum_{i \in SV} \mathbf{y}_i \alpha_i^0 \left(\mathbf{x}_i \mathbf{x}\right) + b_0\right) \tag{9}$$

donde el signo resultante indicará a cual clase pertenece un dato determinado. Esta expresión conlleva gran economía computacional, ya que la sumatoria no se realiza sobre todos los puntos de entrenamiento, únicamente sobre los que son vectores de soporte y el número $N_{SV}$ de **SV** puede ser muy pequeño, siendo en general mucho menor que $N$.

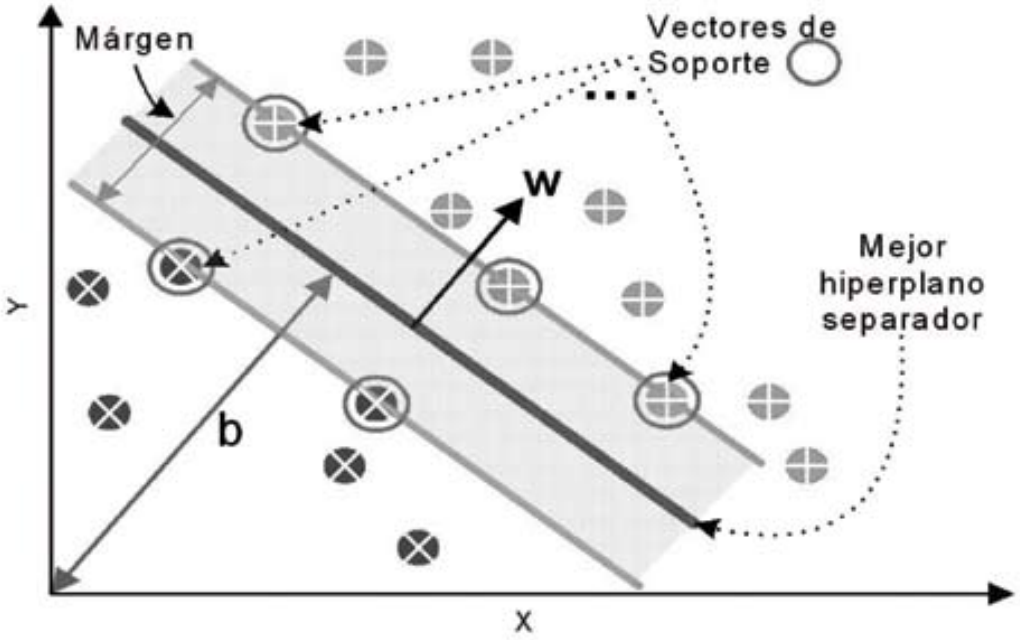


*Figura 1. Hiperplano de Separación Óptima* $\mathbf{w} \cdot \mathbf{x} + b$ *para el caso bidimensional*

## 3 Clasificadores no lineales mediante Kernel

Cuando no existe una apropiada superficie lineal de decisión en el espacio de entrada, se considera el mapeo del vector de entrada $\mathbf{x}$ en un espacio de mayor dimensión $\mathbb{R}^c$ llamado **espacio de características** $\tau$, que esté dotado de producto escalar. Eligiendo el espacio $\tau$ apropiado, se realiza el mapeo y se busca el HSO siguiendo la mecánica expuesta en la sección 2 y que será lineal en $\mathbb{R}^c$, pero representa un hiperplano no lineal en el espacio de entrada $\mathbb{R}^n$.

Supóngase que este mapeo se realiza mediante una función no lineal de la forma

$$\begin{aligned} \Phi(\mathbf{x}) &: \mathbb{R}^n \longrightarrow \mathbb{R}^c, c > n \\ \Phi(\mathbf{x}) &: \mathbf{x} \longmapsto \Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_c(\mathbf{x})) . \end{aligned}$$

Se puede observar una dificultad práctica para hallar el HSO en $\tau$, ya que se debe solucionar el problema planteado en la ecuación (5), replanteada ahora como la minimización de $L_d = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$ donde los vectores de entrenamiento solo aparecen en la forma de producto escalar, pero definido en el espacio de características. El c álculo de $(\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}))$ es demasiado exigente computacionalmente, ya que $c$ es mucho más grande que $n$. La solución de este grave inconveniente viene de mano de las funciones kernel.

Una función kernel es una función $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ tal que

$$K : (\mathbf{x}_i, \mathbf{x}_j) \rightarrowtail \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{10}$$

que representa el producto punto en un espacio de características de dimensión arbitraria [1][3]. Al utilizar la Función Kernel se puede obtener el producto escalar en el espacio de características, pero realizando el cálculo en el espacio de entrada cuya complejidad es mucho menor. Más aún, con el método del Kernel no es necesario conocer explícitamente la función $\Phi$.

**Ejemplo 3.1 (Kernel)** *Sea* $\Phi(\mathbf{x}) : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ *la función tal que*

$$\Phi(\mathbf{x}) : (x_1, x_2) \rightarrowtail \left(4x_1^2, 4\sqrt{2}x_1x_2, 4x_2^2\right) \tag{11}$$

*Realizando el producto punto se observa que*

$$\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) = \left(16x_{i1}^2 x_{j1}^2 + 32x_{i1}x_{i2}x_{j1}x_{j2} + 16x_{i2}^2 x_{j2}^2\right) = \left[4\left(\mathbf{x}_i \cdot \mathbf{x}_j\right)\right]^2 \tag{12}$$

*donde la última expresión es la Función Kernel de la ecuación (11):*

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left[4\left(\mathbf{x}_i \cdot \mathbf{x}_j\right)\right]^2 = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

Un desarrollo detallado se expone en Vapnik [1], Kecman [3], Cristianini [4] y Schölkopft [5] entre otros. Para que una función $K$ sea eligible como función kernel, debe cumplir con las condiciones de Mercer [3] y como tal debe ser una función simétrica cuyo codominio sea un espacio de Hilbert, y la matriz Hessiana $H = \left[K\left(\mathbf{x}_i, \mathbf{x}_j\right)\right]_{i,j=1}^{N}$ asociada al problema de optimización planteado en la ecuación (7) debe ser semidefinida positiva, garantizando la convexidad del problema y por tanto la existencia de solución.

Todo lo derivado para el caso lineal es también aplicable para un caso no lineal usando un Kernel conveniente $K$, donde la función de decisión en el espacio $\tau$ es:

$$f(\mathbf{x}) = sign\left(\sum_{i \in SV} \mathbf{y}_i \alpha_i K\left(\mathbf{x}, \mathbf{x}_i\right) + b\right) \tag{13}$$

Construyendo y aplicando diferentes funciones kernel, el algoritmo de SV puede construir una gran variedad de máquinas de aprendizaje [6].

## 4 Al interior de la MSV

Se supone un conjunto de patrones de un problema simple de clasificación binaria con tan solo 12 datos, descritos en la figura 2 (a). Este conjunto de datos representa dos clases: La clase $+1$, la de los puntos al borde o al interior de un círculo de radio 1 con centro en $(0,0)$ y la clase $-1$, la de los puntos exteriores a dicho círculo. Esta clasificación debe ser aprendida por la máquina únicamente a partir del conjunto de datos indicado. La figura 2 (b) de la tabla muestra los mismos datos pero transformados al espacio de características de dimensión 3; aunque no será necesario utilizar éstos en los cálculos, se pueden graficar a partir de la tabla para visualizar el hiperplano separador y el mecanismo de clasificación de nuevos patrones.

Se trata entonces de hallar un clasificador que permita asignar la etiqueta correcta a cualquier dato que se presente. Los pasos generales a seguir para hallar la solución con MSV se esbozan en la figura 3 y se describen a continuación.

### 4.1 Primer paso: determinación del tipo de problema

La inspección del problema suministra información valiosa acerca del problema: su separabilidad lineal, rango de variación de los datos y, con mayor experiencia, el investigador obtiene información valiosa para la adecuada selección de kernel y parámetros.

| Espacio de entrada | | |
|---|---|---|
| Atributos | | Clase |
| $X_1$ | $X_2$ | Y |
| -0,626 | 0,609 | +1 |
| -0,323 | -0,843 | +1 |
| -0,089 | 0,624 | +1 |
| 0,019 | -0,021 | +1 |
| 0,062 | -0,817 | +1 |
| 0,476 | -0,008 | +1 |
| -1,733 | -0,823 | -1 |
| -1,457 | -1,106 | -1 |
| -1,365 | -0,355 | -1 |
| -0,193 | 1,107 | -1 |
| 0,080 | -1,674 | -1 |
| 1,084 | 0,394 | -1 |

(a)

| Espacio de características | | | |
|---|---|---|---|
| Atributos | | | Clase |
| $X_1$ | $X_2$ | $X_3$ | Y |
| 0,392 | -0,539 | 0,371 | +1 |
| 0,104 | 0,385 | 0,711 | +1 |
| 0,008 | -0,079 | 0,389 | +1 |
| 0,000 | -0,001 | 0,000 | +1 |
| 0,004 | -0,072 | 0,667 | +1 |
| 0,227 | -0,005 | 0,000 | +1 |
| 3,003 | 2,017 | 0,677 | -1 |
| 2,123 | 2,279 | 1,223 | -1 |
| 1,863 | 0,685 | 0,126 | -1 |
| 0,037 | -0,302 | 1,225 | -1 |
| 0,006 | -0,189 | 2,802 | -1 |
| 1,175 | 0,604 | 0,155 | -1 |

(b)

*Figura 2. Datos del problema*

En la figura 2. (a) están los datos originales en $\mathbb{R}^2$, y en (b) los datos en el espacio de características en $\mathbb{R}^3$ después de la transformación $\Phi(x) : (x_1, x_2) \rightarrow \left(16x_1^2, 32x_1x_2, 16x_2^2\right)$.

## 4.2 Segundo paso: selección y aplicación de la función Kernel

Aún cuando se pueden hacer recomendaciones para su elección, en problemas complejos se deben realizar muchas pruebas, ajustar parámetros para cada kernel y comparar resultados. Aquí se ha seleccionado un kernel simple, con el cual se obtuvieron resultados satisfactorios: el kernel polinomial $K(\mathbf{x}, \mathbf{x}_i) = [4(\mathbf{x} \cdot \mathbf{x}_i)]^2$ definido en la ecuación (11).

## 4.3 Tercer paso: construcción de la matriz Kernel

A partir de la función kernel $K(\mathbf{x}, \mathbf{x}_i) = [4(\mathbf{x} \cdot \mathbf{x}_i)]^2$ se genera la matriz kernel $\mathbf{H}_{N \times N}$. Cada elemento se construye como

$$\mathbf{H}_{i,j} = y_i y_j K(\mathbf{x}_{i*}, \mathbf{x}_{j*}) = y_i y_j \left[4(\mathbf{x}_i \cdot \mathbf{x}_j)\right]^2 \tag{14}$$

que representa a la matriz Hessiana correspondiente al problema. En este ejemplo, de acuerdo con la ecuación (14), la matriz Kernel se desarrolla como

$$\begin{aligned} \mathbf{H}_{1,1} &= (y_1) \cdot (y_1) \cdot [4((x_{11}x_{11}) + (x_{12}x_{12}))]^2 \\ &= (-1) \cdot (-1) \cdot [4((-0.626 * -0.626) + (0.609 * 0.609))]^2 = 9.3088 \end{aligned}$$

hasta completar la matriz Hessiana $H$ :

$$\mathbf{H}_{12\times12} = \begin{bmatrix} 9.308 & 1.549 & \cdots & -3.078 \\ \cdots & \cdots & \cdots & \cdots \\ -3.078 & -7.447 & \cdots & 28.314 \end{bmatrix}$$

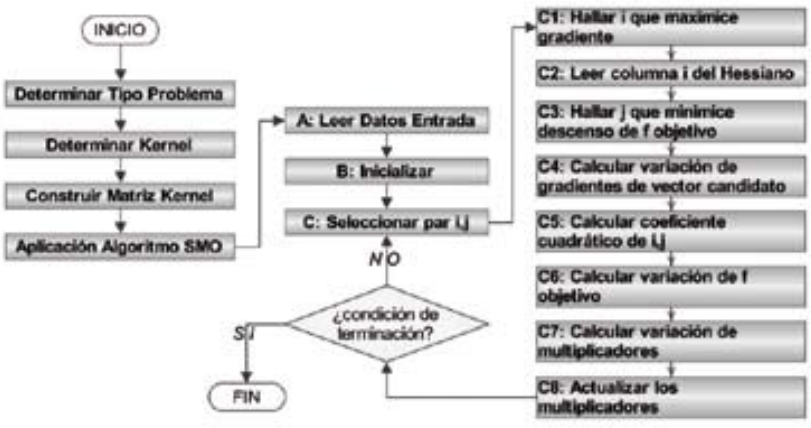


*Figura 3. Algoritmo simplificado para hallar un modelo mediante MSV*

### 4.4 Cuarto paso: minimización con el algoritmo SMO

Se tiene un problema de optimización cuadrático y mediante la matriz Hessiana **H** el problema se reexpresa como lo indica la ecuación (7). Es un problema lineal cuya solución se abordará utilizando el algoritmo SMO, muy eficiente para hallar los multiplicadores desconocidos $\alpha$, desarrollado por Platt [8] y cuya aplicación al problema actual se puede concretar así:

**Paso A.** Se lee la tabla de datos de entrada, asignando 12 ternas $(x_{i1}, x_{i2}, y_i)$ con $0 \leq i \leq 11$.

**Paso B.** Inicializar: se inicializan los multiplicadores de Lagrange: $\alpha_i = 0$ y los gradientes $\nabla_i = -1$, para $0 \leq i \leq 11$.

**Paso C.** Seleccionar un par de vectores $i, j$ para realizar una iteración: se seleccionará primero el vector $i$ y con base en éste se selecciona luego el vector $j$.

**Paso C1.** Hallar vector que maximiza el gradiente: se debe hallar vector identificado con el índice $i$ tal que $[-y_i \nabla_i]$ sea máximo. Si es la primera iteración, todos son iguales y se asigna el último índice ($i = 5$ para la clase $+1$ en la primera iteración). Si no es la primera iteración, de las anteriores iteraciones se tienen ya dos vectores $i', j'$ con su respectivo multiplicador de Lagrange $\alpha_{i'}, \alpha_{j'}$ y se recalcula el gradiente

$$\max\left[-y_i \cdot \nabla_k\right] \quad | \quad \nabla_k = \nabla_k + ((\mathbf{H}_{ik}\Delta\alpha_{i'}) + (\mathbf{H}_{j'k}\Delta\alpha_{j'}))$$

donde $\Delta\alpha_i$ representa la variación en el valor de $\alpha_i$ entre iteraciones y $0 \leq k \leq N$. Para el ejemplo, el valor $\nabla_0$ se calcula

$$\begin{aligned} \nabla_0 &= \nabla_0 + ((\mathbf{H}_{i'0}\Delta\alpha_{i'}) + (\mathbf{H}_{j'0}\Delta\alpha_{j'})) \\ &= (-1) + ((1.4675 \cdot 0.0965) + (-3.0785 \cdot 0.0965)) = -1.1555 \end{aligned}$$

y análogamente para los demás valores. Como $\max(-\nabla_k) = -\nabla_1 = -1.6855$, se toma $i = k = 1$.

**Paso C2.** Leer la columna $i$ de la matriz Hessiana **H**: Para el caso de la primera iteración, $i = 5$ y los valores correspondientes son:

$$\mathbf{H}_{5k} = \begin{bmatrix} 1.4675 & 0.3458 & \cdots\cdots & -0.0424 & -4.2079 \end{bmatrix}$$

donde $0 \leq k \leq N$. Para el caso de la segunda iteración se obtiene $i = 1$, y los valores correspondientes son:

$$\mathbf{H}_{1k} = \begin{bmatrix} 1.5494 & 10.6270 & \cdots\cdots & -30.7068 & -7.4480 \end{bmatrix}$$

**Paso C3.** Minimizar el descenso de la función objetivo: el segundo vector, que se llamará $j$, se selecciona de tal manera que cumpla con esta condición. Para ello se verifica si con el vector $\mathbf{H}_{ik}$ hallado en el paso anterior, se cumple que $-y_k \cdot \mathtt{grad}\,(f_k) < -y_i \cdot \mathtt{grad}\,(f_i)$. Esto se logra en los pasos siguientes.

**Paso C4**. Calcular la variación en gradiente asociado con cada vector candidato $k$ : se llamará $dif\nabla_k$, que se calcula como $dif\nabla_k = \nabla_{\max} + \nabla_k$.

**Paso C5.** Se calcula el valor del coeficiente cuadrático $Qco_{i,k}$ asociado al par de vectores $i, j$ mediante la expresión

$$Qco_{i,k} = \mathbf{H}_{ii} + \mathbf{H}_{kk} \pm (2y_i\mathbf{H}_{ik}) \tag{15}$$

Este cálculo no es necesario si el multiplicador de Lagrange es un valor extremo (cota superior o inferior). En la ecuación (15), el signo $+/-$ para la suma se toma según $y_i$ sea positivo o negativo.

Los valores que se han calculado para la primera iteración son:

$$Qco_{5,6} = 196.1441, ..., Qco_{5,11} = 20.7208.$$

Aparecen únicamente 6 valores, dado que los otros 6 multiplicadores $a$ corresponde a una cota, y por tanto no se calculan.

**Paso C6**. Se calcula la variación de la función objetivo $\Delta f_k$ y se toma el valor mínimo, así

$$\Delta f_k = \min\left[-(dif\nabla_k)^2 Qco_{i,k}\right] \texttt{si} Qco_{i,k} > 0$$

mientras que si $Qco_{i,k} \leq 0$ este valor se reemplaza por $\tau$, que es un parámetro predeterminado asociado con un valor muy pequeño (usualmente $10^{-6}$), que asegure que $\Delta f_k$ tome un valor muy grande y sea descartado eventualmente.

Para la primera iteración, los valores de $\Delta f_k$ encontrados son:
$\Delta f_6 = -0.020393, ..., \Delta f_{11} = -0.193043$. Este último valor es el que hace mínimo el descenso de la función objetivo, y por tanto $\min[\Delta f_k] = \Delta f_{11} = -0.193043$. Luego se toma $j = k = 11$. Con esto se obtiene el par $i, j$ de vectores identificados con $i = 5$ y $j = 11$.

Para la segunda iteración, se ha hecho necesario calcular únicamente los valores correspondientes a $k = 5, 9, 10, 11$, de los cuales se puede comprobar que $\Delta f_9 = -0.579086$ es el *mínimo*. Luego, se toma como par $i, j$ a los vectores $i = 1$ y $j = 9$.

**Paso C7**. Hallar el valor $\Delta_{i,j}$: indica cuanto deben variar los multiplicadores de Lagrange, de acuerdo con la participación de cada uno de ellos en la variación de la función objetivo. Se calcula como:

$$\Delta_{i,j} = \pm\nabla_i - \nabla_j Qco_{i,j}$$

donde se toma el signo $+/-$ dependiendo si los $y's$ correspondientes a $i$ y $j$ son diferentes o iguales.

**Paso C8**. Se actualizan los multiplicadores, así:

$$\alpha_i = \alpha_i + \Delta_{i,j}; \alpha_j = \alpha_j + \Delta_{i,j}$$

y se hacen $dif\_\alpha_{i,j} = \alpha_i - \alpha_j$, y $sum\_\alpha_{i,j} = \alpha_i + \alpha_j$, valores que permitirán realizar ajustes válidos cuando algún $\alpha$ toma valores negativos. Entonces, en la primera iteración se obtuvieron como vectores de trabajo $i = 5$ y $j = 11$.

$$\text{Dado que } \nabla_5 = -1, \nabla_{11} = -1, \text{ y } Qco_{5,11} = 20.7208,$$

$$\text{se obtiene } \Delta_{5,11} = [-(-1) - (-1)]/20.7208 = 0.096521.$$

Como todos los multiplicadores se han inicializado con 0, los multiplicadores correspondientes a los vectores 5 y 11 se actualizan, anotando además la *variación* correspondiente del $\alpha$ como $\Delta\alpha$:

$$\alpha_5 = 0 + 0.096521 = 0.096521 \quad \Delta\alpha_5 = 0.096521 - 0 = 0.096521$$
$$\alpha_{11} = 0 + 0.096521 = 0.096521 \quad \Delta\alpha_{11} = 0.096521 - 0 = 0.096521$$

En la segunda iteración se obtuvieron como vectores de trabajo $i = 1$ y $j = 9$. Los valores de gradiente son $\nabla_1 = -1.6855$ y $\nabla_9 = -0.936127$ con

$$Qco_{1,9} = 11.868707, \text{ donde se obtiene}$$
$$\Delta_{1,9} = [-(-1.6855) - (-0.936127)]/11.868707 = 0.220\,89.$$

Dado que los vectores 1 y 9 no habían sido modificados en la primera iteración, su coeficiente viene con valor 0 y se actualizan así:

$$\alpha_1 = 0 + 0.22089 = 0.22089$$
$$\alpha_9 = 0 + 0.22089 = 0.22089$$

Es claro entonces, que al final de la segunda iteración, los multiplicadores de Lagrange tienen los valores $\alpha_1 = 0.22089$, $\alpha_5 = 0.096521$, $\alpha_9 = 0.22089$, $\alpha_{11} = 0.096521$ y los restantes son aún 0.

Se debe tener en cuenta que si eventualmente algún $\alpha < 0$, se debe realizar una corrección, para lo cual se consideran las siguientes posibilidades:

1. Si los $y's$ asociados al par $i, j$ son de clases diferentes entonces si $\alpha_i < 0$ se toma $\alpha_i = 0$ y $\alpha_j = \pm(dif\_\alpha_{i,j})$, mientras que si $\alpha_j < 0$ se toma $\alpha_j = 0$ y $\alpha_i = \pm dif\_\alpha_{i,j}$. Aquí el signo $+/-$ se utiliza a conveniencia para que haga el multiplicador positivo.

2. Si los $y's$ asociados al par $i, j$ son de la misma clase se realizan las siguientes verificaciones y ajustes, con el fin de obtener valores de los multiplicadores dentro del rango aceptable $[0, C_k]$, donde $C_k$ es una cota que se predefine (como parámetro) para limitar, a voluntad del investigador, el rango de valores que puede tomar el multiplicador: Si $sum\_\alpha_{i,j} \leq C_j$ y $\alpha_i < 0$ se toma $\alpha_i = 0$ y $\alpha_j = sum\_\alpha_{i,j}$. Si $sum\_\alpha_{i,j} > C_j$ y $\alpha_j > C_j$ se toma $\alpha_j = sum\_\alpha_{i,j} - C_j$ y $\alpha_i = C_j$. Si $sum\_\alpha_{i,j} \leq C_i$ y $\alpha_j < 0$, se toma $\alpha_i = sum\_\alpha_{i,j}$ y $\alpha_j = 0$. Y finalmente, si $sum\_\alpha_{i,j} > C_i$ y $\alpha_i > C_i$, se toma $\alpha_i = C_i$ y $\alpha_j = sum\_\alpha_{i,j} - C_i$

Se verifica a continuación la condición de terminación. Si ésta no se ha cumplido aún, se procede a una nueva iteración a partir del paso 4.4. Existen diferentes maneras de examinar la condición de terminación. La más sencilla de ellas pide que se examine la suma de los gradientes de $i$ y de $j$. Si este valor es menor que un valor de umbral predeterminado, se asume que se está cerca a un valor óptimo según las condiciones de Lagrange y el programa finaliza.

Con un umbral de $10^{-3}$, se observa que al superar las 25 iteraciones, los valores cambian muy lentamente y se llega al umbral de terminación luego de 43 iteraciones, obteniéndose 4 multiplicadores de Lagrange diferentes de cero que son: $\alpha_1 = 0.3795 \quad \alpha_2 = 0.6519 \quad \alpha_{10} = -0.6531 \quad \alpha_{12} = -0.3782$. Esto quiere decir que se han obtenido 4 vectores de soporte. El subíndice $i$ de $\alpha_i$ indica cual es el patrón original del conjunto de entrenamiento de la figura 2 que sirve como vector de soporte.

## 4.5 Quinto paso: determinación de la función de decisión

Se toma la función que determina el problema y se reemplazan los valores, que se organizan así:

| i | $\alpha_i$ | $\mathbf{x}_{i1}$ | $\mathbf{x}_{i2}$ | $\mathbf{y}_i$ | $\mathbf{K}(\mathbf{x}_i, \mathbf{x}) = [4(\mathbf{x}_i \cdot \mathbf{x})]^2$ |
|---|---|---|---|---|---|
| 0 | 0.3795 | −0.626 | 0.609 | −1 | $6.27x_1^2 - 12.199x_1x_2 + 5.934\,1x_2^2$ |
| ... | ... | ... | ... | ... | ... |
| 11 | −0.3782 | 1.084 | 0.394 | 1 | $18.801x_1^2 + 13.667x_1x_2 + 2.483\,8x_2^2$ |

donde se tiene que, en la columna correspondiente al Kernel, el valor $\mathbf{x} \in \mathbb{R}^2$ tiene atributos que se denotan $x_1, x_2$ y su valor será dado por los atributos de cada patrón nuevo que se desee clasificar. La función de decisión es:

$$f(x) = sign\left(\sum_{i \in SV} \alpha_i \left(\mathbf{y}_i K(\mathbf{x}, \mathbf{x}_i)\right) + b\right)$$

$$= sign\left(\left(\begin{array}{c} (0.3795) \cdot \left(6.27x_1^2 - 12.199x_1x_2 + 5.934\,1x_2^2\right) + \\ (0.6519) \cdot \left(1.669\,3x_1^2 + 8.713\,2x_1x_2 + 11.37x_2^2\right) + \\ (-0.6531) \cdot \left(0.595\,98x_1^2 - 6.836\,8x_1x_2 + 19.607x_2^2\right) + \\ (-0.3782) \cdot \left(18.801x_1^2 + 13.667x_1x_2 + 2.483\,8x_2^2\right) \end{array}\right) + b\right)$$

$$= sign(-4.035\,3x_1^2 + 0.346\,5x_1x_2 - 4.083x_2^2 + b) \tag{16}$$

Se determina $b$ utilizando la expresión:

$$b = \frac{\left(\sum_{i \in SV} \alpha_i H_{ii}\right)}{N_{sv}} \tag{17}$$

que para este caso particular se tiene que $b = 4.2273$ y el clasificador buscado es finalmente:

$$f(x) = sign\left(-4.0353x_1^2 + 0.3465x_1x_2 - 4.083x_2^2 + 4.2273\right) \tag{18}$$

Esta función permitirá la clasificación correcta de nuevos ejemplos presentados al sistema, como se podrá comprobar fácilmente aplicando la misma a patrones arbitrarios $(x_1, x_2)$, seleccionados de tal manera que algunos patrones caigan dentro y otros fuera del círculo de radio 1 y centro en $(0, 0)$.

## 5 Conclusiones

Comparados con otros sistemas clasificadores, las máquinas de soporte vectorial son muy eficientes desde diversas perspectivas. Por una parte, se obtienen muy buenos resultados aún con conjuntos de datos de entrenamiento muy pequeños, tal como lo muestra el ejemplo desarrollado. Además, el proceso de aprendizaje es un proceso matemático definido que permite obtener siempre el mejor clasificador, no tan solo un buen clasificador como se obtiene en muchos entrenamientos de redes neuronales. Esto sin hablar de que el tiempo de entrenamiento es predecible y relativamente corto.

Por otra parte, una vez obtenido el modelo, es muy fácil implementarlo en diferentes sistemas. Además posee una alta velocidad de ejecución en la clasificación de grandes conjuntos de datos.

En cuanto a los detalles internos de las MSV, es claro que las funciones kernel son de vital importancia, ya que todos los cálculos necesarios son realizados en el espacio de entrada. Esta es una característica peculiar del algoritmo de los métodos de vectores de soporte. Se está tratando con algoritmos complejos para reconocimiento de patrones no lineales, regresión, o extracción de características, pero para los propósitos del cálculo sólo se requiere trabajar con un algoritmo lineal, de relativamente fácil implementación computacional.

En cuanto al ejemplo particular que se ha desarrollado, la solución presentada es la mejor solución que se puede encontrar con el kernel polinomial y los parámetros escogidos, pero puede existir una mejor solución con otro kernels o parámetros y generalmente es la experiencia del usuario la que puede guiar su obtención.

## Referencias bibliográficas

[1] V. Vapnik, *Statistical Learning Theory*, Wiley, New York. (1998)

[2] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, (1995)

[3] V. Kecman, *Learning and Soft Computing*. MIT Press, London (2001)

[4] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machine and other Kernel-based Learning Methods*. Cambridge Universtity Press, New York (2000).

[5] B. Schölkopft, A. Smola, *Learning with Kernels*. MIT Press, London (2002)

[6] H. Wang, X. Fu *Data Mining with Computational Intelligence*. Springer, Germany (2005)

[7] H. Mora, *Optimización No Lineal y Dinámica*. Editorial Unibiblos, Bogotá. (2001)

[8] J. Platt, *Fast Training of Support Vector Machines using Sequential Minimal Optimization* en *Schölkopf, Burgues, Smola: Advances in Kernel Methods - Support Vector Learning,* pags. 185-208. Editorial MIT Press, Cambridge, MA.(1999)

**Dirección de los autores**

Leonardo Jiménez Moscovitz
Fundación Universitaria Konrad Lorenz
Programa de Matemáticas
leonardo.jimenezm@fukl.edu.co

Pervys Rengifo Rengifo
Fundación Universitaria Konrad Lorenz
Programa de Matemáticas
prengifo@fukl.edu