TESIS

MAGÍSTER EN INGENIERÍA ELECTRÓNICA

BOOSTING SUPPORT VECTOR MACHINES

ELKIN EDUARDO GARCÍA DÍAZ

COD 200418195

ASESOR

FERNANDO LOZANO MARTÍNEZ PH.D.

UNIVERSIDAD DE LOS ANDES

FACULTAD DE INGENIERÍA

MAESTRÍA EN INGENIRÍA ELÉCTRICA Y ELECTRÓNICA

BOGOTÁ D.C.

2005

TABLA DE CONTENIDO

INTRODUCCIÓN

En la vida cotidiana el hombre se enfrenta frecuentemente con el problema de clasificación, tal es el caso de los caracteres escritos, el reconocimiento de voz o el diagnóstico médico. Sin embargo se ha probado en muchos casos que solucionar este tipo de problemas haciendo uso de un computador presenta bastantes dificultades **[1]**.

Existen diversas razones que conducen a que la resolución de dichos problemas sea bastante compleja, entre ellas se encuentran:

1. El origen de lo que se busca clasificar (patrones): caracteres escritos, símbolos, dibujos, imágenes biomédicas, objetos tridimensionales, firmas, huellas dactilares, imágenes de Teledetección, cromosomas, etc.

2. La forma adecuada de representar estos elementos.

3. Los requerimientos del sistema, especialmente en tiempo de respuesta, puesto que aunque algunos métodos sean superiores en éxito, no son aplicables en la práctica dadas estas restricciones.

4. Factores económicos en especificaciones del sistema de adquisición de datos (sensores) o en equipos de procesamiento muy potentes pueden dar resultados muy satisfactorios pero no pueden ser asumidos por los usuarios.

Dentro de las técnicas de clasificación se destacan los árboles de decisión, las redes neuronales **[1]** y las máquinas de vectores de soporte SVM **[2]** por sus siglas en inglés (*Support Vector Machines*).

Estas últimas han sido desarrolladas recientemente basándose en la teoría estadística de aprendizaje de Vapnik **[3]** y han tenido gran éxito al ser aplicadas en la resolución de problemas básicos de aprendizaje supervisado (clasificación y regresión) y en problemas prácticos reales **[2]**.

1 MARCO TEÓRICO

1.1 EL PROBLEMA DE CLASIFICACIÓN BINARIA

Sea $\mathcal{X}$ un espacio de entrada, $\mathcal{Y}$ un espacio de etiquetas y Δ una distribución sobre $\mathcal{X}$ y dada una secuencia $\mathcal{S} = \left\{\left\langle \mathbf{x}_i, y_i \right\rangle\right\}_{i=1}^{m}$ de ejemplos etiquetados donde cada $x_i \in \mathcal{X}$ independientes e idénticamente distribuidos de acuerdo a Δ y cada $y_i \in \mathcal{Y}$ es asignado de acuerdo a una regla posiblemente estocástica. En este caso del problema clasificación binario se restringe a $\mathcal{Y} = \{-1, +1\}$.

Una *regla de clasificación* llamada *hipótesis,* es una función $h : \mathcal{X} \mapsto \mathcal{Y}$ que asigna una etiqueta a cada elemento en el espacio de entrada. En el problema de clasificación binaria se tiene que $h : \mathcal{X} \mapsto [-1, +1]$, donde el signo de $h(\mathbf{x})$ es interpretado como la predicción de la etiqueta a ser asignada a la instancia $\mathbf{x}$, mientras que la magnitud $|h(\mathbf{x})|$ es interpretada como la "confianza" de esta predicción. Por otra parte una clase de hipótesis $\mathcal{H}$ es un conjunto de hipótesis compuesto por diferentes hipótesis en el espacio de entrada.

El desempeño de una hipótesis será evaluado utilizando el *error de generalización* R y el error empírico R_{emp} definidos como:

$$R(h) = P_{(\mathbf{x},y)\sim\Delta}\left\{\operatorname{sgn}\left(h(\mathbf{x})\right) \neq y\right\} \tag{1.1}$$

$$R_{emp}(h, \mathcal{S}, D) = \sum_{i=1}^{m} D(i)\, \mathrm{I}_{\left[\operatorname{sgn}(h(\mathbf{x})) \neq y\right]} \tag{1.2}$$

Donde $D \in \mathbb{R}^m$ es una distribución discreta sobre el conjunto de muestras etiquetadas y $\mathrm{I}_{[.]}$ es la función indicadora.

1.2 HIPERPLANO SEPARADOR ÓPTIMO EN EL ESPACIO DE ENTRADA

Considere una familia $\mathcal{H}$ de clasificadores lineales sobre un espacio de entrada $\mathcal{X}$. Es decir clasificadores de la forma $h_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle_{\mathcal{X}} + b\right)$ donde $\mathbf{w} \in \mathcal{X}$ y $b \in \mathbb{R}$.

Se dice que un conjunto de ejemplos etiquetados $\mathcal{S} = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{m}$ con $\mathbf{x}_i \in \mathcal{X}$ y $y_i \in \{-1, +1\}$ es *linealmente separable* cuando existe un $h_{\mathbf{w},b}(\mathbf{x}) \in \mathcal{H}$ tal que $h_{\mathbf{w},b}(\mathbf{x}_i) = y_i$ para todo $\langle \mathbf{x}_i, y_i \rangle \in \mathcal{S}$.

Adicionalmente, para un hiperplano $h_{\mathbf{w},b}(\mathbf{x}) \in \mathcal{H}$, el *margen geométrico* de un punto $\langle \mathbf{x}, y \rangle \in \mathcal{X} \times \{-1, +1\}$ está definido como

$$\rho_{\mathbf{w},b}\left(\langle \mathbf{x}, y \rangle\right) = \frac{y.h_{\mathbf{w},b}(\mathbf{x})}{\|\mathbf{w}\|} \tag{1.3}$$

Mientras que el *margen geométrico* de un hiperplano $h_{\mathbf{w},b}(\mathbf{x})$ sobre $\mathcal{S}$ está definido como

$$\rho_{\mathbf{w},b} = \min_{i=1,\ldots,m} \rho_{\mathbf{w},b}\left(\langle \mathbf{x}_i, y_i \rangle\right) \tag{1.4}$$

1.2.1 CASO SEPARABLE

En problemas linealmente separables, a partir del hiperplano separador definido por $\mathbf{w}$ y b se definen 2 hiperplanos paralelos a éste de tal forma que en los puntos más cercanos al hiperplano ($\mathbf{x}_1$ y $\mathbf{x}_2$ para la *Figura 1.*) se cumpla que $\left|\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right| = 1$, esto se obtiene reescalando de forma adecuada $(\mathbf{w}, b)$.

A partir de esto se tiene que $\langle \mathbf{w}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle = 2$, entonces $\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, (\mathbf{x}_1 - \mathbf{x}_2) \right\rangle = \frac{2}{\|\mathbf{w}\|}$, lo que significa que para esta forma canónica de $(\mathbf{w}, b)$ el *margen* es $\frac{1}{\|\mathbf{w}\|}$ como se aprecia en la *Figura 1.*

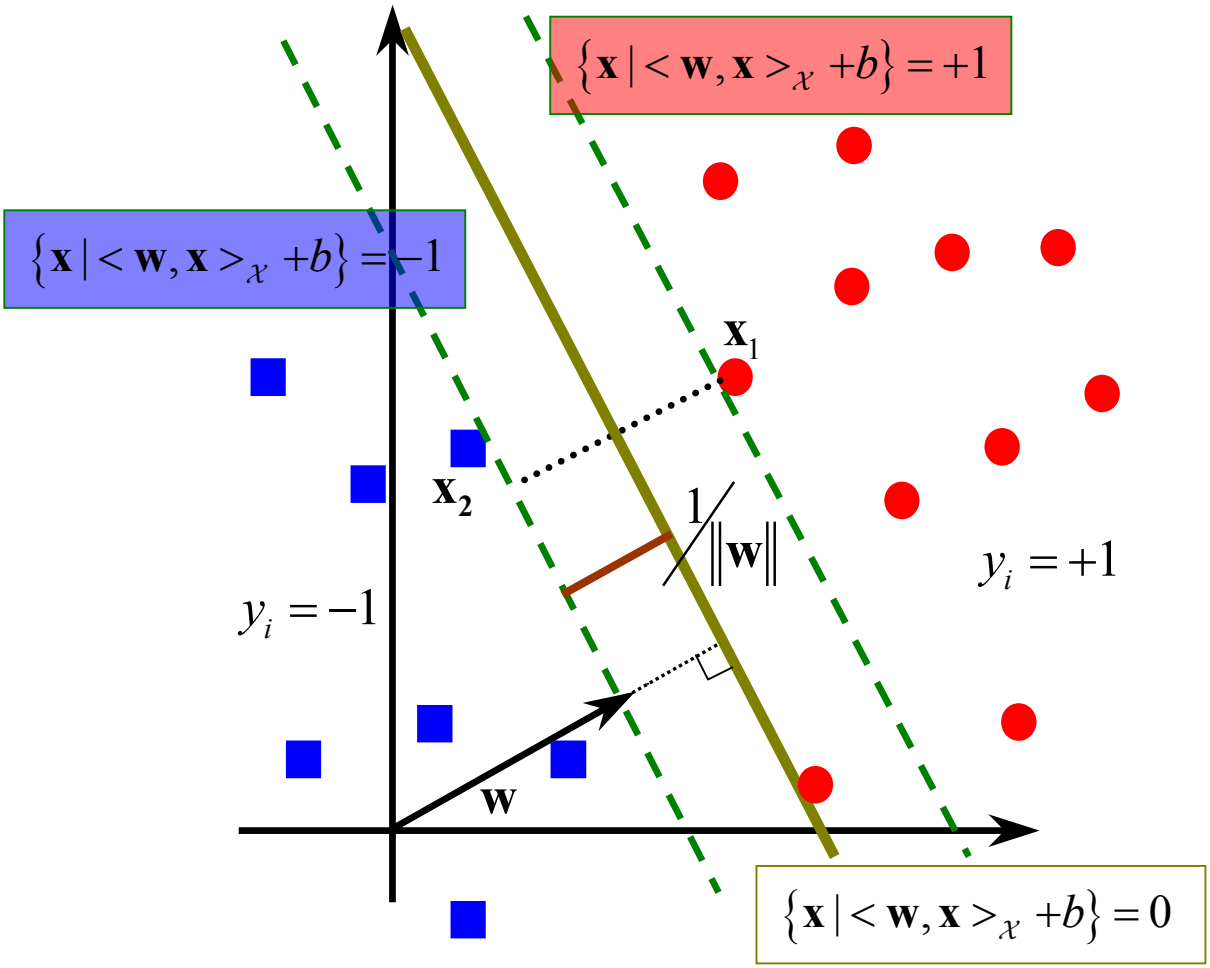

Figura 1. Hiperplano canónico y definición del margen geométrico.

Sin embargo pueden existir diferentes hiperplanos que solucionan el problema, pero unos son mejores que otros de acuerdo su margen, a mayor margen el hiperplano es mejor, como se aprecia en la *Figura 2.*

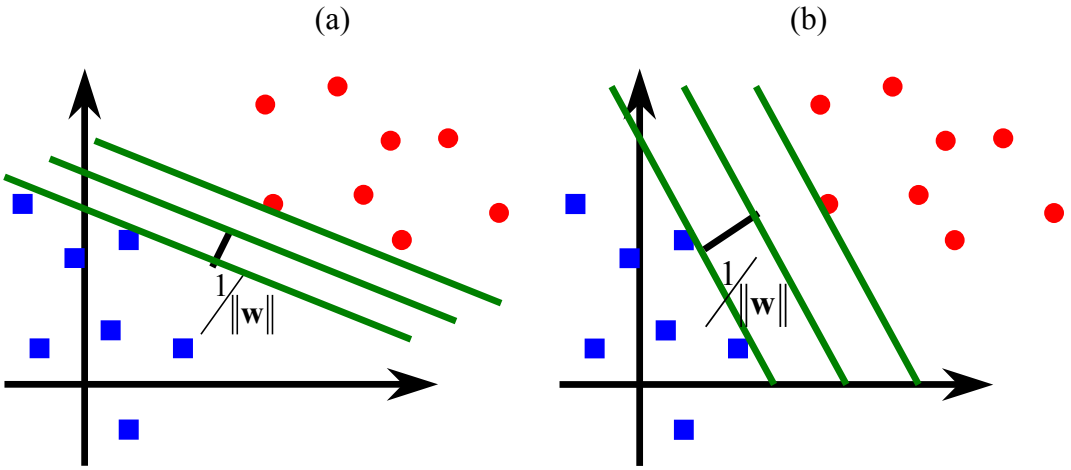

Figura 2. Problema linealmente separable y un posible hiperplano separador. a) hiperplano separador con bajo margen. b) hiperplano separador con alto margen.

Entonces para un conjunto de ejemplos etiquetados $\mathcal{S} = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{m}$ linealmente separables, se puede encontrar un clasificador lineal con máximo margen y que no cometa errores resolviendo el siguiente problema de optimización con restricciones:

$$\begin{aligned} &\min_{\mathbf{w}\in\mathcal{X}, b\in\mathbb{R}} \quad \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ &s.a. \qquad y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{X}} + b\right) \geq 1 \qquad para \quad i = 1,\ldots,m \end{aligned} \tag{1.5}$$

Para resolver este problema se introducen *multiplicadores de Lagrange* $\alpha_i \geq 0$ y el Lagrangiano:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{X}} + b \right) - 1 \right) \tag{1.6}$$

A partir del Lagrangiano, se pueden minimizar las variables primales $\mathbf{w}$ y b o maximizar las variables duales α_i.

Por otra parte las condiciones de Karush-Kuhn-Tucker (KKT) implican que:

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \tag{1.7}$$

y

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \tag{1.8}$$

Aplicando (1.7) y (1.8) a (1.6) se obtiene:

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{1.9}$$

y

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{1.10}$$

Sustituyendo (1.10) en (1.6) se tiene:

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\left\langle \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i, \sum_{j=1}^{m} \alpha_j y_j \mathbf{x}_j \right\rangle_{\mathcal{X}} - \sum_{i=1}^{m} \alpha_i \left(y_i \left(\left\langle \sum_{j=1}^{m} \alpha_j y_j \mathbf{x}_j, \mathbf{x}_i \right\rangle_{\mathcal{X}} + b \right) - 1 \right) \\ &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} - b \sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} - b \sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i \end{aligned}$$

Utilizando (1.9) se llega a

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} \tag{1.11}$$

De (1.11) el planteamiento del problema dual es

$$\begin{aligned} &\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \; W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} \\ &s.a. \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \\ &\qquad \alpha_i \geq 0 \qquad\qquad para \quad i = 1, ..., m \end{aligned} \tag{1.12}$$

Teniendo en cuenta que $\max_x f(x)$ es equivalente $\min_x -f(x)$ y reformulando el problema (1.12) en forma matricial, el nuevo problema de optimización es:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^m} \quad & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} \\ s.a. \quad & \mathbf{y}^T\boldsymbol{\alpha} = 0 \\ & \alpha_i \geq 0 \qquad\qquad para \quad i = 1,...,m \end{aligned} \tag{1.13}$$

donde $Q_{ij} = y_i y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}}$

Puede notarse que (1.13) es un problema de programación cuadrática, pues la función objetivo es cuadrática y tiene restricciones lineales.

A partir de (1.10) el hiperplano separador óptimo puede escribirse como

$$h_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i \left\langle \mathbf{x}, \mathbf{x}_i \right\rangle_{\mathcal{X}} + b \right) \tag{1.14}$$

Es de destacarse que este hiperplano está en función de los datos con los que fue entrenado, sin embargo la mayoría de los α_i son cero, dado que muy pocas restricciones de (1.5) están activas, con lo cual solo es función de los datos para los cuales los α_i son diferentes de cero, éstos son llamados *vectores de soporte.*

Para hallar el valor del umbral b, teniendo en cuenta que del Lagrangiano se sabe que $\alpha_j \left(y_j \left(\left\langle \mathbf{w}, \mathbf{x}_j \right\rangle_{\mathcal{X}} + b \right) - 1 \right) = 0$ para todo $i = 1,...,m$ se tiene que, utilizando (1.10) para los $\alpha_j > 0$

$$\sum_{i=1}^{m} \alpha_i y_i \left\langle \mathbf{x}_j, \mathbf{x}_i \right\rangle_{\mathcal{X}} + b = y_j \tag{1.15}$$

De (1.15), el umbral puede ser obtenido promediando

$$b = y_j - \sum_{i=1}^{m} \alpha_i y_i \left\langle \mathbf{x}_j, \mathbf{x}_i \right\rangle_{\mathcal{X}} \tag{1.16}$$

sobre todos los puntos con $\alpha_j > 0$, es decir, los vectores de soporte.

1.2.2 CASO NO SEPARABLE

Cuando el problema no es linealmente separable, es interesante encontrar un clasificador con el mínimo error empírico sobre el conjunto de ejemplos etiquetados. Sin embargo el problema de

encontrar este clasificador es NP-Hard **[4]**. Sin embargo este problema puede ser aliviado teniendo en cuenta el concepto de *margen* pasando por alto los puntos con algún margen positivo en el hiperplano, en este caso el problema tiene complejidad polinomial.

Por las razones expuestas anteriormente para el caso no separable se puede plantear el siguiente problema de optimización **[3]**,**[5]**:

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{X},\boldsymbol{\xi}\in\mathbb{R}^m,b\in\mathbb{R}} \quad & \tau(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i \\ s.a. \quad & y_i\left(\langle \mathbf{w},\mathbf{x}_i\rangle_{\mathcal{X}} + b\right) \geq 1-\xi_i \\ & \xi_i \geq 0 \qquad\qquad para \quad i=1,...,m \end{aligned} \tag{1.17}$$

Donde $C \geq 0$ es una constante que controla el "trade-off" entre dos objetivos que entran en conflicto:

1. Minimizar el error de entrenamiento.
2. Maximizar el margen.

En este caso cuando $\xi_i = 0$, no existe un margen de error, sin embargo $\xi_i > 0$ implica que las clases se sobrelapan.

Desafortunadamente, no se tiene una forma "a priori" de seleccionar el parámetro C, aunque comúnmente se utiliza $C/m = 10$.

El procedimiento para resolver (1.17) es similar al descrito para resolver (1.5) como se presenta a continuación.

Se introducen *multiplicadores de Lagrange* $\alpha_i, \beta_i \geq 0$ y el Lagrangiano:

$$L(\mathbf{w},\boldsymbol{\xi},b,\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(y_i\left(\langle \mathbf{w},\mathbf{x}_i\rangle_{\mathcal{X}} + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{m}\beta_i\xi_i \tag{1.18}$$

A partir del Lagrangiano, se pueden minimizar las variables primales $\mathbf{w},\boldsymbol{\xi}$ y b o maximizar las variables duales $\boldsymbol{\alpha},\boldsymbol{\beta}$.

Por otra parte las condiciones de Karush-Kuhn-Tucker (KKT) implican que:

$$\frac{\partial}{\partial b} L\left(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = 0 \tag{1.19}$$

$$\frac{\partial}{\partial \boldsymbol{\xi}} L\left(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = 0 \tag{1.20}$$

y

$$\frac{\partial}{\partial \mathbf{w}} L\left(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = 0 \tag{1.21}$$

Aplicando (1.19),(1.20) y (1.21) a (1.18) se obtiene:

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{1.22}$$

$$\alpha_i + \beta_i = C/m \tag{1.23}$$

y

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{1.24}$$

Ahora, manipulando algebraicamente (1.18) se tiene

$$L\left(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i y_i \left\langle \mathbf{w}, \mathbf{x}_i \right\rangle_{\mathcal{X}} - b\sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \alpha_i \xi_i - \sum_{i=1}^{m} \beta_i \xi_i$$

$$L\left(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = \frac{1}{2}\left\langle \mathbf{w}, \mathbf{w} \right\rangle_{\mathcal{X}} - \sum_{i=1}^{m} \alpha_i y_i \left\langle \mathbf{w}, \mathbf{x}_i \right\rangle_{\mathcal{X}} + \sum_{i=1}^{m} \alpha_i - b\sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \xi_i \left(C/m - \alpha_i - \beta_i\right) \tag{1.25}$$

Utilizando (1.22) y (1.23) en (1.25)

$$L\left(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = \frac{1}{2}\left\langle \mathbf{w}, \mathbf{w} \right\rangle_{\mathcal{X}} - \sum_{i=1}^{m} \alpha_i y_i \left\langle \mathbf{w}, \mathbf{x}_i \right\rangle_{\mathcal{X}} + \sum_{i=1}^{m} \alpha_i \tag{1.26}$$

Reemplazando (1.24) en (1.26)

$$L\left(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} \tag{1.27}$$

Por otra parte $\alpha_i, \beta_i \geq 0$ y combinado con (1.23) se obtiene

$$0 \le \alpha_i \le C/m \qquad para \quad i = 1,...,m \tag{1.28}$$

Finalmente de (1.27) y con las restricciones (1.22) y (1.28) el planteamiento del problema dual es

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; & W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} \\ s.a. \; & \sum_{i=1}^{m} \alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C/m \qquad para \quad i = 1,...,m \end{aligned} \tag{1.29}$$

Reformulando el problema (1.29) en forma matricial y en términos de minimización, el nuevo problema de optimización es:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ s.a. \; & \mathbf{y}^T \boldsymbol{\alpha} = 0 \\ & 0 \le \alpha_i \le C/m \qquad para \quad i = 1,...,m \end{aligned} \tag{1.30}$$

donde $Q_{ij} = y_i y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}}$

Puede notarse que (1.30) es bastante similar a (1.13) con excepción de las restricciones sobre α_i, por otra parte el hiperplano separador óptimo $h_{\mathbf{w},b}(\mathbf{x})$ conserva la misma forma de (1.14).

Para calcular el umbral b, a partir de la primera restricción de (1.17) para los vectores de soporte $\mathbf{x}_j$ para los cuales $\xi_j = 0$, se tiene la misma condición que para el caso separable (1.15). Entonces el umbral b puede ser obtenido promediando (1.16), como en el caso no separable, sobre todos los $\mathbf{x}_j$ vectores de soporte con $\alpha_j < C/m$, en otras palabras, los $\mathbf{x}_j$ en donde la restricción $0 \le \alpha_j \le C/m$ del problema (1.30) no esté activa $\left(0 < \alpha_j < C/m\right)$.

1.3 SUPPORT VECTOR MACHINES

A pesar que el clasificador lineal es uno de los clasificadores más simples, usualmente el error empírico y de generalización de este tipo de clasificadores es bastante pobre.

En la *Figura 3.* se pueden apreciar ejemplos simples en los que un clasificador lineal es inapropiado para conseguir un buen error empírico y de generalización.

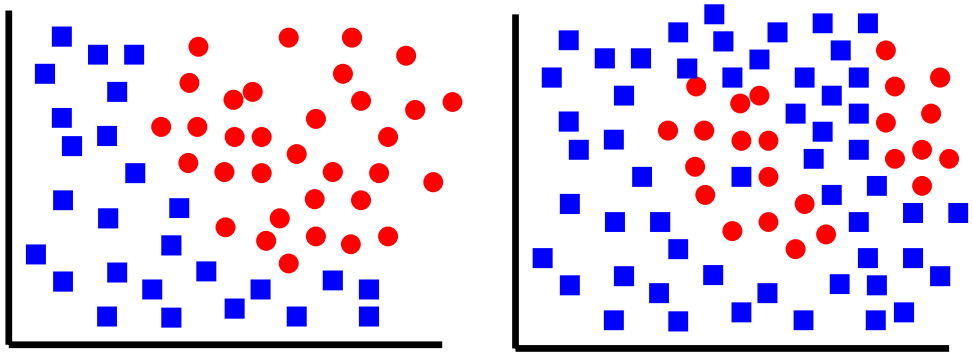

Figura 3. Ejemplos de problemas simples donde un clasificador lineal no tiene un buen desempeño.

Es por esto que el objetivo de Support Vector Machines es encontrar un hiperplano separador *óptimo* en el *espacio de características* $\mathcal{X}'$.

Este espacio de características no es otra cosa que una transformación no lineal $\Phi(x)$ de altas dimensiones del espacio de entrada $\mathcal{X}$ (espacio original de los datos) que busca que los datos sean separables.

$$\begin{aligned} \Phi : \mathcal{X} &\to \mathcal{X}' \\ x &\mapsto \mathbf{x} := \Phi(x) \end{aligned} \tag{1.31}$$

Un ejemplo ilustrativo de una transformación no lineal de este tipo se puede apreciar en la *Figura 4.*

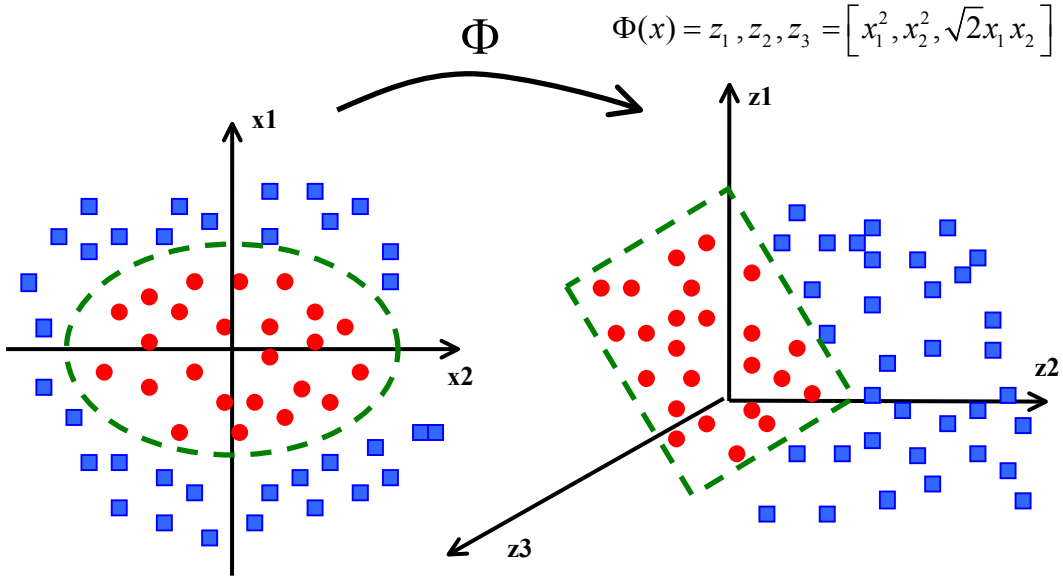

Figura 4. Ejemplo de transformación no lineal

Debido a esto, ahora el hiperplano separador óptimo $h_{\mathbf{w},b}(x)$ para los casos separable y no separable planteados en los problemas de optimización (1.5) y (1.17) dado por (1.14) está en el espacio de características $\mathcal{X}'$ y se transforma en

$$h_{\mathbf{w},b}(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i \left\langle \Phi(x), \Phi(x_i) \right\rangle_{\mathcal{X}'} + b \right) \tag{1.32}$$

Donde $\mathbf{w} \in \mathcal{X}', b \in \mathbb{R}$ y $x_i \in \mathcal{X}$.

Nótese que se hace necesario aplicar la transformación no lineal a los elementos x_i y que adicionalmente es necesario calcular el producto punto en el espacio de características $\mathcal{X}'$.

Es decir que el procedimiento a seguir estaría representado en la *Figura 5*.

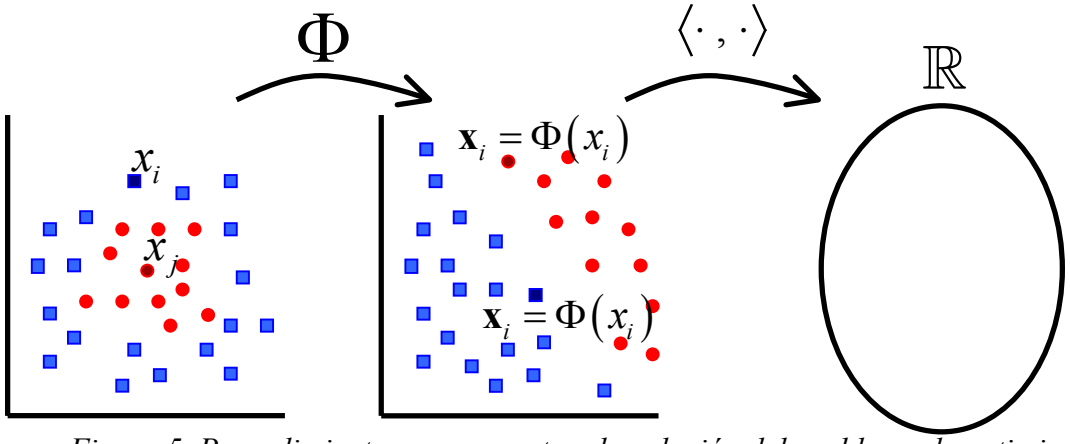

Figura 5. Procedimiento para encontrar la solución del problema de optimización.

Como puede observarse es un procedimiento bastante costoso computacionalmente, puesto que en primer lugar se debe calcular $\mathbf{\Phi}(x)$, una transformación no lineal y de altas dimensiones para luego calcular el producto punto.

Sin embargo gracias a funciones denominadas *kernels* se puede calcular el producto punto en el espacio de características directamente desde el espacio de entrada sin tener que calcular explícitamente el mapeo $\mathbf{\Phi}$ de la siguiente forma:

$$k(x_i, x_j) = \left\langle \mathbf{\Phi}(x_i), \mathbf{\Phi}(x_j) \right\rangle \tag{1.33}$$

Una de las ventajas de trabajar con estas funciones es que el espacio de entrada $\mathcal{X}$ no necesita una estructura (por ejemplo ser subconjunto de $\mathbb{R}^N$), simplemente debe ser un conjunto no vacío.

Esto se adapta perfectamente a muchas situaciones en donde no se tiene una representación vectorial y se trabaja con distancias entre parejas o similaridades entre objetos no vectoriales **[6]**,**[7]**.

Así, a partir de (1.33), el procedimiento de la *Figura 5.* se simplifica, como se aprecia en la *Figura 6.*

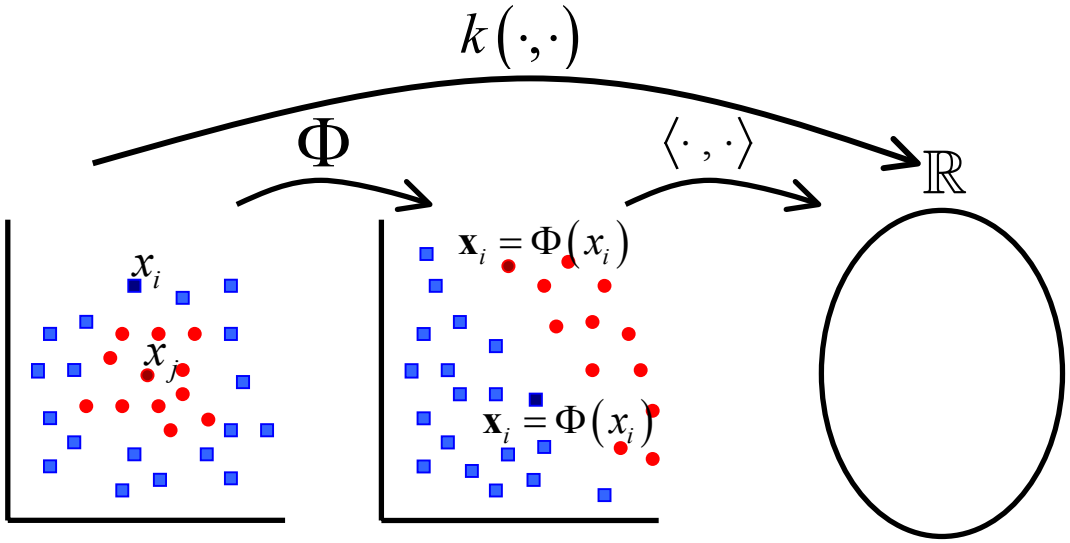

Figura 6. Procedimiento para encontrar la solución del problema de optimización utilizando kernels

Estas funciones kernel es posible calcularlas eficientemente, a continuación el ejemplo de la *Figura 3.*:

$$\begin{aligned} &\mathbf{x} = [x_1, x_2] \\ &\Phi(\mathbf{x}) = [z_1, z_2, z_3] = \left[x_1^2, x_2^2, \sqrt{2}x_1x_2\right] \\ &\left\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \right\rangle = k(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle^2 \end{aligned} \tag{1.34}$$

Es necesario presentar algunas definiciones con respecto a los kernel y algunas propiedades que ellos cumplen **[8]**.

Para i, j con valores de $1...m$ dada una función $k : \mathcal{X}^2 \to \mathbb{K}$ donde $\mathbb{K} = \mathbb{C}$ o $\mathbb{K} = \mathbb{R}$ y los patrones $x_1, ..., x_m \in \mathcal{X}$, la matriz K de $m \times m$ con elementos

$$K_{ij} := k(x_i, x_j) \tag{1.35}$$

es llamada la matriz de *Matriz de Gram* de k con respecto a $x_1, ..., x_m$.

Adicionalmente una matriz compleja K de $m \times m$ que satisface

$$\sum c_i \overline{c}_j K_{ij} \geq 0 \tag{1.36}$$

para todo $c_i \in \mathbb{C}$ es llamada positiva definida. De manera similar una matriz real K de $m \times m$ que satisface (1.36) para todo $c_i \in \mathbb{R}$ es llamada positiva definida.

En el caso particular de matrices simétricas, ésta es positiva definida sí y solo sí todos sus valores propios son no negativos.

Para un conjunto $\mathcal{X}$, un *kernel positivo definido* es una función k en $\mathcal{X} \times \mathcal{X}$ que para todo $m \in \mathbb{N}$ y todo $x_1, ..., x_m \in \mathcal{X}$ induce una matriz de Gram positiva definida.

Si k es un kernel positivo definido y $x_1, x_2 \in \mathcal{X}$, entonces se cumple la inecuación de Cauchy-Schwarz

$$\left|k\left(x_1, x_2\right)\right|^2 \leq k\left(x_1, x_1\right) \cdot k\left(x_2, x_2\right) \tag{1.37}$$

1.3.1 MAPEO DE KERNEL REPRODUCIDO

A continuación se mostrará, cómo la selección de un kernel positivo definido, define un espacio de características. Para ello se asume que k es un kernel positivo definido de valores reales y $\mathcal{X}$ un conjunto no vacío.

Se define un mapeo de $\mathcal{X}$ al espacio de funciones que mapea $\mathcal{X}$ en $\mathbb{R}$, denotado como $\mathbb{R}^{\mathcal{X}} := \left\{f : \mathcal{X} \to \mathbb{R}\right\}$ así:

$$\begin{aligned} \boldsymbol{\Phi} : \mathcal{X} &\to \mathbb{R}^{\mathcal{X}} \\ x &\mapsto k\left(\cdot, x\right) \end{aligned} \tag{1.38}$$

Donde $\boldsymbol{\Phi}\left(x\right)$ denota la función que asigna el valor de $k\left(x', x\right)$ a los $x' \in \mathcal{X}$

Entonces para construir un espacio de características $\mathcal{X}'$ asociado a $\boldsymbol{\Phi}\left(x\right)$ se debe construir un espacio con producto punto que contenga las imágenes de los parámetros de entrada bajo $\boldsymbol{\Phi}$, para ello se necesita definir un espacio vectorial tomando combinaciones lineales de la forma

$$f\left(.\right) = \sum_{i=1}^{m} \alpha_i k\left(., x_i\right) \tag{1.39}$$

Donde $m \in \mathbb{N}, \alpha_i \in \mathbb{R}$ y $x_1, ..., x_m \in \mathcal{X}$ son arbitrarios. Luego se define un producto punto entre f y otra función

$$g(.) = \sum_{j=1}^{m'} \beta_j k\left(., x_j^{'}\right) \tag{1.40}$$

Donde $m' \in \mathbb{N}, \beta_i \in \mathbb{R}$ y $x_1^{'}, ..., x_{m'}^{'} \in \mathcal{X}$

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k\left(x_i, x_j^{'}\right) \tag{1.41}$$

Ahora se verifica que el operador definido $\langle \cdot, \cdot \rangle$ cumple con las propiedades del producto punto.

Usando (1.39) y (1.41) se tiene que

$$\langle f, g \rangle = \sum_{j=1}^{m'} \beta_j f\left(x_j^{'}\right) \tag{1.42}$$

y usando (1.40) y (1.41) adicionando el hecho que $k\left(x_j^{'}, x_i\right) = k\left(x_i, x_j^{'}\right)$ se tiene que

$$\langle f, g \rangle = \sum_{i=1}^{m} \alpha_i g\left(x_i\right) \tag{1.43}$$

Entonces (1.42) y (1.43) muestran que $\langle \cdot, \cdot \rangle$ es bilineal.

De (1.41) es evidente que $\langle \cdot, \cdot \rangle$ es simétrico puesto que $\langle f, g \rangle = \langle g, f \rangle$.

Por otra parte partiendo que k es positivo definido y para cualquier función f escrita como (1.39) se tiene que

$$\langle f, f \rangle = \sum_{i,j=1}^{m} \alpha_i \alpha_j k\left(x_i, x_j\right) \geq 0 \tag{1.44}$$

Con lo cual $\langle \cdot, \cdot \rangle$ es positivo definido y más allá de eso $\langle \cdot, \cdot \rangle$ es un kernel en el espacio de funciones $\mathbb{R}^{\mathcal{X}}$.

$\langle \cdot, \cdot \rangle$ no solamente es definido positivo, sino estrictamente definido positivo puesto que si $\langle f, f \rangle = 0$ entonces $f = 0$.

Una propiedad interesante es la *propiedad de reproducción* puesto que usando (1.39) se tiene que

$$\langle k(.,x), f \rangle = f(x) \tag{1.45}$$

En particular si $f(x') = k(.,x')$, (1.45) se convierte en

$$\langle k(.,x), k(.,x') \rangle = k(x,x') \tag{1.46}$$

Así mismo de acuerdo a la definición del mapeo (1.38)

$$\langle k(.,x), k(.,x') \rangle = \langle \mathbf{\Phi}(x), \mathbf{\Phi}(x') \rangle \tag{1.47}$$

Con lo cual de (1.46) y (1.47)

$$\langle \mathbf{\Phi}(x), \mathbf{\Phi}(x') \rangle = k(x,x') \tag{1.48}$$

El espacio con producto punto $\mathcal{X}'$ construido es uno de los posibles espacios de características asociados al kernel k.

Ahora se considera como a partir de un espacio de características se define un kernel k positivo definido.

Para ello se tiene un mapeo $\mathbf{\Phi}$ de $\mathcal{X}$ a un espacio con producto punto, de allí se obtiene un kernel positivo definido, definiendo éste como

$$k(x,x') := \langle \mathbf{\Phi}(x), \mathbf{\Phi}(x') \rangle \tag{1.49}$$

Puesto que para todo $c_i \in \mathbb{R}$, $x_i \in \mathcal{X}$ con $i = 1,...,m$ se tiene por (1.49)

$$\sum_{i,j} c_i c_j k(x_i, x_j) = \sum_{i,j} c_i c_j \langle \mathbf{\Phi}(x_i), \mathbf{\Phi}(x_j) \rangle \tag{1.50}$$

De (1.50) y por bilinealidad del producto punto

$$\sum_{i,j} c_i c_j k(x_i, x_j) = \left\langle \sum_i c_i \mathbf{\Phi}(x_i), \sum_j c_j \mathbf{\Phi}(x_j) \right\rangle \tag{1.51}$$

Finalmente de (1.51) por la no negatividad de la norma

$$\sum_{i,j} c_i c_j k\left(x_i, x_j\right) = \left\| \sum_i c_i \mathbf{\Phi}\left(x_i\right) \right\|^2 \geq 0 \tag{1.52}$$

Demostrando que el kernel k es positivo definido.

De todo esto sale una definición equivalente de kernel positivo definido como la función para la cual existe un mapeo $\mathbf{\Phi}$ en un espacio con producto punto tal que se cumple (1.48).

Para el caso en el que k es un kernel positivo definido de valores complejos, todas las demostraciones son análogas **[8]**.

1.3.2 TRUCO DEL KERNEL

Dado un algoritmo formulado en términos de un kernel positivo definido k, se puede construir un algoritmo alternativo reemplazando k por otro kernel positivo definido $\tilde{k}$.

1.3.3 ESPACIOS DE HILBERT DE KERNEL REPRODUCIDO

Hasta el momento, el espacio de características asociado a un kernel dado es un espacio vectorial dotado con un producto punto o equivalentemente, un *espacio pre-Hilbert*. Fácilmente este espacio puede ser convertido en un *espacio de Hilbert* mediante un simple truco matemático, esta estructura adicional tiene algunas otras ventajas matemáticas útiles en espacios de dimensión infinita.

Sea un espacio pre-Hilbert de funciones (1.39), dotado con el producto punto (1.41), para convertirlo en un espacio de Hilbert (sobre $\mathbb{R}$), éste se *completa* con la norma correspondiente al producto punto $\|f\| := \sqrt{\langle f, f \rangle}$ adicionando todas las secuencias de Cauchy que convergen en esa norma.

Una secuencia de Cauchy es una secuencia $\left(\mathbf{z}_i\right)_i := \left(\mathbf{z}_i\right)_{i \in \mathbb{N}} = \left(\mathbf{z}_1, \mathbf{z}_2, ...\right)$ en un espacio normado $\mathcal{X}'$ si para todo $\varepsilon > 0$, existe un $n \in \mathbb{N}$ tal que para todo $n', n'' > n$ se tenga que $\left\| \mathbf{z}_{n'} - \mathbf{z}_{n''} \right\| < \varepsilon$.

Se dice que una secuencia de Cauchy converge a un punto $\mathbf{z} \in \mathcal{X}'$ si $\|\mathbf{z}_n - \mathbf{z}\| \to 0$ cuando $n \to \infty$.

Entonces $\mathcal{X}'$ es un espacio de Hilbert de kernel reproducido (RKHS por sus siglas en inglés) si dado $\mathcal{X}$ un conjunto no vacío y $\mathcal{X}'$ un espacio de Hilbert de funciones $f : \mathcal{X} \to \mathbb{R}$ dotado con el producto punto $\langle \cdot, \cdot \rangle$ (y la norma $\|f\| := \sqrt{\langle f, f \rangle}$) existe una función $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ con las siguientes propiedades:

1. k cumple con la propiedad de reproducción

$$\langle f, k(\cdot, x) \rangle = f(x) \quad \text{para todo } f \in \mathcal{X}' \tag{1.53}$$

en particular

$$\langle k(x, \cdot), k(x', \cdot) \rangle = k(x, x') \tag{1.54}$$

2. k expande a $\mathcal{X}'$, es decir $\mathcal{X}' = \overline{\text{span}\{k(x, \cdot) \mid x \in \mathcal{X}\}}$ donde $\text{span}\{k(x, \cdot) \mid x \in \mathcal{X}\}$ hace referencia a todas las combinaciones lineales de $k(x, \cdot)$ y $\overline{X}$ es el encerramiento de X, es decir que X es completado con todas las secuencias de Cauchy.

Es importante destacar que el RKHS determina un único k. Esto se demuestra por contradicción suponiendo que existen dos kernel k y k' que expanden el mismo RKHS $\mathcal{X}'$. Utilizando la simetría del producto punto y la propiedad de reproducción se tiene que

$$\langle k(x, \cdot), k'(x', \cdot) \rangle_{\mathcal{X}} = k(x, x') = k'(x', x) \tag{1.55}$$

Con lo cual se demuestra que son el mismo.

1.3.4 MAPEO DEL KERNEL DE MERCER

Ahora se construirá otro espacio de Hilbert alternativo al RKHS, y aunque se puede argumentar que esto es superfluo dado que dos espacios de Hilbert separables son isométricamente isomorfos, es decir, que es posible definir un mapeo lineal uno a uno entre los espacios que preserva el producto punto, el Teorema de Mercer **[9]** utilizado en esta construcción, ha jugado

un papel fundamental en el entendimiento de SVM, provee información crucial sobre la geometría del espacio de características y en la literatura es generalmente utilizado para introducir el truco del kernel.

Para introducir el teorema de Mercer se asume que $(\mathcal{X},\mu)$ es un *espacio de medida finita* (un conjunto $\mathcal{X}$ con una σ-álgebra definida en éste, en donde se define una medida μ que satisface $\mu(\mathcal{X})<\infty$, note que escalando esta medida por un factor adecuado, μ es una medida de probabilidad). El término casi todos significa excepto para los conjuntos con medida cero.

Teorema de Mercer [9]: Suponiendo que $k\in L_\infty(\mathcal{X}^2)$ es una función simétrica de valores reales tal que el operador integral

$$\begin{aligned} T_k &: L_2(\mathcal{X})\to L_2(\mathcal{X}) \\ (T_k f)(x) &:= \int_{\mathcal{X}} k(x,x')f(x')d\mu(x') \end{aligned} \tag{1.56}$$

es positivo definido, es decir que para todo $f\in L_2(\mathcal{X})$ se tiene que

$$\int_{\mathcal{X}^2} k(x,x')f(x)f(x')d\mu(x)d\mu(x')\geq 0 \tag{1.57}$$

Sean $\psi_j\in L_2(\mathcal{X})$ las funciones propias normalizadas de T_k asociadas con los valores propios $\lambda_j>0$, ordenados no decrecientemente. Entonces

1. $(\lambda_j)_j\in l_1$. Es decir que $\sum_j|\lambda_j|<\infty$

2. $k(x,x')=\sum_{j=1}^{N_{\mathcal{X}}}\lambda_j\psi_j(x)\psi_j(x')$ para casi todos los (x,x') para cualquier $N_{\mathcal{X}}\in\mathbb{N}$ o $N_{\mathcal{X}}=\infty$; en el último caso, la serie converge absoluta y uniformemente para casi todos los (x,x').

El numeral 2 es consecuencia que $k(x,x')$ corresponde al producto punto en $l_2^{N_{\mathcal{X}}}$, es decir $k(x,x')=\langle\Phi(x),\Phi(x')\rangle$ con

$$\Phi : \mathcal{X} \to \quad l_2^{N_{\mathcal{X}}}$$
$$x \mapsto \begin{bmatrix} \sqrt{\lambda_1}\psi_1(x) \\ \sqrt{\lambda_2}\psi_2(x) \\ \vdots \\ \sqrt{\lambda_{N_{\mathcal{X}}}}\psi_{N_{\mathcal{X}}}(x) \end{bmatrix} \tag{1.58}$$

para casi todos los $x \in \mathcal{X}$.

La convergencia uniforme de las series implica que dado un $\varepsilon > 0$ existe un $n \in \mathbb{N}$ tal que si $N_{\mathcal{X}} = \infty$, k puede ser aproximado con precisión ε como un producto punto en $\mathbb{R}^n$ tal que para casi todos los $x, x' \in \mathcal{X}$, $\left| k(x,x') - \left\langle \Phi^n(x), \Phi^n(x') \right\rangle \right| < \varepsilon$ donde

$$\Phi^n : \mathcal{X} \to \quad l_2^n$$
$$x \mapsto \begin{bmatrix} \sqrt{\lambda_1}\psi_1(x) \\ \sqrt{\lambda_2}\psi_2(x) \\ \vdots \\ \sqrt{\lambda_n}\psi_n(x) \end{bmatrix} \tag{1.59}$$

Entonces el espacio de características puede ser pensado siempre como un espacio finito con precisión ε **[7]**.

En consecuencia si un kernel k satisface el teorema de Mercer, se puede construir un mapeo $\mathbf{\Phi}$ en un espacio donde k actúa como un producto punto

$$\left\langle \mathbf{\Phi}(x), \mathbf{\Phi}(x') \right\rangle = k(x,x') \tag{1.60}$$

para casi todos los $x, x' \in \mathcal{X}$. Más aún, dado un $\varepsilon > 0$, existe un mapeo $\mathbf{\Phi}^n$ en un espacio con producto punto n-dimensional (donde n depende de ε) tal que

$$\left| k(x,x') - \left\langle \Phi^n(x), \Phi^n(x') \right\rangle \right| < \varepsilon \tag{1.61}$$

para casi todos los $x, x' \in \mathcal{X}$.

1.3.5 EJEMPLOS DE KERNELS

A continuación se presentan algunos ejemplos de funciones kernel

Asumiendo $\mathcal{X} \subset \mathbb{R}^N$

1. Polinomial

$$k(x, x') = \langle x, x' \rangle^d \tag{1.62}$$

donde $d \in \mathbb{N}$

2. Gaussiano

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2} \right) \tag{1.63}$$

donde $\sigma > 0$

3. Sigmoide

$$k(x, x') = \tanh\left(\kappa \langle x, x' \rangle + \theta \right) \tag{1.64}$$

donde $\kappa > 0$ y $\theta < 0$. Aunque se destaca que este kernel no es positivo definido, es exitoso en la práctica.

4. Polinomial no homogéneo

$$k(x, x') = \left(\langle x, x' \rangle + c \right)^d \tag{1.65}$$

donde $d \in \mathbb{N}$ y $c \geq 0$

5. Curvigrafo B_n de orden impar

$$k(x, x') = B_{2p+1}\left(\|x - x'\| \right) \tag{1.66}$$

con $B_n := \bigotimes_{i=1}^{n} I_{[-1/2, 1/2]}$ en donde $I_{[\cdot]}$ es la función indicadora y $p \in \mathbb{N}$

6. Similaridad de eventos probabilísticos

Sea $(\mathcal{X}, \mathcal{C}, P)$ un espacio de probabilidad con σ-álgebra $\mathcal{C}$ y medida de probabilidad P

$$k(A,B) = P(A \cap B) - P(A)P(B) \tag{1.67}$$

1.4 HIPERPLANO SEPARADOR ÓPTIMO EN EL ESPACIO DE CARACTERÍSTICAS

Dado que a partir de las funciones kernel es posible hallar el producto punto en el espacio de características, utilizando el truco del kernel se busca encontrar un hiperplano separador óptimo en este espacio, reformulando los problemas para los casos separable y no separable.

1.4.1 CASO SEPARABLE

Anteriormente se había encontrado que el hiperplano separador óptimo estaba determinado por (1.14), ahora en el espacio de características está dado por

$$h_{\mathbf{w},b}(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i \left\langle \Phi(x), \Phi(x_i) \right\rangle_{\mathcal{X}'} + b \right) \tag{1.68}$$

Pero gracias a los kernels, (1.68) se convierte en

$$h_{\mathbf{w},b}(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b \right) \tag{1.69}$$

Así mismo el problema dual de optimización para el caso no separable que era dado por (1.12) se convierte en

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \; & W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j k(x_i, x_j) \\ s.a. \; & \sum_{i=1}^{m} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \qquad para \quad i = 1, ..., m \end{aligned} \tag{1.70}$$

Ahora (1.70) en términos de minimización y en forma matricial (análogo a (1.13)) es

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \; & f(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ s.a. \; & \mathbf{y}^T \boldsymbol{\alpha} = 0 \\ & \alpha_i \geq 0 \qquad para \quad i = 1, ..., m \end{aligned} \tag{1.71}$$

donde $Q_{ij} = y_i y_j k\left(x_i, x_j\right)$.

Así mismo para hallar el umbral b, de forma análoga a (1.16), ahora se promedia

$$b = y_j - \sum_{i=1}^{m} \alpha_i y_i k\left(x_i, x_j\right) \tag{1.72}$$

para todos los puntos con $\alpha_j > 0$.

1.4.2 CASO NO SEPARABLE

A continuación se presentarán los cambios para el caso no separable a la hora de encontrar el hiperplano separador óptimo en el espacio de características. Existen dos alternativas, la primera, explicada anteriormente, conocida como C-SVM, además de un planteamiento alternativo al problema de optimización conocido como ν-SVM.

1.4.2.1 C-SVM

Para el caso no separable, el problema de optimización estaba planteado por (1.17), cuyo hiperplano separador óptimo en el espacio de entrada estaba dado, al igual que en el caso separable, por (1.14) . De la misma forma que en caso separable, utilizando el truco del kernel, en el espacio de características el hiperplano separador óptimo estará dado por (1.69).

Así mismo el problema dual planteado en (1.29) pero en el espacio de características se convierte en

$$\begin{aligned} &\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \; W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j k\left(x_i, x_j\right) \\ &s.a. \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \\ &\qquad 0 \le \alpha_i \le C/m \qquad\qquad para \quad i = 1, \ldots, m \end{aligned} \tag{1.73}$$

Ahora (1.73) en términos de minimización y en forma matricial (análogo a (1.30))se expresa como

$$\begin{aligned} &\min_{\boldsymbol{\alpha}\in\mathbb{R}^m} \quad f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} \\ &s.a. \quad \mathbf{y}^T\boldsymbol{\alpha} = 0 \\ &\qquad 0 \le \alpha_i \le {}^{C}\!/_{m} \qquad\qquad para \quad i = 1,...,m \end{aligned} \tag{1.74}$$

donde $Q_{ij} = y_i y_j k(x_i, x_j)$.

Adicionalmente, para calcular el umbral b, tal como se hacía en el espacio de entrada promediando (1.16), en el espacio de características se obtiene promediando (1.72) sobre los puntos con $0 < \alpha_i < {}^{C}\!/_{m}$, es decir donde las restricciones sobre α no están activas.

1.4.2.2 ν-SVM

Existe una modificación al problema de optimización propuesta en **[10]** en donde es reemplazado C por el parámetro ν que adicionalmente controla el número de *errores de margen* y de *vectores de soporte*, este nuevo problema es

$$\begin{aligned} &\min_{\mathbf{w}\in\mathcal{X},\boldsymbol{\xi}\in\mathbb{R}^m,\rho,b\in\mathbb{R}} \quad \tau(\mathbf{w},\boldsymbol{\xi},\rho) = \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\xi_i \\ &s.a. \qquad\quad y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{X}} + b\right) \ge \rho - \xi_i \\ &\qquad\quad \xi_i \ge 0 \qquad\qquad para \quad i = 1,...,m \\ &\qquad\quad \rho \ge 0 \end{aligned} \tag{1.75}$$

Se destaca que es introducido un nuevo parámetro ρ para ser optimizado, éste controla el margen de separación entre las clases, puesto que si $\boldsymbol{\xi} = 0$, la primera restricción de (1.75) establece que el margen es ${}^{2\rho}\!/_{\|\mathbf{w}\|}$.

Nuevamente para resolver (1.75) se recurre a hallar el dual

$$\begin{aligned} L(\mathbf{w},\boldsymbol{\xi},b,\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\delta) = &\frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\xi_i \\ &- \sum_{i=1}^{m}\alpha_i\left(y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{X}} + b\right) - \rho + \xi_i\right) - \sum_{i=1}^{m}\beta_i\xi_i - \delta\rho \end{aligned} \tag{1.76}$$

donde $\alpha_i \beta_i, \delta \geq 0$ son los multiplicadores de Lagrange de las restricciones.

A partir de las condiciones de KKT para $b, \boldsymbol{\xi}, \mathbf{w}$ y ρ se obtiene

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{1.77}$$

$$\alpha_i + \beta_i = 1/m \tag{1.78}$$

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{1.79}$$

y

$$\sum_{i=1}^{m} \alpha_i - \delta = \nu \tag{1.80}$$

Manipulando algebraicamente (1.76)

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\xi}, b, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta) &= \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{X}} - b\sum_{i=1}^{m} \alpha_i y_i \\ &\quad + \rho\sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \alpha_i \xi_i - \sum_{i=1}^{m} \beta_i \xi_i - \delta\rho \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{X}} - b\sum_{i=1}^{m} \alpha_i y_i \\ &\quad + \rho\left(\sum_{i=1}^{m} \alpha_i - \delta\right) - \sum_{i=1}^{m} (\alpha_i + \beta_i)\xi_i \end{aligned} \tag{1.81}$$

Reemplazando (1.77), (1.78) y (1.80) en (1.81) se tiene

$$L(\mathbf{w}, \boldsymbol{\xi}, b, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{X}} \tag{1.82}$$

Sustituyendo (1.79) en (1.82) y simplificando

$$L(\mathbf{w}, \boldsymbol{\xi}, b, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle_{\mathcal{X}} \tag{1.83}$$

Como $\alpha_i, \beta_i \geq 0$ y junto con (1.78), una restricción es

$$0 \leq \alpha_i \leq 1/m \qquad para \quad i = 1, \ldots, m \tag{1.84}$$

Como $\delta \geq 0$, a partir de (1.80)

$$\sum_{i=1}^{m} \alpha_i \geq \mu \tag{1.85}$$

Con base en Lagrangiano (1.83) y en las restricciones (1.77), (1.84) y (1.85) el problema dual es

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \; & W(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} \\ s.a. \; & \sum_{i=1}^{m} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq 1/m \qquad para \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i \geq \mu \end{aligned} \tag{1.86}$$

Aplicando el truco del kernel para plantear el problema dual en el espacio de características, el planteamiento resultante es

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \; & W(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j k\left(x_i, x_j\right) \\ s.a. \; & \sum_{i=1}^{m} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq 1/m \qquad para \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i \geq \mu \end{aligned} \tag{1.87}$$

El problema (1.87) en términos de minimización y en forma matricial es

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \; & f(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^T Q \boldsymbol{\alpha} \\ s.a. \; & \mathbf{y}^T \boldsymbol{\alpha} = 0 \\ & 0 \leq \alpha_i \leq 1/m \qquad para \quad i = 1, \ldots, m \\ & \mathbf{e}^T \boldsymbol{\alpha} \geq \nu \end{aligned} \tag{1.88}$$

donde $Q_{ij} = y_i y_j k\left(x_i, x_j\right)$ y la función de decisión está determinada por (1.69).

Para hallar el umbral b y el parámetro del margen ρ, a partir de la primera restricción de (1.75) para los vectores de soporte $\mathbf{x}_j$ para los cuales $\xi_j = 0$ ($\alpha_j < 1/m$), se tiene del Lagrangiano (1.76) que $\alpha_j \left(y_j \left(\langle \mathbf{w}, \mathbf{x}_j \rangle + b \right) - \rho \right) = 0$, entonces para los casos en los que las restricciones sobre α_j no están activas, es decir para $0 < \alpha_j < 1/m$, se cumple que

$$y_j \left(\langle \mathbf{w}, \mathbf{x}_j \rangle + b \right) - \rho = 0 \tag{1.89}$$

Ahora (1.89) en el espacio de características se puede expresar como

$$y_j \left(\sum_{i=1}^{m} \alpha_i y_i k\left(x_i, x_j\right) + b \right) - \rho = 0 \tag{1.90}$$

Definiendo

$$r_1 := \rho - b \tag{1.91}$$

$$r_2 := \rho + b \tag{1.92}$$

y dos conjuntos S_+ y S_- de medidas s_+ y s_- que contienen los vectores de soporte x_j con $0 < \alpha_j < 1/m$ para $y_j = +1$ y $y_j = -1$ respectivamente, se tienen los siguientes casos.

Para cada $y_j = +1$, combinando (1.90) con (1.91) se tiene

$$\sum_{i=1}^{m} \alpha_i y_i k\left(x_i, x_j\right) - r_1 = 0 \tag{1.93}$$

A partir de (1.93) se puede obtener r_1 promediando sobre los elementos de S_+

$$r_1 = \frac{1}{s_+} \sum_{x_j \in S_+} \sum_{i=1}^{m} \alpha_i y_i k\left(x_i, x_j\right) \tag{1.94}$$

Ahora para cada $y_j = -1$, combinando (1.90) con (1.92) se tiene

$$\sum_{i=1}^{m} \alpha_i y_i k\left(x_i, x_j\right) + r_2 = 0 \tag{1.95}$$

A partir de (1.95) se puede obtener r_2 promediando sobre los elementos de S_-

$$r_2 = -\frac{1}{s_-}\sum_{x_j \in S_-}\sum_{i=1}^{m}\alpha_i y_i k\left(x_i, x_j\right) \tag{1.96}$$

A partir de las definiciones (1.91) y (1.92) se tienen que

$$b = \frac{r_2 - r_1}{2} \tag{1.97}$$

y

$$\rho = \frac{r_1 + r_2}{2} \tag{1.98}$$

Los cuales se obtienen con los valores de r_1 y r_2 hallados de (1.94) y (1.96).

Es importante destacar el papel del parámetro ν, para esto se define el *margen de error* como aquellos puntos con $\xi_i > 0$, los cuales son o errores o violaciones al margen, la *fracción de errores de margen* se puede definir entonces como

$$R_{emp}^{\rho}[g] := \frac{1}{m}\left|\left\{i \mid y_i g\left(x_i\right) < \rho\right\}\right| \tag{1.99}$$

donde g denota el argumento del sgn en la función de decisión (1.14), en otras palabras $h_{\mathbf{w},b} = \mathrm{sgn} \circ g$.

Como se mencionó en un principio, ν controla el número de *errores de margen* y de *vectores de soporte* como se demuestra en **[10]**.

1.5 SOLUCIÓN DEL PROBLEMA DE PROGRAMACIÓN CUADRÁTICA

Es claro que los problemas duales (1.74) y (1.88) son problemas de programación cuadrática con restricciones lineales mucho más sencillos de resolver que los originales puesto que únicamente están en función de los kernel. Sin embargo, no pueden ser fácilmente resueltos por técnicas tradicionales dado que involucran una matriz Hessiana de dimensión $m \times m$.

Por esta razón, para resolver (1.74) y (1.88) han surgido múltiples métodos que buscan ser más eficientes. Las técnicas de *chunking* propuestas inicialmente por Vapnik **[11]** y que se basan en

dos aspectos: 1) Remover datos de entrenamiento con $\alpha_i = 0$ no cambia la solución al problema de optimización. 2) Es más sencillo y eficiente descomponer el problema original (1.74) en subproblemas más pequeños. A partir de estos aspectos se han generado diversos métodos para solucionar el problema de optimización **[12]**,**[13]**; todos ellos con la desventaja de requerir resolver numéricamente problemas cuadráticos. Platt **[14]** propone una técnica llamada *Secuential Minimal Optimization (SMO)* en donde se descompone el problema original de tal forma que solo se requieren resolver problemas cuadráticos de dos variables de forma analítica, facilitando y acelerando la resolución del problema de optimización.

Para resolver (1.88) dado que existe una restricción lineal de desigualdad, Crisp y Burges **[15]** y Chang y Lin **[16]** han demostrado que $\mathbf{e}^T\boldsymbol{\alpha} \geq \nu$ puede ser reemplazado por $\mathbf{e}^T\boldsymbol{\alpha} = \nu$ sin que cambie la solución del problema. Gracias a esto todas las técnicas anteriormente descritas pueden ser adaptadas.

La combinación de *SMO* con técnicas de *shrinking* y *caching* **[13]** ha permitido que la complejidad computacional en la resolución de (1.74) y (1.88) esté entre cuadrática y cúbica, dependiendo del tipo de problema y dominada básicamente por el número de evaluaciones que se requieren de la función kernel **[17]**.

1.6 ALGORITMOS DE BOOSTING

1.6.1 CONCEPTOS PRELIMINARES

El *error de Bayes* R_{Bayes} está definido como el más pequeño error de generalización que puede ser alcanzado en un problema en particular.

Una hipótesis es ε-fuerte si posee un error de generalización menor que $R_{Bayes} + \varepsilon$ ($R < R_{Bayes} + \varepsilon$). De forma similar una hipótesis es débil si su error empírico es menor a $1/2 - \gamma$ ($R < 1/2 - \gamma$) para algún $\gamma > 0$.

Un algoritmo de aprendizaje $Learner(S,D)$ es un procedimiento eficiente que toma como entradas el conjunto de muestras etiquetadas S y la distribución discreta $D \in \mathbb{R}^m$ sobre S y retorna una hipótesis $h \in H$.

Un algoritmo de aprendizaje $Strong(S,D)$ es fuerte si para todo $\varepsilon > 0$ y $\delta > 0$ es capaz de retornar una hipótesis con error de generalización acotado por ε con probabilidad mayor a $1-\delta$. De forma similar, un algoritmo $Weak(S,D)$ es débil si retorna una hipótesis con error empírico menor a $1/2 - \gamma$ ($R < 1/2 - \gamma$) para algún $\gamma > 0$.

Se dice que $H(x)$ es un *clasificador combinado* cuando es una combinación lineal de varias hipótesis h_t llamadas *clasificadores base*. Es decir

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{1.100}$$

Normalizando los coeficientes α_t en (1.100) y con $\alpha_t \geq 0$ el clasificador combinado es una combinación convexa de los clasificadores base

$$H(x) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_t \alpha_t} h_t(x) \tag{1.101}$$

1.6.2 ADABOOST

Las estrategias de Boosting pretenden elevar el desempeño de un algoritmo de aprendizaje débil combinando varias hipótesis adecuadamente generando un algoritmo de aprendizaje fuerte. El algoritmo Adaboost **[18]** o Boosting adaptativo introducido en **[19]** es un meta-algoritmo (un procedimiento que usa otro procedimiento como subrutina) que toma un conjunto de muestras etiquetadas $\mathcal{S}$, una distribución discreta D y un aprendiz débil *Weak* para retornar un clasificador combinado en T iteraciones. En cada iteración t Adaboost ejecuta *Weak* sobre el conjunto $\mathcal{S}$ con la distribución D_t para obtener la hipótesis h_t.

De acuerdo al desempeño de h_t, el algoritmo modifica D_t dándole menor peso a las muestras bien clasificadas y mayor peso a las muestras mal clasificadas con el objetivo que el siguiente

clasificador se concentre en estas últimas, maximizando la cantidad de información que obtendrá en la siguiente ronda. La *Figura 7* muestra este algoritmo como fue presentado en **[18].**

Algoritmo : Adaboost$\left(S, D_1, T, Weak\right)$

Entrada : $S = \left\{x_i, y_i\right\}_{i=1}^{m}, D_1, T, Weak\left(\cdot,\cdot\right)$

Salida : $H\left(\cdot\right)$

Para $t = 1$ hasta T

Obtener una hipótesis débil usando D_t

$h_t \leftarrow Weak\left(S, D_t\right)$

Escoger adecuadamente $\alpha_t \in \mathbb{R}$. Usualmente

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1 - R_{emp}\left(h_t, S, D_t\right)}{R_{emp}\left(h_t, S, D_t\right)}\right)$$

Actualizar

$$D_{t+1}\left(i\right) = \frac{D_t\left(i\right)\exp\left(-\alpha_t y_i h_t\left(x_i\right)\right)}{Z_t}$$

Donde Z_t es un factor de normalización escogido para que D_{t+1} sea una distribución.

Fin

Salida: $H\left(x\right) = \operatorname{sgn}\left(\sum_{t=1}^{T}\frac{\alpha_t}{\sum_t \alpha_t} h_t\left(x\right)\right)$

Figura 7. Algoritmo Adaboost

2 BOOSTING SUPPORT VECTOR MACHINES

Support Vector Machines ha sido en los últimos años una técnica ampliamente aplicada a problemas de clasificación y regresión **[2]**. Desde el punto de vista de aprendizaje estadístico, una de las razones de su éxito es que para ciertas funciones kernel se ha demostrado que *SVM* es un aprendiz fuerte **[20]**, es decir que puede alcanzar un error de generalización arbitrariamente cercano al error de Bayes con un conjunto de entrenamiento lo suficientemente grande.

La principal desventaja de *SVM* es la complejidad temporal del algoritmo. Siendo *m* el número de elementos del conjunto de entrenamiento, *SVM* resuelve un problema de programación cuadrática que implica inicialmente complejidad $O(m^3)$. La aplicación de los métodos anteriormente descritos mejoran esta complejidad **[11]-[14]** llegando a que sea usualmente $O(m^2)$.

Por otra parte algoritmos como Adaboost **[21]** encuentran una buena hipótesis combinando adecuadamente hipótesis dadas por un aprendiz débil, es decir un algoritmo que retorna una hipótesis cuyo desempeño es mejor que adivinar.

Sin embargo usar un algoritmo fuerte como clasificador base de Adaboost no representa gran ventaja desde el punto de vista de la generalización. Wickramaratna, Holden y Buxton han usado *SVM* como clasificador base de Adaboost, pero el desempeño del clasificador resultante se degrada con el aumento del número de rondas **[22]**. Por esta razón puede ser útil hacer de *SVM* un algoritmo débil para aprovechar las ventajas de Adaboost.

Adicionalmente, el debilitar *SVM* trae otras ventajas ya que puede utilizarse para reducir el conjunto de entrenamiento con el fin de simplificar la representación de *SVM* **[23]**, lo que se conoce como *algoritmo de editing.*

Debido a que la complejidad de *SVM* depende intrínsecamente del número de datos de entrenamiento, se busca que en el planteamiento del problema cuadrático intervenga solo un subconjunto de los datos originales, sin que esto implique que se descarten totalmente. Razón

por la cual las estrategias de Boosting son una alternativa para combinar varias hipótesis generadas de esta forma.

Adicionalmente, entrenar con una fracción de los datos μm y combinar q hipótesis puede demorar mucho menos tiempo que entrenar con los datos completos, pues si la complejidad del algoritmo original está acotada por Am^x con $A \in \mathbb{R}$, al entrenar con la fracción μm está acotado por $A(\mu m)^x$; y combinando q hipótesis por $Aq(\mu m)^x$ (despreciando la complejidad del algoritmo que las combina). Con $x > 1,\ 0 > \mu > 1$ y $q \leq 1/\mu$ se tiene que:

$$Aq\left(\mu m\right)^x \leq \frac{A}{\mu}\left(\mu m\right)^x = A\mu^{x-1}m^x \leq Am^x \qquad (2.1)$$

De donde aunque el algoritmo resultante sigue siendo $O(m^x)$, la constante que lo acota es menor.

Por esta razón se busca utilizar *SVM* como clasificador débil de un algoritmo de Boosting similar a Adaboost, para ello se hace necesario modificar los problemas de optimización (1.74) y (1.88) para tener en cuenta las distribuciones y garantizar que *SVM* es un algoritmo débil.

2.1 SUPPORT VECTOR MACHINES CON DISTRIBUCIONES

Para el caso en el que se tenga una distribución discreta $D \in \mathbb{R}^m$ sobre el conjunto de muestras etiquetadas $\mathcal{S} = \left\{\left\langle \mathbf{x}_i, y_i \right\rangle\right\}_{i=1}^m$ se hace necesario dar mayor importancia a aquellos datos con mayor peso dentro de la distribución, en otras palabras es más grave tener un bajo margen o un margen negativo para un dato o conjunto de datos con peso considerable que para uno con bajo peso, por esta razón se deben involucrar los valores de las distribuciones en los planteamientos originales de C-SVM y ν-SVM.

Adicionalmente, algoritmos como *Boosting* hacen necesario un aprendiz débil que entrene basado en cierta distribución producto del desempeño de las hipótesis anteriores, con base en esta premisa existen dos alternativas, la primera y tal vez más sencilla en muchos casos, es utilizar muestras *boostrap* del conjunto original que reflejen la distribución con la que se debe entrenar **[24]**.

Sin embargo existen varios inconvenientes si se piensa en tener SVM como algoritmo débil. El primero hace referencia a que se debe tomar un número considerable de muestras boostrap para simular adecuadamente la distribución, esto conlleva a que el algoritmo tardará mucho más tiempo en el entrenamiento debido al crecimiento en el número de datos. El segundo, y más grave problema es que los datos que tengan más peso dentro de la distribución, estarán representados varias veces en el conjunto de muestras boostrap, esto ocasionará que la matriz de Gram (1.35) no necesariamente esté bien condicionada, ocasionando problemas en el entrenamiento.

Por esta razón la mejor alternativa es modificar los planteamientos del problema de SVM en términos de distribuciones.

2.1.1 C-SVM CON DISTRIBUCIONES

Teniendo en cuenta el problema original planteado en (1.17), éste se modifica penalizando más, en la función objetivo, aquellos términos con más peso en la distribución.

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{X},\boldsymbol{\xi}\in\mathbb{R}^m,b\in\mathbb{R}} \quad & \tau\left(\mathbf{w},\boldsymbol{\xi}\right)=\frac{1}{2}\left\|\mathbf{w}\right\|^2+C\sum_{i=1}^{m}D_i\xi_i \\ s.a. \quad & y_i\left(\left\langle \mathbf{w},\mathbf{x}_i\right\rangle_{\mathcal{X}}+b\right)\geq 1-\xi_i \\ & \xi_i\geq 0 \qquad\qquad para \quad i=1,...,m \end{aligned} \tag{2.2}$$

Donde D_i es el peso del par $\left\langle \mathbf{x}_i, y_i\right\rangle$.

Es de destacar que si la distribución es uniforme, es decir $D_i=\frac{1}{m}$ para $i=1,...m$, el problema (2.2) es equivalente a (1.17).

Se busca hallar el problema dual a (2.2) en el espacio de características. El primer paso es hallar el Lagrangiano

$$\begin{aligned} L\left(\mathbf{w},\boldsymbol{\xi},b,\boldsymbol{\alpha},\boldsymbol{\beta}\right)=&\frac{1}{2}\left\|\mathbf{w}\right\|^2+C\sum_{i=1}^{m}D_i\xi_i \\ &-\sum_{i=1}^{m}\alpha_i\left(y_i\left(\left\langle \mathbf{w},\mathbf{x}_i\right\rangle_{\mathcal{X}}+b\right)-1+\xi_i\right)-\sum_{i=1}^{m}\beta_i\xi_i \end{aligned} \tag{2.3}$$

Con los multiplicadores de Lagrange $\alpha_i, \beta_i \geq 0$.

De las condiciones de Karush-Kuhn-Tucker (KKT) para $b, \boldsymbol{\xi}$ y $\mathbf{w}$ se obtiene

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{2.4}$$

$$\alpha_i + \beta_i = C \cdot D_i \tag{2.5}$$

y

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{2.6}$$

Ahora, manipulando algebraicamente (2.3) se tiene

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle_{\mathcal{X}} - \sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{X}} + \sum_{i=1}^{m} \alpha_i - b \sum_{i=1}^{m} \alpha_i y_i \\ + \sum_{i=1}^{m} \xi_i \left(C \cdot D_i - \alpha_i - \beta_i \right) \end{aligned} \tag{2.7}$$

Utilizando (2.4) y (2.5) en (2.7)

$$L(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle_{\mathcal{X}} - \sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{X}} + \sum_{i=1}^{m} \alpha_i \tag{2.8}$$

Reemplazando (2.6) en (2.8)

$$L(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle_{\mathcal{X}} \tag{2.9}$$

Por otra parte combinado $\alpha_i, \beta_i \geq 0$ con (2.5) se obtiene

$$0 \leq \alpha_i \leq C \cdot D_i \qquad para \quad i = 1, ..., m \tag{2.10}$$

Finalmente, utilizando el truco del kernel en (2.9) y con las restricciones (2.4) y (2.10) el planteamiento del problema dual es

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; & W(\boldsymbol{\alpha}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i y_i \alpha_j y_j k\left(x_i, x_j\right) \\ s.a. \; & \sum_{i=1}^{m}\alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C \cdot D_i \qquad\qquad para \quad i = 1,...,m \end{aligned} \tag{2.11}$$

Reformulando el problema (2.11) en forma matricial y en términos de minimización, el nuevo problema de optimización es:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ s.a. \; & \mathbf{y}^T \boldsymbol{\alpha} = 0 \\ & 0 \le \alpha_i \le C \cdot D_i \qquad\qquad para \quad i = 1,...,m \end{aligned} \tag{2.12}$$

donde $Q_{ij} = y_i y_j k\left(x_i, x_j\right)$.

Es de destacar que la única diferencia entre el problema dual para distribuciones (2.12) y el problema dual original (1.73) es la restricción sobre los α_i, puesto que ahora cada uno tiene una restricción diferente, a mayor peso del dato en la distribución es mayor el valor que puede tomar α_i.

Una forma alternativa de escribir (2.12) es

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ s.a. \; & \mathbf{y}^T \boldsymbol{\alpha} = 0 \\ & 0 \le \alpha_i \le {}^{C}\!/_{m} \cdot mD_i \qquad\qquad para \quad i = 1,...,m \end{aligned} \tag{2.13}$$

Para poder tomar empíricamente el valor de ${}^{C}\!/_{m} = 10$.

El hiperplano separador óptimo $h_{\mathbf{w},b}(\mathbf{x})$ sigue conservando la forma de (1.69) y el umbral b se calcula de la misma forma que para C-SVM promediando (1.72) para los x_j para los cuales $0 < \alpha_j < {}^{C}\!/_{m} \cdot mD_j$.

2.1.2 ν-SVM CON DISTRIBUCIONES

De forma análoga a C-SVM con distribuciones, en el problema original (1.75) se penaliza cada violación del margen de acuerdo al peso del dato en la distribución

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{H},\boldsymbol{\xi}\in\mathbb{R}^m,\rho,b\in\mathbb{R}} \quad & \tau(\mathbf{w},\boldsymbol{\xi},\rho)=\frac{1}{2}\|\mathbf{w}\|^2-\nu\rho+\sum_{i=1}^{m}D_i\xi_i \\ s.a. \quad & y_i\left(\langle\mathbf{w},\mathbf{x}_i\rangle_{\mathcal{X}}+b\right)\geq\rho-\xi_i \\ & \xi_i\geq 0 \qquad\qquad para \quad i=1,\ldots,m \\ & \rho\geq 0 \end{aligned} \tag{2.14}$$

El Lagrangiano de (2.14) es

$$\begin{aligned} L(\mathbf{w},\boldsymbol{\xi},b,\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\delta)=&\frac{1}{2}\|\mathbf{w}\|^2-\nu\rho+\sum_{i=1}^{m}D_i\xi_i \\ &-\sum_{i=1}^{m}\alpha_i\left(y_i\left(\langle\mathbf{w},\mathbf{x}_i\rangle_{\mathcal{X}}+b\right)-\rho+\xi_i\right)-\sum_{i=1}^{m}\beta_i\xi_i-\delta\rho \end{aligned} \tag{2.15}$$

donde $\alpha_i\beta_i,\delta\geq 0$ son los multiplicadores de Lagrange de las restricciones.

A partir de las condiciones de KKT para $b,\boldsymbol{\xi},\mathbf{w}$ y ρ se obtiene

$$\sum_{i=1}^{m}\alpha_i y_i=0 \tag{2.16}$$

$$\alpha_i+\beta_i=D_i \tag{2.17}$$

$$\mathbf{w}=\sum_{i=1}^{m}\alpha_i y_i\mathbf{x}_i \tag{2.18}$$

y

$$\sum_{i=1}^{m}\alpha_i-\delta=\nu \tag{2.19}$$

Manipulando algebraicamente (2.15)

$$\begin{aligned} L(\mathbf{w},\boldsymbol{\xi},b,\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\delta)=&\frac{1}{2}\|\mathbf{w}\|^2-\nu\rho+\sum_{i=1}^{m}D_i\xi_i-\sum_{i=1}^{m}\alpha_i y_i\langle\mathbf{w},\mathbf{x}_i\rangle_{\mathcal{X}}-b\sum_{i=1}^{m}\alpha_i y_i \\ &+\rho\left(\sum_{i=1}^{m}\alpha_i-\delta\right)-\sum_{i=1}^{m}(\alpha_i+\beta_i)\xi_i \end{aligned} \tag{2.20}$$

Reemplazando (2.16), (2.17) y (2.19) en (2.20) se tiene

$$L\left(\mathbf{w},\boldsymbol{\xi},b,\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\delta\right)=\frac{1}{2}\left\|\mathbf{w}\right\|^2-\sum_{i=1}^{m}\alpha_i y_i\left\langle\mathbf{w},\mathbf{x}_i\right\rangle_{\mathcal{X}} \tag{2.21}$$

Sustituyendo (2.18) en (2.21) y simplificando

$$L\left(\mathbf{w},\boldsymbol{\xi},b,\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\delta\right)=-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i y_i\alpha_j y_j\left\langle\mathbf{x}_i,\mathbf{x}_j\right\rangle_{\mathcal{X}} \tag{2.22}$$

Como $\alpha_i,\beta_i\geq 0$ y junto con (2.17), una restricción es

$$0\leq\alpha_i\leq D_i \qquad para \quad i=1,...,m \tag{2.23}$$

Como $\delta\geq 0$ y a partir de (2.19)

$$\sum_{i=1}^{m}\alpha_i\geq\mu \tag{2.24}$$

Aplicando el truco del kernel al Lagrangiano (2.22) y junto con las restricciones (2.16), (2.23) y (2.24) el problema dual en el espacio de características es

$$\begin{aligned}
&\max_{\boldsymbol{\alpha}\in\mathbb{R}^m}\ W\left(\boldsymbol{\alpha}\right)=-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i y_i\alpha_j y_j k\left(x_i,x_j\right)\\
&s.a.\quad \sum_{i=1}^{m}\alpha_i y_i=0\\
&\qquad 0\leq\alpha_i\leq D_i \qquad\qquad para\quad i=1,...,m\\
&\qquad \sum_{i=1}^{m}\alpha_i\geq\mu
\end{aligned} \tag{2.25}$$

El problema (2.25) en términos de minimización y en forma matricial es

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}\in\mathbb{R}^m}\ f\left(\boldsymbol{\alpha}\right)=\frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha}\\
&s.a.\quad \mathbf{y}^T\boldsymbol{\alpha}=0\\
&\qquad 0\leq\alpha_i\leq D_i \qquad\qquad para\quad i=1,...,m\\
&\qquad \mathbf{e}^T\boldsymbol{\alpha}\geq\nu
\end{aligned} \tag{2.26}$$

donde $Q_{ij}=y_i y_j k\left(x_i,x_j\right)$.

Al igual que en el caso de *C*-SVM , en problema dual (2.26) con respecto al dual original (1.88), solo varía en las restricciones sobre los α_i dependiendo ahora de la distribución.

Así mismo la función de decisión está determinada por (1.69), mientras que el umbral b y el parámetro del margen ρ se computan según (1.97) y (1.98) respectivamente, con los valores de r_1 y r_2 dados por (1.94) y (1.96) con la salvedad que los conjuntos S_+ y S_- ahora contienen los vectores de soporte x_j con $0 < \alpha_j < D_i$ para $y_j = +1$ y $y_j = -1$ respectivamente.

2.1.3 HARD C-SVM CON DISTRIBUCIONES

Alternativamente no sólo se puede penalizar la función objetivo de acuerdo al peso en la distribución, sino que además se puede obligar a incrementar el margen de acuerdo a este peso, en este caso el problema de optimización es

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{X},\boldsymbol{\xi}\in\mathbb{R}^m,b\in\mathbb{R}} \quad & \tau(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} D_i\xi_i \\ s.a. \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{X}} + b\right) \geq mD_i\left(1-\xi_i\right) \\ & \xi_i \geq 0 \qquad\qquad para \quad i = 1,\ldots,m \end{aligned} \tag{2.27}$$

Donde D_i es el peso del par $\langle \mathbf{x}_i, y_i\rangle$.

Nuevamente si la distribución es uniforme, el problema (2.27) es equivalente a (1.17).

Para hallar el dual de (2.27) se utiliza el Lagrangiano

$$\begin{aligned} L(\mathbf{w},\boldsymbol{\xi},b,\boldsymbol{\alpha},\boldsymbol{\beta}) = & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} D_i\xi_i \\ & - \sum_{i=1}^{m}\alpha_i\left(y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{X}} + b\right) - mD_i\left(1-\xi_i\right)\right) - \sum_{i=1}^{m}\beta_i\xi_i \end{aligned} \tag{2.28}$$

Con los multiplicadores de Lagrange $\alpha_i, \beta_i \geq 0$.

De las condiciones de Karush-Kuhn-Tucker (KKT) para $b, \boldsymbol{\xi}$ y $\mathbf{w}$ se obtiene

$$\sum_{i=1}^{m}\alpha_i y_i = 0 \tag{2.29}$$

$$mD_i\alpha_i + \beta_i = C \cdot D_i \tag{2.30}$$

y

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{2.31}$$

Ahora, manipulando algebraicamente (2.28) se tiene

$$\begin{aligned} L(\mathbf{w},\boldsymbol{\xi},b,\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle_{\mathcal{X}} - \sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{w},\mathbf{x}_i\rangle_{\mathcal{X}} + m\sum_{i=1}^{m} \alpha_i D_i - b\sum_{i=1}^{m} \alpha_i y_i \\ + \sum_{i=1}^{m} \xi_i \left(C \cdot D_i - m\alpha_i D_i - \beta_i\right) \end{aligned} \tag{2.32}$$

Utilizando (2.29) y (2.30) en (2.32)

$$L(\mathbf{w},\boldsymbol{\xi},b,\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle_{\mathcal{X}} - \sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{w},\mathbf{x}_i\rangle_{\mathcal{X}} + m\sum_{i=1}^{m} \alpha_i D_i \tag{2.33}$$

Reemplazando (2.31) en (2.33)

$$L(\mathbf{w},\boldsymbol{\xi},b,\boldsymbol{\alpha},\boldsymbol{\beta}) = m\sum_{i=1}^{m} \alpha_i D_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \langle \mathbf{x}_i,\mathbf{x}_j\rangle_{\mathcal{X}} \tag{2.34}$$

Ahora despejando β_i de (2.30) y teniendo en cuenta que $\beta_i \geq 0$, entonces

$$mD_i\left(C/m - \alpha_i\right) \geq 0 \tag{2.35}$$

Como $mD_i \geq 0$ para todo $i = 1,...,m$ y unido al hecho que $\alpha_i \geq 0$ se tiene que

$$0 \leq \alpha_i \leq C/m \qquad para \quad i = 1,...,m \tag{2.36}$$

Finalmente, utilizando el truco del kernel en (2.34) y con las restricciones (2.29) y (2.36) el planteamiento del problema dual es

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; W(\boldsymbol{\alpha}) = m\sum_{i=1}^{m} \alpha_i D_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j k(x_i,x_j) \\ &s.a. \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \\ &\qquad 0 \leq \alpha_i \leq C/m \qquad\qquad para \quad i = 1,...,m \end{aligned} \tag{2.37}$$

Reformulando el problema (2.37) en forma matricial y en términos de minimización, el nuevo problema de optimización es:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^m} \quad & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - m\mathbf{D}^T\boldsymbol{\alpha} \\ s.a. \quad & \mathbf{y}^T\boldsymbol{\alpha} = 0 \\ & 0 \le \alpha_i \le C/m \qquad\qquad para \quad i = 1,...,m \end{aligned} \tag{2.38}$$

donde $Q_{ij} = y_i y_j k(x_i, x_j)$ y $\mathbf{D}$ es el vector de la distribución.

En este caso, teniendo en cuenta la distribución para la función objetivo primal y las restricciones, no se ven afectadas las restricciones del problema dual pero sí se modifica la función objetivo, como se aprecia en (2.38).

El hiperplano separador óptimo $h_{\mathbf{w},b}(\mathbf{x})$ sigue conservando la forma de (1.69) y el umbral b se calcula de la misma forma que para el C-SVM original de acuerdo a (1.72).

2.1.4 HARD ν-SVM CON DISTRIBUCIONES

Así mismo para ν-SVM se puede incluir la distribución tanto en la función objetivo como en la restricción sobre el margen.

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{H},\boldsymbol{\xi}\in\mathbb{R}^m,\rho,b\in\mathbb{R}} \quad & \tau(\mathbf{w},\boldsymbol{\xi},\rho) = \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \sum_{i=1}^{m} D_i\xi_i \\ s.a. \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{X}} + b\right) \ge mD_i(\rho - \xi_i) \\ & \xi_i \ge 0 \qquad\qquad para \quad i = 1,...,m \\ & \rho \ge 0 \end{aligned} \tag{2.39}$$

El Lagrangiano de (2.39) es

$$\begin{aligned} L(\mathbf{w},\boldsymbol{\xi},b,\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\delta) = & \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \sum_{i=1}^{m} D_i\xi_i \\ & - \sum_{i=1}^{m}\alpha_i\left(y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{X}} + b\right) - mD_i(\rho - \xi_i)\right) \\ & - \sum_{i=1}^{m}\beta_i\xi_i - \delta\rho \end{aligned} \tag{2.40}$$

donde $\alpha_i \beta_i, \delta \geq 0$ son los multiplicadores de Lagrange de las restricciones.

A partir de las condiciones de KKT para $b, \boldsymbol{\xi}, \mathbf{w}$ y ρ se obtiene

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{2.41}$$

$$m\alpha_i D_i + \beta_i = D_i \tag{2.42}$$

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{2.43}$$

y

$$m\sum_{i=1}^{m} \alpha_i D_i - \delta = \nu \tag{2.44}$$

Manipulando algebraicamente (2.40)

$$\begin{aligned} L\left(\mathbf{w}, \boldsymbol{\xi}, b, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta\right) = \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \sum_{i=1}^{m} D_i \xi_i - \sum_{i=1}^{m} \alpha_i y_i \left\langle \mathbf{w}, \mathbf{x}_i \right\rangle_{\mathcal{X}} - b\sum_{i=1}^{m} \alpha_i y_i \\ + \rho\left(m\sum_{i=1}^{m} \alpha_i D_i - \delta \right) - \sum_{i=1}^{m} \left(m\alpha_i D_i + \beta_i \right) \xi_i \end{aligned} \tag{2.45}$$

Reemplazando (2.41), (2.42) y (2.44) en (2.45) se tiene

$$L\left(\mathbf{w}, \boldsymbol{\xi}, b, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta\right) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i y_i \left\langle \mathbf{w}, \mathbf{x}_i \right\rangle_{\mathcal{X}} \tag{2.46}$$

Sustituyendo (2.43) en (2.46) y simplificando

$$L\left(\mathbf{w}, \boldsymbol{\xi}, b, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta\right) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle_{\mathcal{X}} \tag{2.47}$$

Como $\alpha_i, \beta_i \geq 0$ y junto con(2.42), una restricción es

$$0 \leq \alpha_i \leq 1/m \qquad para \quad i = 1, \ldots, m \tag{2.48}$$

Como $\delta \geq 0$ y a partir de (2.44)

$$m\sum_{i=1}^{m} \alpha_i D_i \geq \mu \tag{2.49}$$

Aplicando el truco del kernel al Lagrangiano (2.47) y junto con las restricciones (2.41), (2.48) y (2.49) el problema dual en el espacio de características es

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; & W(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i y_i \alpha_j y_j k(x_i, x_j) \\ s.a. \; & \sum_{i=1}^{m}\alpha_i y_i = 0 \\ & 0 \le \alpha_i \le 1/m \qquad para \quad i = 1,...,m \\ & m\sum_{i=1}^{m}\alpha_i D_i \ge \mu \end{aligned} \tag{2.50}$$

El problema (2.50) en términos de minimización y en forma matricial es

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} \\ s.a. \; & \mathbf{y}^T\boldsymbol{\alpha} = 0 \\ & 0 \le \alpha_i \le 1/m \qquad para \quad i = 1,...,m \\ & \mathbf{D}^T\boldsymbol{\alpha} \ge \nu \end{aligned} \tag{2.51}$$

donde $Q_{ij} = y_i y_j k(x_i, x_j)$ y $\mathbf{D}$ el vector de la distribución.

En este caso, la función objetivo no cambia respecto al problema dual original (1.88), solo cambia la restricción de desigualdad correspondiente al ν.

Así mismo la función de decisión está determinada por (1.69), mientras que el umbral b y el parámetro del margen ρ se computan según (1.97) y (1.98) respectivamente, con los valores de r_1 y r_2 dados por (1.94) y (1.96) como en el ν-SVM sin distribuciones.

2.2 SUPPORT VECTOR MACHINES COMO ALGORITMO DÉBIL

Debido a la degradación del desempeño que experimentan las técnicas de Boosting con *SVM* respecto al número de rondas **[25]**, resulta útil debilitarlo para aprovechar las ventajas de Boosting con respecto a la generalización.

Para hacer de *SVM* un clasificador débil y teniendo en cuenta que la complejidad está entre cuadrática y cúbica respecto al número de datos, es posible despreciar cierta cantidad de datos μ,

de tal forma que se resuelva el problema con un porcentaje mucho menor, siempre y cuando con respecto al conjunto total, el error pesado no supere el 50%.

Adicionalmente, los datos que se descarten deben ser los menos representativos del conjunto, es decir los que tengan menos peso en la distribución. De esta forma el subconjunto $\mathcal{J} \subset \mathcal{S}$ estará definido por

$$\sum_{j \in \mathcal{J}} D_j \leq (1-\mu) \tag{2.52}$$

donde $\mathcal{J}$ tiene mínima cardinalidad, presentando como ventaja adicional que el nuevo conjunto de entrenamiento es el más pequeño posible, minimizado al máximo el tiempo de entrenamiento, este algoritmo se presenta en la *Figura 8*.

Algoritmo : $WSVM\left(\mathcal{S}, D, SVM, \text{kernel}, C \text{ o } \nu, \mu\right)$
Entrada : $S = \{x_i, y_i\}_{i=1}^{m}, D, SVM(\cdot,\cdot), \text{kernel}, C \text{ o } \nu, \mu$
Salida : $h(\cdot)$
Seleccionar $\mathcal{J}$ tal que $\sum_{j \in \mathcal{J}} D_j \leq (1-\mu)$ y
$\mathcal{J}$ tenga mínima cardinalidad
Seleccionar $\mathcal{S}^* = \left\{\left\langle x_j, y_j \right\rangle\right\}_{\mathcal{J}}$ y $D^* = D_{\mathcal{J}}$
Obtener una hipótesis débil usando D^*
$h \leftarrow SVM\left(S^*, D^*, \text{kernel}, C \text{ o } \nu\right) = \sum_{i=1}^{m} y_i \beta_i k(x, x_i) + b$
Salida: $h(x)$

Figura 8. Algoritmo Weak SVM

Es posible obtener una cota para el porcentaje de datos rechazados μ utilizando un kernel universal (un kernel para el cual cualquier conjunto $\mathcal{S}$ es separable en el espacio de características como por ejemplo un kernel gaussiano) puesto que al ser *SVM* por si solo un aprendiz fuerte con respecto al subconjunto $\mathcal{J}$, se puede conseguir un error acotado por un ε lo suficientemente pequeño y en el peor de los casos equivocarse en todos los elementos del conjunto $\mathcal{J}$', de esta forma si *SVM* es un aprendiz débil se cumple que $(1-\mu)\varepsilon + \mu < 1/2$ de donde

$$\mu < \frac{1/2 - \varepsilon}{1 - \varepsilon} \tag{2.53}$$

Con lo cual μ es cercano al 50%. Sin embargo esta cota no es lo suficientemente ajustada puesto que los elementos de $\mathcal{J}$ y $\mathcal{J}$' están correlacionados y la regla de clasificación obtenida mediante $\mathcal{J}$ no tendrá un error considerable para $\mathcal{J}$' siempre y cuando el número de elementos de $\mathcal{J}$ sea representativo de la distribución original. Por esta razón μ es un parámetro adicional del algoritmo el cual se puede determinar mediante alguna estrategia de selección de modelo.

2.3 ALGORITMO BOOSTING SUPPORT VECTOR MACHINES (BSVM)

Luego de tener un algoritmo debilitado para *SVM,* se puede aplicar directamente un algoritmo de Boosting como Adaboost, con lo cual la hipótesis final del clasificador combinado de acuerdo a (1.69) y (1.101) está dada por

$$H_T(x) = \operatorname{sgn}\left(\sum_{t=1}^{T} \frac{\alpha_t}{\sum_t \alpha_t} \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b\right)\right) \tag{2.54}$$

Sin embargo es claro que (2.54) no es un hiperplano clasificador en el espacio de características y que la complejidad de la hipótesis $H_T(x)$ es mucho mayor que la de *SVM* dada por (1.69).

Por esta razón se desea modificar el algoritmo Adaboost de la *Figura 7.* de tal forma que el clasificador combinado también tenga la forma de (1.69). Este algoritmo denominado *BSVM* se detalla en la *Figura 9.*

Algoritmo : $\mathrm{BSVM}\left(S, D_1, T, WSVM, \mu, \text{kernel}, C \text{ o } \nu\right)$

Entrada : $S = \left\{x_i, y_i\right\}_{i=1}^{m}, D_1, T, WSVM, \mu, \text{kernel}, C \text{ o } \nu$

Salida : $H\left(\cdot\right)$

$H\left(x\right) \leftarrow WSVM\left(S, D_1, SVM, \text{kernel}, C \text{ o } \nu, \mu\right)$

$$= \sum_{i=1}^{m} y_i \beta_{1,i} k\left(x, x_i\right) + b_1$$

Para $t = 2$ hasta T o hasta condicion de terminación

$$D_t\left(i\right) = \frac{D_{t-1}\left(i\right)\left(\frac{1 - R_{emp}\left(h_{t-1}, S, D_{t-1}\right)}{R_{emp}\left(h_{t-1}, S, D_{t-1}\right)}\right)^{-\frac{1}{2} y_i h_{t-1}\left(x_i\right)}}{Z_{t-1}}$$

Donde Z_{t-1} es un factor de normalización escogido para que D_t sea una distribución.

$h_t\left(x\right) \leftarrow WSVM\left(S, D_t, SVM, \text{kernel}, C \text{ o } \nu, \mu\right)$

$$= \sum_{i=1}^{m} y_i \beta_{t,i} k\left(x, x_i\right) + b_t$$

$H\left(x, \alpha\right) = \alpha h_t\left(x\right) + \left(1 - \alpha\right) H\left(x\right)$

Escoger $\alpha_t \in \left[0,1\right]$ tal que

$\alpha_t = \arg\min\left(R_{emp}\left(H\left(\alpha, x\right), S, D_1\right)\right)$

$H\left(x\right) = H\left(x, \alpha_t\right)$

Fin

Salida: $H\left(x\right) = \text{sgn}\left(H\left(x\right)\right)$

Figura 9. Algoritmo BSVM

El algoritmo *BSVM* conserva las características principales de Adaboost. En cada iteración modifica de la misma forma la distribución de los datos, sin embargo la principal diferencia radica en que, debido a que la hipótesis final es también un hiperplano clasificador en el espacio de características, la forma de hallar los coeficientes α_t de cada hipótesis base se hace planteando un problema de optimización cuyo objetivo es minimizar el error en los datos de entrenamiento de la combinación convexa de la hipótesis actual $h_t(x)$ y la hipótesis combinada de los clasificadores anteriores $H(x)$. Por esta razón el problema de hallar α_t en cada ronda se reduce a un problema de búsqueda de línea con restricciones, de fácil resolución por métodos numéricos como búsqueda dorada o interpolación con la cúbica.

Adicionalmente, puesto que *BSVM* no aumenta la complejidad del modelo y por esta razón cada vez se hace más difícil encontrar un hiperplano combinado mejor, existe otro criterio de parada

diferente al número de rondas, el criterio es α_t =0 pues esto implica que el error de entrenamiento no mejoró y que la hipótesis resultante *H(x)* no sufre modificaciones. También lo es $R_{emp}(H(x),\mathcal{S},D_1)$ = 0 pues cuando el error de entrenamiento es cero, no es posible mejorar más.

Teniendo en cuenta que el objetivo de este proceso de entrenamiento es la generalización, no necesariamente el obtener el error más bajo en entrenamiento implica un error bajo de generalización, puesto que se puede presentar sobre ajuste a los datos, por esta razón se hace necesario incluir otros criterios de parada cuyo objetivo sea prevenir esto. Una primera alternativa es utilizar un sistema de *parada temprana* en donde a partir de un subconjunto de los datos de entrenamiento se verifique y controle cuando existe sobre ajuste y así finalizar el algoritmo.

Otra alternativa es utilizar los valores de error de entrenamiento de rondas anteriores para, aprovechando que forman una sucesión descendente, hallar en qué porcentaje se mejoró el error; un porcentaje muy bajo es indicio de sobre ajuste. Con lo cual se puede utilizar

$$\left(R_{emp.train}[t-1]-R_{emp.train}[t]\right)\Big/R_{emp.train}[t-1]\leq f[t] \qquad (2.55)$$

Siendo $f[t]$ decreciente o constante. Así mismo los α_t forman una sucesión, en la mayoría de los casos descendente, y valores muy bajos son prohibitivos para la generalización. De allí que otro criterio de terminación es

$$\alpha_t \leq g[t] \qquad (2.56)$$

Con $g[t]$ decreciente o constante.

3 PRUEBAS Y RESULTADOS EXPERIMENTALES

En la selección de experimentos de clasificación binaria para la aplicación del algoritmo *BSVM* propuesto, se buscó utilizar tanto datos artificiales como de problemas reales de diversas características respecto a dimensión y número de datos de entrenamiento y validación. A continuación se describen brevemente de los conjuntos seleccionados.

El primero de ellos es el conjunto de datos *MNIST* de dígitos manuscritos **[26]**, para este conjunto de 10 clases se toma el problema binario de clasificar las clases 3 y 8 puesto que es un problema de alta complejidad, adicionalmente es un conjunto de gran tamaño y de altísimas dimensiones.

Four-norm es un problema de clasificación binario artificial de 20 dimensiones en donde los datos de la primera clase provienen con igual probabilidad de dos distribuciones normales con matriz de covarianza identidad y medias en (*a,a,...,a*) y (*-a,-a,...,-a*) mientras los datos de la segunda clase provienen con igual probabilidad de dos distribuciones normales con matriz de covarianza identidad y medias en (*a,-a,...,a,-a*) y (*-a,a,...,-a,a*) para este caso se toma $a = \sqrt{2}$, se destaca en este conjunto, adicional a su dimensión, que las clases están bastante superpuestas, razón por la cual el error de Bayes es apreciable, pero a diferencia de un problema real, es calculable teóricamente.

Finalmente *Breast cancer, diabetes y australian* son tres bases de datos del repositorio de UCI **[27]**; la primera para identificar si un tumor es benigno o maligno, es un problema sencillo puesto que es separable fácilmente, la segunda para diagnosticar diabetes y la tercera para aprobar créditos; este grupo aunque es de menor número de datos respecto a las primeras, los datos son de dimensiones apreciables.

Es importante aclarar que en los casos en los que el conjunto no está fraccionado en entrenamiento y validación, se toma el 90% para entrenamiento y el 10% para validación, además en todos los experimentos se utilizó un kernel gaussiano. Para mayor detalle, las características de las bases de datos y los parámetros del kernel utilizado se encuentran en la *tabla I.*

Bases de Datos	# Elementos Entrenamiento	# Elementos Validación	Dim.	σ
MNIST	11982	1984	784	4M
Four norm	1000	10000	20	1k
Breast Cancer	615	68	9	100k
Diabetes	692	76	8	40
Australian	621	69	14	10

TABLA I CARACTERÍSTICAS BASES DE DATOS

Por el amplio número de datos en entrenamiento y validación, se utiliza MNIST para hacer un análisis general del algoritmo. La *Figura 10.* muestra el error de entrenamiento respecto al porcentaje de datos rechazados μ. Es claro que a medida que se rechazan más datos, el error de entrenamiento no puede disminuir al valor original aunque se corran varias rondas. También se observa la diferencia al incluir los criterios de parada, descritos anteriormente, para evitar sobre ajuste. Durante las pruebas realizadas se utilizó $f[t]=0.25$ y $g[t]=1/t^2$ como las funciones que intervienen en estos criterios.

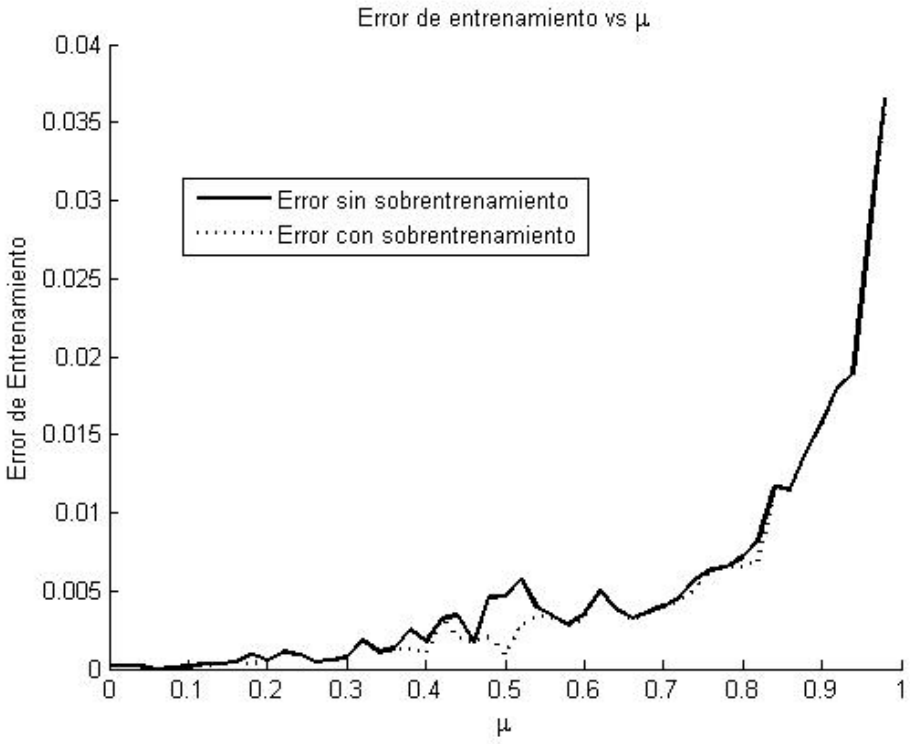

Figura 10. Relación entre el error de entrenamiento y la fracción μ de datos rechazados

Aunque para el entrenamiento no se disminuya tanto el error, la *Figura 11.* muestra cómo para la evaluación sí se mejora el desempeño, disminuyendo el error sobre todo a medida que se

rechazan más datos, mejorando la generalización y disminuyendo el tiempo de entrenamiento al tardar menos rondas. Adicionalmente, la aplicación de los criterios de parada diseñados permite un mejor desempeño respecto a la generalización pues se obtienen errores de validación menores o similares respecto al algoritmo sin estos criterios.

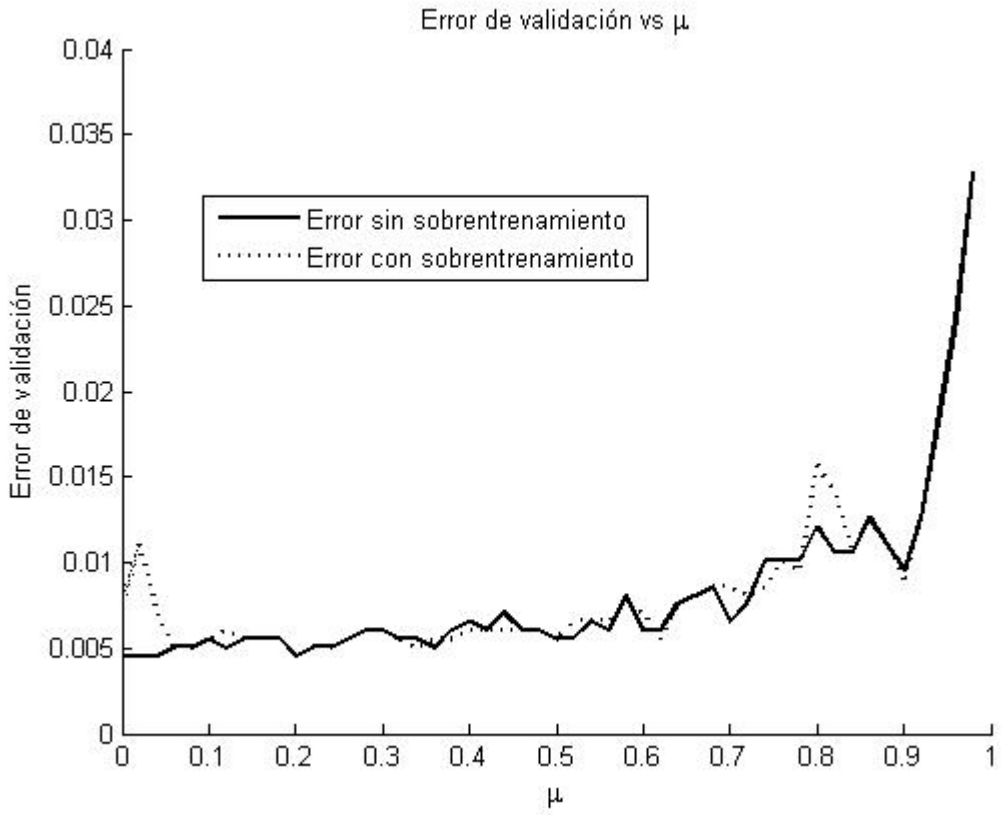

Figura 11. Relación entre el error de validación y la fracción μ de datos rechazados

La *Figura 12* muestra la relación entre el error de validación (sin sobre entrenamiento) y el tiempo de entrenamiento a medida que se rechazan más datos. A pesar que el algoritmo sigue generalizando a valores similares a los del modelo obtenido sin rechazar datos, el tiempo de entrenamiento decrece dramáticamente a valores inferiores a 1/10 del tiempo original. Sin embargo el modelo no generaliza bien si se desprecian demasiados datos.

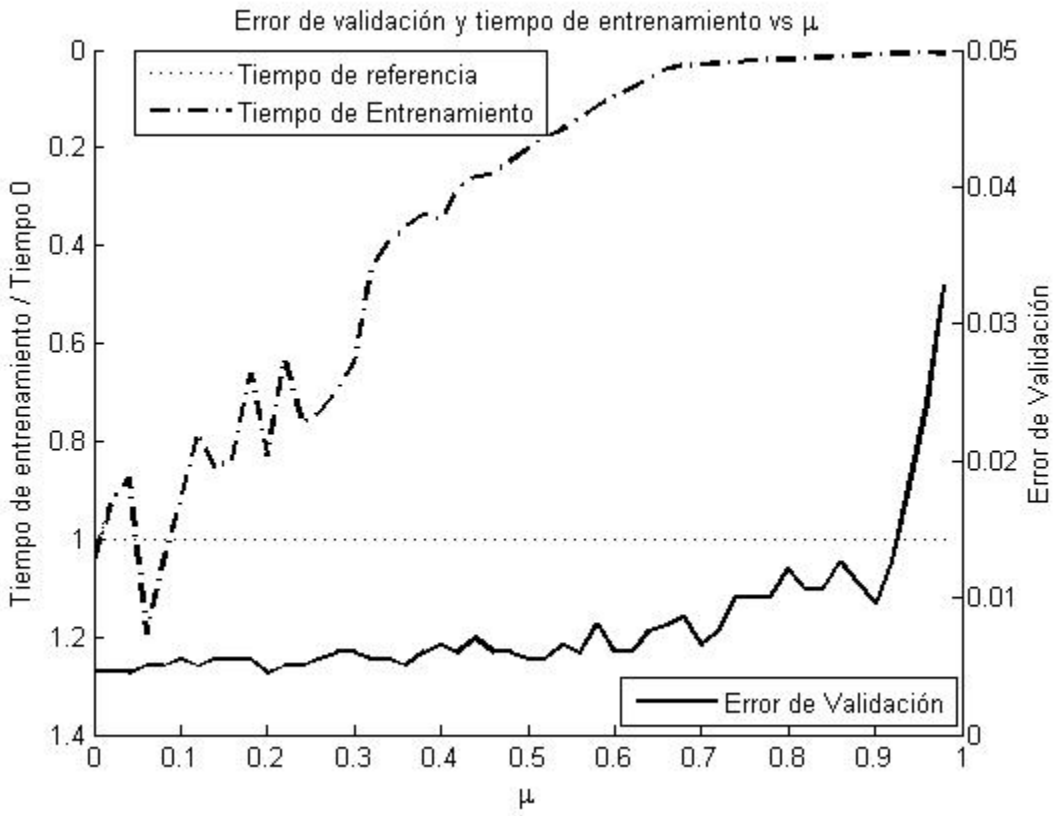

Figura 12. Relación entre el error de validación, el tiempo de entrenamiento y el porcentaje de datos rechazados μ

Respecto al número de vectores de soporte, la *Figura 13.* muestra como, aplicando los criterios de parada descritos, a medida que se rechazan más datos el número de vectores de soporte también disminuye independientemente del error de entrenamiento y de evaluación, sin embargo se destaca que en pruebas preliminares en ausencia de estos criterios, el sobre ajuste del modelo también implicaba, en algunos casos, un incremento sustancial en el número de vectores de soporte, lo que puede aprovecharse como un criterio adicional de parada.

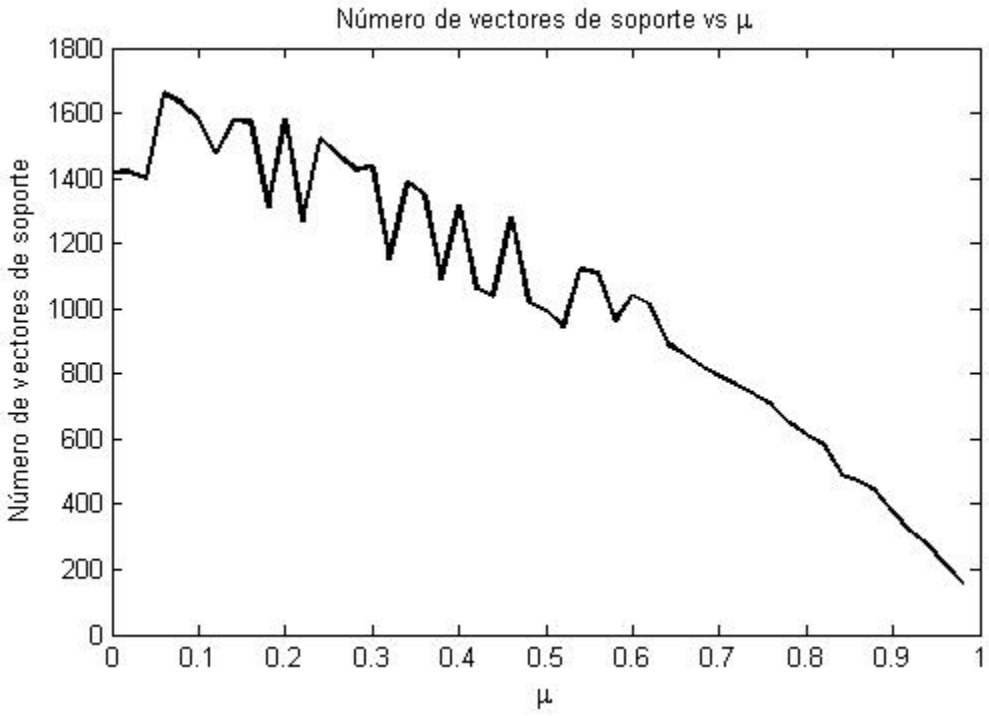

Figura 13. Relación entre el número de vectores de soporte y el porcentaje de datos rechazados μ sin sobreentrenamiento

La *tabla II.* resume los resultados obtenidos para los diferentes conjuntos probados para *C-SVM*, mientras que la *tabla III.* lo hace para *ν-SVM.*

Bases de Datos	C	E. Valid. *SVM* (%)	μ	# Iter	E. Valid. *BSVM*(%)	t. *BSVM* / t. *SVM*
MNIST	10	0.45	0.64	3	0.60	0.0762
Four norm	50	23.50	0.60	4	20.77	0.5212
	200	19.68	0.60	4	18.48	0.5617
B. Cancer	100	0.00	0.7	3	0.00	0.9286
Diabetes	50	19.74	0.7	3	22.37	0.3636
Australian	100	18.84	0.7	3	15.94	0.135

TABLA II RESULTADOS PARA C-*SVM*

Bases de Datos	ν	E. Valid. *SVM* (%)	μ	# Iter	E. Valid. *BSVM*(%)	t. *BSVM* / t. *SVM*
MNIST	0.15	4.83	0.60	3	4.54	0.6554
Four norm	0.3	21.05	0.60	4	26.06	0.8532
B. Cancer	0.4	1.47	0.80	3	0.00	1.52
Diabetes	0.2	26.32	0.80	3	26.32	1.13
Australian	0.4	15.94	0.80	3	13.04	0.76

TABLA III RESULTADOS PARA ν-*SVM*

De acuerdo a esto hay varios aspectos a destacar. El primero de ellos hace referencia a que en todos los casos *BSVM* obtuvo con muy pocas rondas un error de generalización similar al del algoritmo original, demostrando la efectividad y rápida convergencia del algoritmo diseñado.

Adicionalmente en la mayoría de los casos lo hace en un tiempo menor, en particular con *MNIST* y *australian,* conjuntos con muchos datos y/o altas dimensiones. Finalmente, de acuerdo a las relaciones de tiempos es claro que el algoritmo *C-BSVM* es más eficiente que *ν-BSVM* puesto que la proporción de tiempo para este último algoritmo es mayor comparada con el primero.

En relación con el número de vectores de soporte, la *tabla IV.* muestra como *BSVM* obtiene un número reducido de vectores de soporte, sin comprometer la generalización del modelo, esta reducción es mucho mayor en aquellas clases que son no separables y por lo tanto tienen un error de Bayes apreciable, como es el caso de *four-norm* o *diabetes.*

Bases de Datos	C	# SV C *SVM*	# SV C *BSVM*	ν	# SV ν *SVM*	# SV ν *BSVM*
MNIST	10	1417	1012	0.15	1817	1175
Four norm	200	688	454	0.3	340	395
B. Cancer	100	78	46	0.4	216	58
Diabetes	50	327	147	0.2	121	50
Australian	100	202	104	0.4	245	93

TABLA IV COMPARACIÓN DEL NÚMERO DE VECTORES DE SOPORTE

4 CONCLUSIONES

El algoritmo propuesto BSVM, combina eficientemente diversos clasificadores SVM por medio de técnicas de Boosting, sin aumentar la complejidad de la hipótesis resultante y en un tiempo mucho menor, en particular cuando el conjunto de entrenamiento es extenso y/o la dimensión de los datos es alta.

Adicionalmente, con esta implementación los modelos son mucho más compactos puesto que poseen un número menor de vectores de soporte.

Las estrategias propuestas para evitar el sobre ajuste son efectivas, de tal forma que BSVM presenta valores similares en cuanto a generalización respecto a la implementación original.

Quedan planteados como temas de investigación futuros, el hallar cotas teóricas más ajustadas para el porcentaje de datos rechazado y determinar criterios de parada más robustos y precisos.

BIBLIOGRAFÍA

[1] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press Inc, New York. 1995.

[2] B. Schölkopf, A. J. Smola, *Learning with Kernels.* MIT Press, Cambridge. 2002.

[3] N. Vapnik, *Statistical Learning Theory.* Wiley, 1998.

[4] D. Jhonson and F. Preparata, "The densest hemisphere problem" *Teorical computer Science,* No 6, pp. 93-107. 1978.

[5] C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 20:273–297, 1995.

[6] T. Hofmann and J. M. Buhmann. "Pairwise data clustering y deterministic annealing". *IEEE Transactions on Pattern analysis and Machine Intelligence,* 19(1):1-25, 1997.

[7] B. Schölkopf. "*Support Vector Learning"*. R. Oldenourg Verlag, München, 1997. Doktorarbeit, Technische Universität Berlin. Disponible en http://www.kyb.tuebingen.mpg.de/~bs.

[8] C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups.* Springer-Verlag, New York, 1984.

[9] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integrla equations" in *Philosopical transactions of the Royal Society of London Serie A,* Vol 209, pages 415-446. 1909.

[10] B. Schölkopf, A. J. Smola, R. Williamson, and P. Bartlett, *"New Support Vector Algoritms",* Royal Holloway College, University of London, UK, NeuroCOLT Technical Report NC-TR-98-031, 1998.

[11] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.

[12] E. Osuna, R. Freud, and F. Girosi. " An improved training algorithm for support vector machines" in J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for signal Processing VII – Proceedings of the 1997 IEEE Workshop,* pages 276-285, New York, 1997. IEEE.

[13] T. Joachims, "Making large-scale SVM learning practical" in B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning,* pages 169-184, Cambridge, MA, 1999. MIT Press.

[14] J. Platt. "Fast training of support vector machinesusing sequential minimal optimization" in B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning,* pages 185-208, Cambridge, MA, 1999. MIT Press.

[15] D. J. Crisp and C. J. C. Burges. "A geometric interpretation of ν-SVM classifiers" *Advances in Neural Information Processing Systems,*, S.Solla, T. Leen, and K.-R. Müller, Eds., volume 12, Cambridge, MA, MIT Press, 2000.

[16] C.-C. Chang and C.-J. Lin. "Training ν-support vector classifiers: Theory and algorithms" *Neural Computation*, 13(9):2119–2147, 2001.

[17] C.-C. Chang and C.-J. Lin. "LIBSVM: a library for support vector machines". Technical report, Computer Science and Information Engineering, National Taiwan University, 2001-2005. [Online]. Available: http://www.csie.ntu.edu.tw/_cjlin/libsvm.

[18] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning,* vol. 37, no. 3, pp. 297–336, Dec. 1999. [Online]. Available: http://www.boosting.org/papers/SchSin99b.ps.gz

[19] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997. [Online]. Available: http://www.boosting.org/papers/FreSch97.ps.gz

[20] I. Steinwart, "Cosistency of support vector machines and other regularized kernel classifiers," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 128–142, January 2005.

[21] Y. Freund and R. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[22] J. Wickramaratna, S. Holden, and B. Buxton, "Performance degradation in boosting," in *Proceedings of the 2nd International Workshop on Multiple Classifier Systems MCS2001*, ser. LNCS, J. Kittler and F. Roli, Eds. Springer, 2001, vol. 2096, pp. 11–21.

[23] P. Rangel, F. Lozano, E. García, "Boosting of Support Vector Machines with application to editing" in *Proceeding of the 4th Int. Conf. of Machine Learning and Applications ICMLA'05,* Dec. 2005.

[24] D. Pavlov, J. Mao, and D. Dom, "Scaling-up support vector machines using boosting algorithm," *15th International Conference on Pattern Recognition*, vol. 2, pp. 219–222, 2000.

[25] Y. Freud and R. E. Shapire. "Experiments with a new boosting algorithm" in *Proceedings of the 13th International Conference on Machine Learning,* pages 148-156. Morgan Kaufmann. 1996.

[26] Y. LeCun, "The MNIST database of handwritten digits", URL: http://yann.lecun.com/exdb/mnist/

[27] C. Blake and C. Merz, "UCI repository of machine learning databases" 1998, URL: http://www.ics.uci.edu/_mlearn/MLRepository.html