Máquinas ℓ-SVCR con salidas probabilísticas

Luis González
Dpto. Economía Aplicada I
Universidad de Sevilla
Avda Ramón y Cajal s/n SEVILLA
luisgon@us.es

Cecilio Angulo
Grup Recerca Enginyeria Coneixement
UPC
Vilanova i la Geltrú, BARCELONA
cangulo@esaii.upc.es

Francisco Velasco
Dpto. Economía Aplicada I
Universidad de Sevilla
Avda Ramón y Cajal s/n SEVILLA
velasco@us.es

Maria Luisa Vilchez
Dpto. Economía Aplicada
Universidad de Huelva
Plaza de la Merced s/n HUELVA
lobato@uhu.es

Resumen

En este trabajo se presenta una máquina ℓ-SVCR (ℓ-clases Support Vector Machine con restricciones de Clasificación y Regresión) basada en la teoría de aprendizaje estadístico, introducida en [AC01], la cual ha sido modificada de forma que la salida se puede interpretar en términos probabilísticos. Además esta nueva máquina proporciona tantas salidas intermedias como máquinas de soporte vectorial se han utilizado en el proceso de clasificación así como el grado de confianza que presenta cada una de ellas. Con objeto de comprobar la versatilidad de la máquina, ésta se aplica sobre un conjunto de datos sobre el que se comprueba empíricamente una de sus características más destacable, la prudencia en la elección del etiquetado.

Palabras clave: SVM, Aprendizaje, Reconocimiento de patrones.

1. Probabilidades en las SVMs

La Máquina de Vectores Soporte (o Máquina de Soporte Vectorial –SVMs–) estándar no proporciona probabilidades en el sentido de estimar la probabilidad de acertar en las predicciones, es decir, de estimar la distribución condicional $P[Y|X = x]$ con objeto de cuantificar la incertidumbre asociada a una predicción. Por ello, dentro de las SVMs se han elaborado diferentes aproximaciones a este problema, entre ellas la desarrollada por Peter Sollich en [Sol00], la cual esbozamos brevemente.

Sea un conjunto de entrenamiento

$$Z = \{(x_1, y_1), \cdots, (x_n, y_n)\} \qquad (1)$$

donde $\{x_1, \cdots, x_n\} \subset \mathcal{X} \subset \mathbb{R}^d$ e $y_i \in \mathcal{Y} = \{-1, 1\}$, correspondiente a un problema con dos etiquetas (dicotomías). En el análisis de las SVMs, el primer paso teórico a realizar consiste en transformar los inputs x en otros nuevos inputs $\phi(x)$ dentro de algún espacio característico

dotado de un producto escalar donde se trabaja con hiperplanos de decisión:

$$\pi \equiv \langle \omega, \phi(x) \rangle + b = 0$$

de tal forma que el problema de optimización SVM resultante es:

$$\min_{\omega \in \mathbb{R}^{d'}} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{n} \xi_i \qquad (2)$$

$$s.a. \begin{cases} y_i \left(\langle \omega, \phi(x_i) \rangle + b \right) - 1 + \xi_i \geq 0, \ \forall i \\ \xi_i \geq 0, \ \forall i. \end{cases}$$

En este problema se optimiza una función objetivo que es suma de la función $\frac{1}{2} \|\omega\|^2$ que expresa la búsqueda de un hiperplano de clasificación suave y la función $\sum_{i=1}^{n} \xi_i$ que cuantifica como de bien se lleva a cabo el proceso de clasificación (ajuste). Se complementa con una constante C que permite establecer una relación de intercambio entre suavidad y ajuste a los datos de la función solución.

Si el hiperplano solución obtenido a partir de los vectores de entrenamiento es $\pi(x)$ entonces la generalización del proceso de clasificación frente a un nuevo input es:

- Si $\pi(x) > 0$ asignamos a este vector la etiqueta 1.
- Si $\pi(x) > 0$ asignamos a este vector la etiqueta -1.
- Si $\pi(x) = 0$ no asignamos etiqueta alguna.

Con objeto de establecer este mismo problema pero de manera que nos conduzca a una interpretación probabilística se razona de la siguiente forma:
En el conjunto de entrenamiento, se cumple que los vectores (x_i, y_i) donde $y_i(\langle \omega, \phi(x_i) \rangle + b) \geq 1$ verifican que $\xi_i = 0$, y por tanto la función solución clasifica estos vectores correctamente y al no incurrir en error no penalizan la función objetivo. Por otro lado, los restantes vectores de entrenamiento contribuyen cada uno a incrementar la función objetivo en la cantidad

$$C\,\xi_i = C\left[1 - y_i(\langle \omega, \phi(x_i) \rangle + b)\right]$$

que se sigue de $\alpha_i \neq 0$ y de la igualdad $y_i(\langle \omega, \phi(x_i) \rangle) - 1 + \xi_i = 0$ (una de las condiciones de Karush-Kuhn-Tucker).

De esta forma el problema de optimización de las máquinas de vectores soporte biclasificadoras queda de la siguiente forma: encontrar el vector de parámetros $\omega \in \mathbb{R}^{d'}$ y el parámetro $b \in \mathbb{R}$ que minimiza

$$\frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{n} l(y_i(\langle \omega, \phi(x_i) \rangle + b))$$

donde $l(z)$ es la denominada función hinge loss o función soft margin, definida como:

$$l(z) = |1 - z|_+ = \begin{cases} 1 - z & \text{si } 1 - z > 0 \\ 0 & \text{en otro caso} \end{cases}$$

A partir de este problema, en [Sol00] se encuentra una distribución conjunta para el vector aleatorio (X, Y), de tal forma que el problema (2) coincide con un problema de máxima verosimilitud en los parámetros ω y b. De estos desarrollos se sigue que la probabilidad de $Y = y$ condicionada a $X = x$ y al vector de parámetros $\theta = (\omega, b)$ es, considerando la función $\theta(x) = \langle w, \phi(x) \rangle + b$:
$P(y|x, \theta) =$

$$\begin{cases} \dfrac{1}{1 + e^{-2Cy\theta(x)}} & \text{si} \quad |\theta(x)| \leq 1 \\ \dfrac{1}{1 + e^{-Cy[\theta(x) + signo(\theta(x))]}} & \text{si} \quad |\theta(x)| > 1 \end{cases}$$

Por otro lado, el proceso de generalización (asignación de nuevas etiquetas) que se lleva a cabo con estas probabilidades no sufre ninguna modificación con respecto a la generalización de las SVM estándar ya que si para un nuevo x, la solución $\theta^*(x) > 0$ entonces

$$P(Y = 1/\theta^*(x)) > P(Y = -1/\theta^*(x))$$

y la salida[1] de la máquina es $Y = 1$; y análogamente si $\theta^*(x) < 0$, la salida de la máquina es $Y = -1$. Sin embargo, cuando se plantee un problema de multiclasificación utilizaremos probabilidades para llevar a cabo el proceso de interpretación de la solución en lugar de los valores de la función de decisión ya que las probabilidades están normalizadas y pueden compararse entre ellas[2].

[1] Se elige como etiqueta aquella que presente una mayor probabilidad (verosimilitud).

[2] Es conocido en SVM que las salidas numéricas de diferentes máquinas no son comparables entre si ya que en el planteamiento de cada problema se lleva a cabo distintas normalizaciones.

2. SVMs para la multiclasificación

Consideramos en este caso que el conjunto de etiquetas posibles es $\{\theta_1, \cdots, \theta_\ell\}$, siendo $\ell > 2$ y sin una relación de orden definida en el etiquetado. Sea Z el conjunto de entrenamiento definido en (1); se construyen los subconjuntos

$$Z_k = \{(x_i, y_i), \text{ tales que } y_i = \theta_k\},$$

que determinan una partición de Z, es decir, se tiene:

$$\bigcup_{k=1}^{\ell} Z_k = Z$$
$$Z_k \bigcap Z_h = \emptyset \quad \forall\ k \neq h$$

Denotamos por n_k el número de vectores de entrenamiento del conjunto Z_k ($n = n_1 + n_2 + \cdots + n_\ell$); y por I_k el conjunto de índices i tales que $(x_i, y_i) \in Z_k$ de donde se sigue que $\bigcup_{i \in I_k} \{(x_i, y_i)\} = Z_k$.

La forma, más habitual, de utilización de las máquinas de vectores soporte para resolver problemas de multiclasificación admiten dos tipos de arquitectura:

- Máquinas biclasificadoras SV generalizadas: Construyen una función clasificadora global a partir de un conjunto de funciones clasificadoras dicotómicas (biclasificadoras).
- Máquinas multiclasificadoras SV: Construyen una función clasificadora global directamente considerando todas las diferentes clases a la vez.

Nosotros consideramos más adecuado trabajar con máquinas biclasificadoras SV generalizadas que con máquinas multiclasificadoras SV, puesto que con las primeras, podemos obtener como salidas los resultados de todas y cada una de las máquinas implementadas, y de esta forma disponer de un conjunto de resultados que nos permita tener una mayor capacidad de evaluación de la funcionalidad global del modelo.

Las máquinas biclasificadoras SV generalizadas dan solución al problema de la multiclasificación transformando las ℓ particiones del conjunto de entrenamiento en un conjunto de L biparticiones, en las cuales construye la correspondiente función clasificadora (es lo que se denomina **esquema de descomposición**) obteniendo $f_1, \cdots f_L$ clasificadores dicotómicos o biclasificadores. A continuación, mediante un **esquema de reconstrucción**, realiza la fusión de los biclasificadores f_i, $i = 1, \cdots, L$ con objeto de proporcionar como salida final, una de las ℓ clases posibles, $\{\theta_1, \cdots, \theta_\ell\}$.

Dentro del esquema de descomposición, las máquinas más utilizadas son:

- Máquinas 1-v-r SV (iniciales de *one-versus- rest*). Máquinas de vectores soporte, donde cada función clasificadora parcial f_i, enfrenta la clase θ_i contra el resto de las clases.
- Máquinas 1-v-1 SV (iniciales de *one-versus- one*). Máquinas de vectores soporte, donde cada función clasificadora parcial f_{ij}, enfrenta la clase θ_i contra la clase θ_j, sin considerar las restantes clases.

Por otro lado, puesto que empíricamente se ha contrastado que las máquinas 1-v-1 proporcionan mejores resultados que las máquinas 1-v-r, nosotros optamos por aquellas. Por ello, una vez repasada las características de este tipo de máquinas 1-v-1, en la siguiente sección, buscamos mejorarla, primero con la incorporación de las máquinas ℓ-SVCR y a continuación dando a las salidas de ésta última una interpretación probabilística.

2.1. Máquinas 1-v-1 SV

En esta aproximación del problema de multiclasificación se construyen $L = \frac{\ell \cdot (\ell - 1)}{2}$ biclasificadores donde la función discriminante f_{kh}, $1 \leq k < h \leq \ell$ discrimina los vectores de entrenamiento de la clase k, Z_k, de los vectores de entrenamiento de la clase h, Z_h, esto es, si el biclasificador f_{kh} lleva a cabo la discriminación de las clases sin error, entonces $sign(f_{kh}(x_i)) = 1$, si el vector $x_i \in Z_k$ y $sign(f_{kh}(x_i)) = -1$, si el vector $x_i \in Z_h$. Los restantes vectores de entrenamiento $Z \setminus \{Z_k \bigcup Z_h\}$ no se consideran en la construcción del problema de optimización.

De esta forma, dado un nuevo input x, la salida numérica de la máquina $f_{kh}(x)$ se interpreta de

la siguiente forma:

$$\Theta(f_{kh}(x)) = \begin{cases} \theta_k & \text{si } sign(f_{kh}(x)) = 1 \\ \theta_h & \text{si } sign(f_{kh}(x)) = -1. \end{cases}$$

En la reconstrucción del problema se utiliza algún esquema de votación que tenga en cuenta la distribución de las etiquetas asignadas por las máquinas parciales:

Etiquetas	Votos
θ_1	m_1
$\vdots$	$\vdots$
θ_k	m_k
$\vdots$	$\vdots$
θ_ℓ	m_ℓ
	$\frac{\ell\cdot(\ell-1)}{2}$

donde m_k es el número de votos que las máquinas f_i, $i = 1, \cdots, \frac{\ell\cdot(\ell-1)}{2}$ dan a la etiqueta θ_k.

Las características más significativas de este multiclasificador son las siguientes:

c1.- Se necesita estimar entrenar $\frac{\ell\cdot(\ell-1)}{2}$ SVMs, aunque con un conjunto de entrenamiento más reducido.

c2.- Es conocido que este procedimiento es, generalmente, preferido al esquema 1-v-r como así lo demuestran diferentes estudios empíricos[3].

Los dos principales inconvenientes que presentan este multiclasificador son:

i1.- Cada uno de los biclasificadores es entrenado con datos extraídos de solo dos clases del conjunto de entrenamiento por lo que la varianza es mayor y no proporciona información sobre el resto de clases. Además, cada máquina f_{kh} entrenada, no utiliza la información disponible en los datos que quedan fuera de las etiquetas θ_k y θ_h, lo que supone una preocupante pérdida de información.

i2.- El número de clasificadores, en comparación con las máquinas 1-v-r es alto, si el número de etiquetas ℓ es grande.

[3] Ver por ejemplo en [Kre99].

Hay que destacar, que a pesar de los buenos resultados empíricos apuntados por este esquema de multiclasificación, el principal inconveniente que presenta es el de no utilizar toda la información disponible dentro del conjunto de entrenamiento, lo que conlleva no sólo pérdida de información disponible sino también introducción de error debido a su no inclusión dentro de la frontera de la solución SVM. Esto se ve claro con el siguiente ejemplo: si consideramos la función biclasificadora f_{kh} y tomamos cualquier vector input del conjunto de entrenamiento x_i tal que $i \notin I_k \bigcup I_h$, para que esta función no proporcione una salida incorrecta debe verificar que $f_{kh}(x_i) = 0$. De esta forma, en la construcción de esta máquina estamos obligando a que todas las clases distintas de θ_k y θ_h estén dentro del hiperplano $f_{kh}(x) = 0$, y lo que resulta menos creíble, sin tener en cuenta las características de estas clases.

3. Máquinas ℓ-SVCR para multiclasificación

En [Ang01] se introduce un nuevo tipo de SVM para la multiclasificación, denominada ℓ-SVCR (ℓ-clases Support Vector Machine con restricciones de Clasificación y Regresión), con objeto de evitar el principal inconveniente que presentan las máquinas 1-v-1, pero manteniendo a la vez todas las ventajas de este tipo de esquema.

Con objeto de dar una mayor claridad a los posteriores desarrollos, supongamos se quiere buscar una función que clasifique los vectores inputs de entrenamiento correspondiente a la clase θ_1 de los de la clase θ_2. Para ello, realizamos, en primer lugar, una ordenación de los vectores de entrenamiento de tal forma que los n_1 primeros pertenezcan a la clase θ_1, los n_2 siguientes a la clase θ_2 y los restantes $(n-n_1-n_2)$ pertenecen al restos de las clases, $\{\theta_3, \cdots, \theta_\ell\}$.

Como en el problema clásico de las SVMs buscamos, inicialmente, un hiperplano clasificador $f_{12}(x) = 0$ que separe adecuadamente las clases θ_1 y θ_2, pero ahora imponemos que se tenga en cuenta el resto de las clases, en la construcción del problema de optimización. De esta forma, al hiperplano $f_{12}(x)$ se le exige que deje los vectores inputs de la clase θ_1 en la región $\left\{x \in \mathbb{R}^d, \text{ tal que } f_{12}(x) \geq 1\right\}$,

a los vectores inputs de la clase θ_2 en la región $\left\{x \in \mathbb{R}^d, \text{ tal que } f_{12}(x) \leq -1\right\}$ y para los vectores inputs restantes se le reserva una región dependiente de un parámetro[4] $0 \leq \delta < 1$ de tal forma que caigan en la región $\left\{x \in \mathbb{R}^d, \text{ tal que } |f_{12}(x)| \leq \delta\right\}$, es decir, a diferencia de las máquinas SV 1-v-1, donde se obligaba implícitamente a los restantes vectores de entrenamiento a que perteneciese al hiperplano clasificador, con el parámetro δ se habilita una región de holgura alrededor de la solución donde incluir todos los restantes vectores de entrenamiento.

Si dicha solución existe considerando un hiperplano clasificador de la forma $f_{12}(x) = \langle \omega, x \rangle + b$, entonces se podrá resolver el siguiente problema de optimización[5] ℓ-SVCR sin pérdidas[6]:

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{2} \|w\|^2$$

sujeto a

$$y_i \left(\langle \omega, x_i \rangle + b\right) \geq 1, \ \ \forall i = 1, \cdots, n_3 \quad (3)$$

$$-\delta \leq \langle \omega, x_i \rangle + b \leq \delta, \ \ \forall i = n_3 + 1, \cdots, n \quad (4)$$

con $0 \leq \delta < 1$, y la clase que proporciona como salida la máquina ante un nuevo vector input x se interpreta, a partir de una función intérprete Θ, de la siguiente forma:

$$\Theta(f_{12}(x)) = \begin{cases} \theta_1 & \text{si } f_{12}(x) > \delta \\ \theta_2 & \text{si } f_{12}(x) < -\delta \\ \theta_0 & \text{si } |f_{12}(x)| < \delta \end{cases} \quad (5)$$

donde θ_0 es una etiqueta artificial que recoge cuando la máquina no realiza un etiquetado concreto, es decir, cuando no clasifica.

Si no existe solución al problema de optimización anterior, entonces se relaja las restricciones (3) y (4) utilizando variables holguras (ξ_i, φ_i y φ_i^*) y se busca un hiperplano de la forma $f_{12}(x) = \langle \omega, x \rangle + b$, que resuelva el siguiente problema ℓ-SV:

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{2} \|\omega\|^2 + C_1 \sum_{i=1}^{n_3} \xi_i + C_2 \sum_{i=n_3+1}^{n} (\varphi_i + \varphi_i^*) \quad (6)$$

sujeto a

$$y_i \left(\langle \omega, x_i \rangle + b\right) \geq 1 - \xi_i, \quad \forall i = 1, \cdots, n_3 \quad (7)$$

$$-\delta - \varphi_i^* \leq \langle \omega, x_i \rangle + b \leq \delta + \varphi_i, \quad \forall i = n_3 + 1, \cdots, n \quad (8)$$

$$\begin{array}{ll} \xi_i \geq 0, & \forall i = 1, \cdots, n_3 \\ \varphi_i^*, \varphi_i \geq 0, & \forall i = n_3 + 1, \cdots, n. \end{array} \quad (9)$$

Este problema aparece resuelto en [Ang01] y la solución viene dada explícitamente en la forma:

$$f_{12}(x) = \sum_{i=1}^{N_{sv}} \alpha_i \langle x_i, x \rangle + b$$

donde los α_i son los multiplicadores de Karush-Kuhn-Tucker asociados al problema (6), cumpliendo:

$$\sum_{i=1}^{N_{sv}} \alpha_i = 0$$

donde N_{sv} denota el número de vectores de entrenamiento tales que $\alpha_i \neq 0$, vectores que se denominan **vectores soporte** y son los que dan nombre a este tipo de máquinas. Por otro lado, b se obtiene a partir de las igualdades proporcionadas por las condiciones de Karush-Kuhn-Tucker sobre los vectores soporte.

Al igual que en las SVMs estándar, la solución obtenida en el problema de optimización ℓ-SVCR puede ser generalizada al caso no lineal[7] utilizando funciones núcleo[8], obteniéndose de esta forma como solución general al problema ℓ-SVCR:

$$f_{12}(x) = \sum_{i=1}^{N_{sv}} \alpha_i \, k(x_i, x) + b$$

[4] La condición de ser menor que la unidad viene impuesta con objeto de no solapar las diferentes regiones que se construyen. Este parámetro es elegido a priori.

[5] Si $n_3 = n_1 + n_2$.

[6] Es decir, la solución obtenida clasifica perfectamente todos los vectores de entrenamiento.

[7] No se buscan necesariamente hiperplanos en el espacio de los inputs.

[8] Dado el espacio de los vectores inputs $\mathcal{X}$ se considera una transformación ϕ de este espacio en un espacio vectorial dotado de un producto escalar $\mathcal{H}$ (denominado espacio característico) en la forma:

$$\phi : \mathcal{X} \subset \mathbb{R}^d \rightarrow \mathcal{H} \subset \mathbb{R}^{d'}$$

donde normalmente la dimensión de $\mathcal{H}$ (d') es muy superior a la dimensión del espacio $\mathcal{X}$ ($d << d'$). A partir de esta función ϕ, se dice que: una función **núcleo** es una función real de dos variables, denotada por k, que verifica:

$$\begin{array}{rcl} k : \mathcal{X} \times \mathcal{X} & \rightarrow & \mathbb{R} \\ (x, x') & \rightarrow & k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} \end{array}$$

donde $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denota el producto escalar en $\mathcal{H}$,

es decir, el problema de clasificación no se realiza sobre los vectores inputs directamente sino a través de sus transformados $\phi(x)$.

Así, dentro del problema de optimización de la máquina ℓ-SVCR se deben considerar los siguientes parámetros:

k.- Función núcleo.

C_1.- Ponderación dada a la suma de los errores de las dos clases que se discriminan.

C_2.- Ponderación dada a la suma de los errores de las restantes clases.

δ.- Factor de insensibilidad.

Respecto a la función núcleo es conveniente destacar la gran importancia que tiene ya que debe estar definiendo un espacio característico con una alta dimensión, con objeto de que la máquina presente un buen funcionamiento, puesto que hemos de obligar a que todos los vectores inputs con etiqueta θ_k, con $k = 3, \cdots, \ell$ estén dentro de una región relativamente "pequeña".

Para los parámetros C_1 y C_2 (que relacionan el intercambio entre ajuste y suavidad), al igual que en el caso SVM estándar, no hay ninguna regla adecuada para asignarles unos valores concretos, salvo el método de validación cruzada, con el coste en términos de datos que esto supone.

El parámetro δ, debe estar entre 0 y 1 con objeto de no solapar las regiones de clasificación. Como se indica en [Ang01], cuanto menor sea δ, menor es la capacidad de generalización de la máquina para los vectores con etiqueta θ_0, es decir, peor lleva a cabo la clasificación de estos vectores; y mayor es el número de vectores soporte necesarios en la construcción de la solución[9]. La idea originaria que fundamenta la adopción de este parámetro esta relacionada con la función de sensibilidad ε, desarrollada por Vapnik en [Vap98], que aparece en el tratamiento de las SVMR (Máquinas de Vectores Soporte para los problemas de Regresión).

[9]La solución se expresa a partir de un número alto de vectores de entrenamiento.

3.1. Probabilidades en las máquinas ℓ-SVCR

Consideramos el problema (6) sujeto a las restricciones (7)-(9). Sea $\theta(x) = \langle \omega, x \rangle + b$ una posible solución de la máquina, dependiendo de los parámetros ω y b, con $\omega \in \mathbb{R}^d$ y $b \in \mathbb{R}$. Se sigue que:

- Si el vector x_i tiene etiqueta θ_1, entonces dentro de la clasificación llevada a cabo por la máquina ℓ-SVCR la salida correcta sería $\theta(x_i) \geq 1$ ya que en este caso la etiqueta asignada en el planteamiento del problema es $y_i = 1$, que se corresponde con θ_1. En caso contrario, se sigue de (7) que la pérdida que ocasiona dentro de la función objetivo es $\xi_i = 1 - \theta(x_i) \geq 0$.

- Si el vector x_i tiene etiqueta θ_2, entonces dentro de la clasificación llevada a cabo por la máquina ℓ-SVCR la salida correcta sería $\theta(x_i) \leq -1$ ya que en este caso la etiqueta asignada en el planteamiento del problema es $y_i = -1$, que se corresponde con θ_2. En caso contrario, la pérdida que ocasiona dentro de la función objetivo es $\xi_i = 1 + \theta(x_i)$.

- Si el vector x_i tiene etiqueta θ_k con $k \neq 1, 2$, entonces dentro de la clasificación llevada a cabo por la máquina ℓ-SVCR, la salida correcta sería $|\theta(x_i)| \leq \delta$ ya que en este caso la etiqueta asignada en el planteamiento del problema es $y_i = 0$ que se corresponde con θ_0. En caso contrario, la pérdida que ocasiona es: $\varphi_i^* = -\theta(x_i) - \delta$ si $\theta(x_i) < -\delta$ y $\varphi_i = \theta(x_i) - \delta$ si $\theta(x_i) > \delta$.

Si consideramos la función hinge loss entonces asignamos a las salidas $y = 1$ e $y = -1$ de la máquina ℓ-SVCR las siguientes[10] "probabilidades", en función de un nuevo input x, y los parámetros ω y b:

$$Q[y = 1|\theta(x)] = \kappa(C_1, C_2)\, e^{[-C_1\ l(\theta(x))]},$$
$$Q[y = -1|\theta(x)] = \kappa(C_1, C_2)\, e^{[-C_1\ l(-\theta(x))]},$$

con $\kappa(C_1, C_2)$ a determinar.

[10]Se sigue de los desarrollos de [Sol00], pero añadiendo una nueva constante C_2. Realmente no es una probabilidad pero nos servirá para su construcción.

Si consideramos la función de insensibilidad δ:

$$|z|_\delta = \begin{cases} -z-\delta & \text{si} \quad z < -\delta \\ 0 & \text{si} \quad -\delta \leq z \leq \delta \\ z-\delta & \text{si} \quad \delta < z \end{cases}$$

entonces asignamos a la salida $y = 0$ de la máquina ℓ-SVCR, la siguiente "probabilidad", en función de un nuevo input x, y los parámetros ω y b:

$$Q[y = 0|\theta(x)] = \kappa(C_1, C_2)\, exp\left[-C_2\ |\theta(x)|_\delta\right].$$

Para conseguir que ciertamente sean probabilidades basta considerar que $\kappa(C_1, C_2)$ es igual al recíproco de

$$v(\theta(x)) = \sum_{y\in\{-1,0,1\}} Q[y|\theta(x)]$$

y se tiene que eligiendo una adecuada distribución de X, de w y b, el problema de máxima verosimilitud que se sigue considerando las probabilidades:

$$\begin{aligned} P[y = 1|\theta(x)] &= e^{[-C_1\ l(\theta(x))]}/v(\theta(x)), \\ P[y = -1|\theta(x)] &= e^{[-C_1\ l(-\theta(x))]}/v(\theta(x)), \\ P[y = 0|\theta(x)] &= e^{\left[-C_2\ |\theta(x)|_\delta\right]}/v(\theta(x)) \end{aligned}$$

es el mismo que se plantea en las máquinas ℓ-SVCR.

En la Figura 1 aparece recogido un caso particular de estas probabilidades. Nótese, como se siguen los resultados que intuitivamente se esperan tenga la máquina, ya que si:

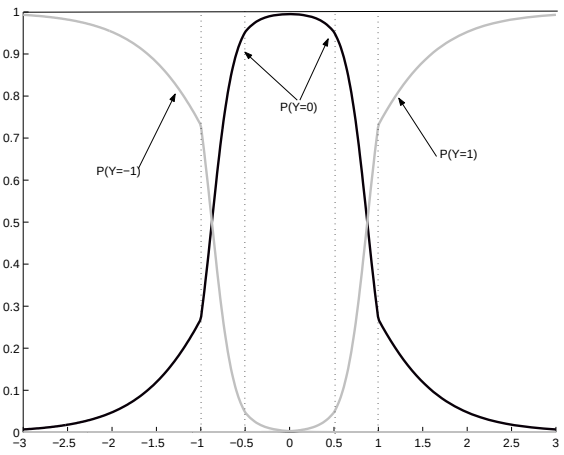

Figura 1: Funciones de probabilidad para $\delta = 0'5$, $C_1 = 6$ y $C_2 = 2$.

- $\theta(x) < -1$, la probabilidad de asignar la etiqueta $y = -1$ es mayor que las otras dos probabilidades, y es tanto mayor cuanto menor es $\theta(x)$.
- $\theta(x) > 1$, la probabilidad de asignar la etiqueta $y = 1$ es mayor que las otras dos probabilidades, y es tanto mayor cuanto mayor es $\theta(x)$.
- $-\delta < \theta(x) < \delta$, la probabilidad de asignar la etiqueta $y = 0$ es mayor que las otras dos probabilidades, y es tanto mayor cuanto más próximo se encuentre de 0.

3.2. Esquema de reconstrucción

Si tenemos en cuenta las probabilidades introducidas en estos modelos, podemos establecer una función que interprete la solución dada por la máquina ℓ-SVCR diferente a la dada en (5), de la siguiente forma:

$$\Theta(f_{12}(x)) = \begin{cases} \theta_1 & \text{si } P_0 > \text{máx}\{P_0, P_{-1}\} \\ \theta_0 & \text{si } P_0 \geq \text{máx}\{P_{-1}, P_1\} \\ \theta_2 & \text{si } P_{-1} > \text{máx}\{P_0, P_1\}. \end{cases} \tag{10}$$

con $P_i = P[Y = i/\theta(x)]$ para $i = 0, -1, +1$.

De esta forma, este nuevo intérprete es más reacio a etiquetar θ_1 y θ_2 que la interpretación inicial. Pensamos que esta elección mejora la anterior ya que si posteriormente se le da tanta importancia a los votos, con esta interpretación se hace más caro la obtención de ellos y lo que resulta más interesante, el difícil problema de resolver empates entre etiquetas lo resolvemos asignando a cada etiqueta un valor promedio de los grados de confianza conseguido con las funciones que lo votan. De esta forma en caso de empate asignamos como salida, aquella etiqueta que tengan una mayor grado de confianza en promedio. Con esta aproximación evitamos la comparación entre valores numéricos de distintas soluciones SVMs.

Además hemos considerado adecuado abrir la posibilidad de que nuestra máquina, proporcione tantas salidas, cuando se necesiten, como ℓ-SVCRs se hayan implementado, donde cada salida parcial indique:

- La etiqueta predicha por la ℓ-SVCR.

- Grado de confianza depositado en la salida.

De esta forma, se esta proporcionando al investigador un conjunto de información que le permite tomar una decisión mucho más precisa que si únicamente se le proporciona la salida final de una máquina.

Aclaremos lo anterior a partir de un ejemplo: Supongamos tenemos un problema con 4 clases distintas en las cuales hemos aplicado el clasificador 4-SVCR y se ha obtenido la siguiente salida asociada a un vector input x:

f_{kh}	1-2	1-3	1-4	2-3	2-4	3-4
	θ_1	θ_0	θ_4	θ_0	θ_4	θ_0
	65 %	80 %	70 %	80 %	80 %	63 %

En este caso, no se produce ningún empate en el número de etiquetas y la salida global de la máquina para un input x sería la etiqueta θ_4 con un grado de confianza del 75 % que es el valor medio[11] de los porcentajes dado por la máquina f_{14} (70 %) y la máquina f_{24} (80 %). Además, el investigador ha podido observar como el clasificador f_{12} ha asignado erróneamente[12] la etiqueta θ_1, pero también se equivoca el clasificador f_{34}, por lo que ha de ser considerado en un estudio a posteriori. En otras palabras, sería muy útil sobre un conjunto test, no utilizado en la construcción del problema de optimización, en el cual se conoce los inputs y sus etiquetas estudiar el comportamiento de los diferentes clasificadores parciales con objeto de disminuir el grado de incertidumbre presente cuando nos dispongamos a etiquetar un nuevo vector inputs.

Por otro lado, si la salida del multiclasificador fuese la siguiente:

f_{kh}	1-2	1-3	1-4	2-3	2-4	3-4
	θ_1	θ_1	θ_4	θ_0	θ_4	θ_0
	65 %	80 %	70 %	80 %	80 %	63 %

igual que anteriormente, la salida global de la máquina para x sería la etiqueta θ_4 con un grado de confianza del 75 %, pero se habría producido un empate a dos votos, con la etiqueta θ_1 y en el desempate se habría seleccionado θ_4 ya que el grado de confianza es mayor. Pero aún más, de las salidas intermedia se tiene que en el emparejamiento de ambas etiquetas, la etiqueta θ_4 es la "ganadora", lo cual debe tener un peso alto en la interpretación final dada por el investigador.

Con este último ejemplo, hemos intentado dar una visión general de como se trabajaría con este tipo de clasificadores. Por supuesto, habría que valorar todas las posibles combinaciones de etiquetas, por ejemplo ¿qué interpretación haría un investigador si todas los clasificadores etiquetase θ_0? ¿cómo se interpreta la situación, de un clasificador donde hay dos etiquetas que empatan a votos, presentan el mismo grado de confianza y en el enfrentamiento entre ellas, el clasificado proporciona la etiqueta θ_0? Pensamos que en estos casos la única solución es aquella que resulta del estudio que un experto realice de toda la información proporcionada por la salida de la máquina.

4. Datos Hatco

Los datos utilizados para contrastar la versatilidad de la máquina, que llamaremos **hatco**, esta compuesto por 100 vectores, tomados de [HATB00], tiene tres variables explicativas y una variable dependiente que presenta las diferentes etiquetas.

Las variables explicativas son:

- X_1 = Velocidad de entrega de un producto.
- X_2 = Nivel de precios.
- X_3 = Flexibilidad de precios.

La variable dependiente (etiquetado) presenta cuatro modalidades distintas que no presentan ningún orden establecido, y denotamos por $Y = \{1, 2, 3, 4\}$. Respecto a la elección de estos datos, los hemos elegidos debido a que el análisis discriminante clásico presenta unos pobres resultados, ya que si se eligen todos los datos como conjunto de entrenamiento el porcentaje de aciertos es de solo un 45 % (ver en [Gon02]). También introducimos un número como identificador de cada vector de trabajo.

La distribución del etiquetado es:

[11] Evidentemente, se podría elegir otro tipo de promedio.

[12] Si ciertamente θ_4 es la etiqueta correcta.

Etiquetas	1	2	3	4
Número	30	30	20	20

Así, si se asignase un etiquetado completamente aleatorio, la probabilidad de acierto, sería:

$A = \{\text{Acertar la clase de un determinado dato }\}$;

$$P(A) = 0'3^2 + 0'3^2 + 0'2^2 + 0'2^2 = 0{,}26$$

es decir, el 26 %. Por ello, en este caso, debemos exigir que nuestro modelo tenga una proporción de aciertos superior a esta cantidad.

Consideramos un conjunto de entrenamiento de tamaño 70 obtenido de forma aleatoria pero que mantenga la distribución del etiquetado (muestreo estratificado aleatorio):

Etiquetas	1	2	3	4
Número	21	21	14	14

Para llevar a cabo la clasificación hemos decidido dar más importancia a los errores que al suavizamiento de la solución final y consideramos:

- $C_1 = 100$ y $C_2 = 100$.

Nos decidimos por una máquina que sea medianamente conservadora en sus pronósticos y tomamos un parámetro de insensibilidad:

- $\delta = 0'5$.

La función núcleo elegida es:

- nucleo = función gaussiana[13]; con
- $p_1 = 0'5$,

de esta forma, tomando p_1 tan pequeño buscamos conseguir una función discriminadora "suficientemente" suave (dentro de lo que permite la elección de los parámetros C_1 y C_2).

Introduciendo estos parámetros en la máquina ℓ-SVCR, el etiquetado que ésta realiza sobre el conjunto de entrenamiento, expresado a partir de una matriz de clasificación, en la cual se ha recogido la etiqueta artificial $\theta_0 = 0$, es:

[13] $k(x,y) = e^{-\frac{1}{p_1}\|x-y\|^2}$

70	1	2	3	4	0
1	21	0	0	0	0
2	0	21	0	0	0
3	0	0	14	0	0
4	0	0	0	14	0

Observamos que no se cometen errores en la clasificación del conjunto de entrenamiento

En cuanto al comportamiento de la máquina sobre los restantes 30 datos de **hatco**, es como sigue:

30	1	2	3	4	0
1	6	0	0	0	3
2	0	7	0	0	2
3	0	0	4	0	2
4	0	0	0	4	2

Se observa que la máquina sigue sin equivocarse, clasificando correctamente el 70 % (21 de 30) de los datos pero esta vez no clasifica 9 de los 30 vectores de test (30 %). A continuación extraemos cuales son los identificadores de estos vectores no clasificados:

10	17	24	33	45	74	80	86	100

Si consideramos que el número de vectores sin etiquetar es grande, o queremos forzar a la máquina a que realice el etiquetado sobre un conjunto más grande, podemos variar el parámetro de insensibilidad δ. Así, si dejamos fijos los restantes parámetros y tomamos $\delta = 0'05$ (de esta forma la región reservada a los vectores con etiqueta θ_0 se reduce y la región de etiquetado aumenta) y se entrena nuevamente la máquina, entonces se tiene que la máquina clasifica correctamente todos los vectores de entrenamiento y sobre los vectores test, la matriz de clasificación es:

30	1	2	3	4	0
1	8	0	0	0	1
2	0	9	0	0	0
3	0	0	4	0	2
4	0	0	0	4	2

Consiguiendo clasificar correctamente el 83'33 % (25 de 30) y no etiqueta 5 de los 30 vectores de test (16'67 %). Estos vectores se corresponden con los datos cuyo identificador es:

```
24    74     80     86     100
```

Observamos que estos vectores ya se encontraban entre los no clasificados en la implementación anterior. Estos vectores sin etiquetar se pueden interpretar como vectores donde su etiquetado resulta "difícil" y por ello, la máquina "prudentemente" no los clasifica. Sin embargo, podemos forzar a través de algún otro parámetro[14] a que la máquina lleve a cabo el etiquetado de estos vectores. Así, en este caso, considerando $p_1 = 0'9$ se tiene la misma matriz de clasificación para el conjunto de entrenamiento utilizado en la implementación anterior, pero la matriz de clasificación para el conjunto de test es:

```
30    1    2    3    4    0
 1    8    1    0    0    0
 2    0    9    0    0    0
 3    0    0    4    2    0
 4    0    0    0    6    0
```

Es decir, en este caso se han etiquetado todos los vectores pero se ha errado tres veces en los correspondientes a los identificadores:

```
24          74          80
```

Aún así, el porcentaje de aciertos ha aumentado al 90 % (27 de 30).

Como ya hemos indicado, una de las características más resaltable de esta máquina ℓ-SVCR con salidas probabilísticas es la de poder realizar un análisis más detallado de las salidas ya que en el proceso de construcción de la función clasificadora obtenemos un buen conjunto de información intermedia. Esta información podemos verla recogida en el cuadro 1, referente al etiquetado llevado a cabo sobre el vector con identificador número 15. Para ello, la máquina a partir de las salidas intermedias construye una matriz de etiquetado y grado de confianza.

Por tanto, veamos como son los resultados intermedios de la máquina en los casos donde se ha llevado a cabo una clasificación errónea, y así de esta forma somos capaces de ver cual ha sido el comportamiento de la máquina en los fallos cometidos:

[14] Tomar un δ menor que $0'05$ haría que la capacidad de generalización de la máquina fuese muy pobre.

Cuadro 1: Interpretación de las salidas.

15	Identificador del vector.
1	Etiqueta real
1	Etiqueta predicha
1	Etiqueta asignada por f_{12}
1	Etiqueta asignada por f_{13}
1	Etiqueta asignada por f_{14}
0	Etiqueta asignada por f_{23}
0	Etiqueta asignada por f_{24}
0	Etiqueta asignada por f_{34}
0.9915	Grado de confianza de f_{12}
0.9814	Grado de confianza de f_{13}
0.9642	Grado de confianza de f_{14}
0.8595	Grado de confianza de f_{23}
0.8515	Grado de confianza de f_{24}
0.9856	Grado de confianza de f_{34}

```
    24        74        80
     1         3         3
     2         4         4
     2         0         0
     0         0         0
     0         4         4
     2         0         0
     2         4         0
     0         4         4
0.9972    1.0000    1.0000
1.0000    1.0000    1.0000
1.0000    1.0000    1.0000
1.0000    1.0000    1.0000
1.0000    0.9664    1.0000
1.0000    1.0000    1.0000
```

Observamos que en los tres datos, la máquina asigna el máximo número de votos (3) a la misma etiqueta pero de forma errónea. También se observa que los grados de confianza, en cada una de las máquinas es máximo (1=100 %), lo cual no nos permite atisba ningún tipo de inconsistencia en la clasificación, a pesar de ser clasificaciones erróneas. Por ello, uno de los trabajos futuros es modificar la máquina de manera que en el grado de confianza asignado a la etiqueta final permita clarificar lo más posible el proceso de etiquetado.

5. Conclusiones y trabajos futuros

La máquina ℓ-SVCR presentada en este trabajo es muy flexible gracias a los parámetros que presentan. Además la interpretación de estos es muy intuitiva y permite, desde un punto de vista subjetivo, una asignación a priori que generalmente suele proporcionar buenos resultados sin tener por ello que llevar a cabo ningún procedimiento de validación de parámetros.

En [AC01] ya se han realizado algunas comparativas con otros modelos y los resultados obtenidos han sido satisfactorios.

Otra línea de investigación que seguimos actualmente, es la de modificar ligeramente la máquina con objeto que ésta determine de forma automática el parámetro de insensibilidad δ, con lo cual se facilitaría la implementación.

Reconocimientos

Este trabajo ha sido soportado en parte por la ayuda ACC-265-TIC-2001 y ACC-265-TIC-2001 concedida por la Junta de Andalucía.

Referencias

[AC01] C. Angulo and A. Català. Ordinal regression with k-svcr machines. *Proc. of the IWAN*, 2001.

[Ang01] C. Angulo. *Aprendizaje con máquinas núcleo en entornos de multiclasificación.* Tesis doctoral, Universidad Politécnica de Catalunya, Mayo 2001.

[Gon02] L. González. *Análisis discriminante utilizando máquinas núcleos de vectores soporte. Función núcleo similitud.* Tesis doctoral, Universidad de Sevilla, Marzo 2002.

[HATB00] J. Hair, R. Anderson, R. Tatham, and W. Black. *Análisis Multivariante.* Prentice Hall, quinta edition, 2000.

[Kre99] U. Kressel. Pairwise classification and support vector machine. *In B. Schölkopf, C. Burgues and A. Smola, editors*, Advances in Kernel Methods: support Vector Learning. MIT Press. Cambridge, MA, 1999.

[Sol00] P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. Kluwer Academic Publidhers, 2000.

[Vap98] V.N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, Inc, 1998.