

Categorización de Documentos Usando Máquinas de Vectores de Soporte

Sergio Villazana

César Seijas

Antonino Caralli

Jesús Jiménez

José Pacheco

Centro de Investigación y Tecnología en Automatización, Electrónica y Control (CITAEC)

Facultad de Ingeniería, Universidad de Carabobo, Valencia, Edo. Carabobo, Venezuela

svillaza@ieee.org

cseijas@uc.edu.ve

acaralli@ieee.org

jjjimenes@uc.edu.ve

jpacheco@uc.edu.ve

RESUMEN

En este trabajo se desarrolló un clasificador de documentos (tesis de pregrado) basado en el clasificador nu-SVC y un kernel de cadenas. Se generó un corpus de entrenamiento con frases compuestas por una y dos de las palabras más representativas de las áreas seleccionadas (Ing Eléctrica e Ingeniería Civil), las cuales fueron extraídas de los títulos de los documentos. El conjunto de prueba se formó con todos los títulos de las tesis entre los años 1997 y 2006 (ambos inclusive). El desempeño del clasificador tuvo una predicción muy buena luego del proceso de entonación, al variar los parámetros de la nu-SVC y del kernel de cadenas. Los resultados demuestran el gran potencial que tienen las máquinas de vectores de soporte en el área de clasificación de texto.

Palabras claves: máquinas de vectores de soporte, categorización de texto, kernel de cadenas, clasificación, nu-SVC.

1. INTRODUCCIÓN

En la actualidad debido a la dinámica en la generación de información, el volumen de información electrónica crece rápidamente, y por tanto existe la necesidad de desarrollar herramientas que ayuden a las personas encargadas de manejarlas, a optimizar el proceso de búsqueda, y administración de estos recursos. La categorización (clasificación por categorías) de documentos es una actividad que consume mucho tiempo y que limita su aplicabilidad en casos de bases de datos muy grandes, de allí la necesidad de desarrollar tecnologías para una categorización automática.

La meta principal en la categorización de texto es hallar una función óptima para la clasificación de documentos a partir de atributos formados por palabras que definen las clases de las categorías de interés. En el proceso de organización de la data a partir del documento, éste se representa por un vector de palabras al igual de como se hace para el caso de recuperación de información (“*information retrieval*”).

Una técnica propuesta por [1] para categorizar texto basado en la representación clásica de texto donde se proyecta un documento como un vector en un espacio de rasgos de muchas dimensiones (hiperdimensional), donde cada componente del vector representa la presencia o ausencia de un rasgo. La principal desventaja de esta técnica es la pérdida de información contenida en el orden de las palabras, reteniendo solamente la frecuencia de los términos en el documento. Además de lo anterior, durante el pre-procesamiento del documento se remueven palabras que no dan información y se reemplazan las palabras por sus raíces, perdiéndose así información de inflexión.

En este trabajo, el documento se considera como una secuencia de símbolos, y su proyección al espacio de rasgos se realiza mediante el uso de un kernel conocido como kernel de

cadenas (“*string kernel*”) [1],[2],[3]. La principal ventaja de esta técnica es que no se considera a priori la pertenencia del documento a un área determinada, solo considera al documento como una secuencia larga de texto. En este caso, el espacio de rasgos se genera por el conjunto de todas las sub-secuencias o sub-cadenas (contiguas o no) de texto. En la medida que dos documentos tengan más sub-cadenas de texto en común mayor será su similaridad.

2. FUNDAMENTOS TEÓRICOS

Clasificación usando SVM

En el caso de aplicar las SVM para clasificación, la aplicación puede restringirse a un problema de dos clases sin pérdida de generalidad. El problema es separar dos clases por una función inducida de los ejemplos disponibles (datos de entrenamiento). Luego de hallada la función (SVM entrenada) que clasifica apropiadamente las clases, el objetivo es clasificar correctamente los datos o ejemplos que estén fuera del conjunto de entrenamiento, es decir, que la SVM capaz de generalizar.

En la Figura 1 se ilustra un problema de clasificación de objetos donde se busca definir un hiperplano separador que separe las clases u objetos: cuadrados de los ovalados. Obsérvese que existen infinitos hiperplanos separadores (solo se muestran dos posibles planos, señalados con líneas punteadas) pero solo uno maximiza la distancia entre el mismo y el objeto más cercano de cada clase (línea sólida), a este hiperplano lineal se le denomina hiperplano separador óptimo (HSO).

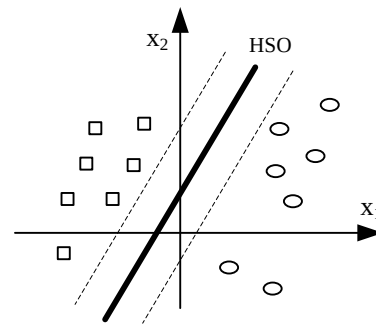


Figura 1. Clasificación de objetos con hiperplanos.

La definición formal del HSO es la siguiente: considere el problema de separar las clases (1 ó -1) del conjunto de de datos de entrenamiento D , como se indica a continuación:

$$D = \{(x^1, y^1), \dots, (x^l, y^l)\}, x \in R^n, y \in \{-1, 1\} \quad (1)$$

Donde l es el número de datos de entrenamiento, y representa cualquiera de las clases, y x es la posición de la clase y en el plano.

La ecuación del hiperplano para separar las clases y (1 ó -1) es

$$\langle w, x \rangle + b = 0, \quad w \in R^n, b \in R \quad (2)$$

El conjunto de vectores (datos) se dice que son óptimamente separados, si el hiperplano los separa sin error y la distancia entre el vector mas próximo al hiperplano es máxima [4],[5]. En ese caso el hiperplano definido por Ec. (2) es el *HSO*. Sin pérdida de generalidad, usualmente se considera un hiperplano canónico, donde los parámetros w y b se restringen según,

$$\min_i |\langle w, x^i \rangle + b| = 1 \quad (3)$$

es decir, la norma del vector de pesos w debe ser igual al inverso de la distancia del punto mas cercano en el conjunto de datos al *HSO*. Un *HSO* en forma canónica debe satisfacer las siguientes restricciones,

$$y^i [\langle w, x^i \rangle + b] \geq 1, \quad i = 1, \dots, l. \quad (4)$$

donde la distancia $d(w, b; x)$ de un punto x al *HSO* es,

$$d(w, b; x) = \frac{|\langle w, x \rangle + b|}{\|w\|} \quad (5)$$

El *HSO* se consigue maximizando el margen ρ , donde el margen se considera como la distancia entre el *HSO* y los vectores (datos) correspondientes a cada clase mas próximos a dicho hiperplano [6]. La maximización de ρ está sujeta a las restricciones definidas por Ec. (4); el margen ρ , se obtiene como,

$$\begin{aligned} \rho(w, b) &= \min_{x^i: y^i = -1} d(w, b; x^i) + \min_{x^i: y^i = +1} d(w, b; x^i) \\ &= \min_{x^i: y^i = -1} \frac{|\langle w, x^i \rangle + b|}{\|w\|} + \min_{x^i: y^i = +1} \frac{|\langle w, x^i \rangle + b|}{\|w\|} \\ &= \frac{1}{\|w\|} \left(\min_{x^i: y^i = -1} |\langle w, x^i \rangle + b| + \min_{x^i: y^i = +1} |\langle w, x^i \rangle + b| \right) \\ &= \frac{2}{\|w\|} \end{aligned} \quad (6)$$

en consecuencia el *HSO* minimiza,

$$\Phi(w) = \frac{1}{2} \|w\|^2 \quad (7)$$

independiente de b , para el siguiente límite

$$\|w\| < A \quad (8)$$

entonces de la Ec. (7) y la Ec. (8)

$$d(w, b; x) \geq \frac{1}{A} \quad (9)$$

es decir, los *HSO* no pueden estar mas cerca de $1/A$ de los puntos de datos. La solución al problema de optimización planteado por la Ec. (7), sujeto a las restricciones establecidas por Ec. (8) corresponde al punto de silla del siguiente funcional *Lagrangiano*:

$$\Phi(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i (y^i [\langle w, x^i \rangle + b] - 1) \quad (10)$$

En la Ec. (10), los α_i son los *multiplicadores de Lagrange*. La solución del problema anterior tiene un dual dado por,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} W \left(\min_{w, b} \Phi(w, b, \alpha) \right) \quad (11)$$

el mínimo con respecto a w y b del *Lagrangiano*, Φ , se deriva de:

$$\frac{\partial \Phi}{\partial b} = 0 \Rightarrow \sum_{i=1}^l \alpha_i y_i = 0 \quad (12)$$

$$\frac{\partial \Phi}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^l \alpha_i y_i x_i$$

luego de las Ec. (10), Ec. (11) y Ec. (12) se tiene:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^l \alpha_k \quad (13)$$

y por tanto, la solución es,

$$\alpha^* = \arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^l \alpha_k \quad (14)$$

sujeta a las restricciones,

$$\alpha_i \geq 0, \quad i = 1, \dots, l, y$$

$$\sum_{j=1}^l \alpha_j y_j = 0$$

Los parámetros del *HSO* son:

$$w^* = \sum_{i=1}^l \alpha_i y_i x_i \quad (15)$$

$$b^* = -\frac{1}{2} \langle w^*, x_r + x_s \rangle$$

donde x_r y x_s son los *vectores de soporte* de cada una de las clases que satisfacen,

$$\alpha_r \alpha_s > 0, \quad y_r = -1, y_s = +1 \quad (16)$$

definido el *HSO*, se puede diseñar un clasificador, usando una función fuertemente selectiva como lo es la función *signo* (*sgn*), luego un clasificador o SVC “duro” viene dado por,

$$f(x) = \text{sgn}(\langle w^*, x \rangle + b) \quad (17)$$

de las condiciones de *Kuhn-Tucker* [2],

$$\alpha_i (y^i [\langle w, x^i \rangle + b] - 1) = 0, \quad i = 1, \dots, l \quad (18)$$

solo los puntos x_i que satisfagan,

$$y^i [\langle w, x^i \rangle + b] = 1 \quad (19)$$

tendrán *multiplicadores de Lagrange* no nulos; esos x_i se denominan *vectores de soporte* (VS). Si los datos son linealmente separables, todos los VS residen en el margen y por tanto la cantidad de VS es pequeña. Consecuentemente el hiperplano es determinado por un pequeño subconjunto del conjunto de entrenamiento; es decir, descartando todos los otros vectores o datos del conjunto de entrenamiento y re-entrenando se obtendrá el mismo *HSO* y los mismos VS. Esta es una de las características que hacen atractivas las SVM, ya que puede resumirse la información contenida en el conjunto de datos a los VS producidos.

En el caso general, los datos del conjunto de entrenamiento no son linealmente separables. Existen dos estrategias para generalizar el problema, las cuales dependen del conocimiento a priori del problema en análisis y del estimado de ruido en los datos. En el caso donde se considera que un hiperplano puede separar correctamente la mayoría de los datos puede introducirse una función de costo adicional, que penalice los errores de clasificación. En ese sentido se puede definir un *HSO*

generalizado introduciendo variables no negativas, $\xi_i \geq 0$, y una función de castigo,

$$F_\sigma(\xi) = \sum_i \xi_i^\sigma \quad \sigma > 0 \quad (20)$$

donde los ξ_i representan una medida de los errores de clasificación. En el caso en análisis, el problema de optimización ahora debe minimizar tanto el error de clasificación como el acotamiento de la dimensión VC del SVC. Las restricciones impuestas por la Ec. (4), para el caso no separable, deben ser modificadas como,

$$y^i \left[\langle w, x^i \rangle + b \right] \geq 1 - \xi_i, i = 1, \dots, l. \quad (21)$$

donde $\xi_i \geq 0$. El HSO generalizado es determinado por el vector w , que minimiza el funcional,

$$\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad (22)$$

Obsérvese que en la ecuación anterior, se introduce el parámetro C o capacidad de la SVM, el cual se fija de modo que refleje, en alguna medida, el ruido estimado en los datos. La solución al problema de optimización planteado en Ec. (22) corresponde ahora al punto de silla del Lagrangiano,

$$\Phi(w, b, \alpha, \xi, \beta) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l (y^i [w^T x^i] - 1 + \xi_i) - \sum_{j=1}^l \beta_j \xi_j \quad (23)$$

siendo α y β los multiplicadores de Lagrange. La solución es de la forma,

$$\alpha^* = \arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \langle x_i x_j \rangle + \sum_{k=1}^l \alpha_k \quad (24)$$

con las restricciones,

$$0 \leq \alpha_i \leq C \quad i = 1, \dots, l$$

$$\sum_{j=1}^l \alpha_j y_j = 0 \quad (25)$$

es decir, la solución al problema de optimización es idéntica al caso separable excepto por una modificación al límite superior de los multiplicadores de Lagrange.

Usualmente, no es posible separar con una frontera lineal, en la dimensión del espacio de los datos; en es caso, la SVM puede proyectar el vector de entrada x , en un hiper-espacio de rasgos z . Con una proyección no lineal adecuada, será posible separar los vectores en ese hiper-espacio. El problema de optimización de Ec. (22) es ahora, de la forma,

$$\alpha^* = \arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(x_i x_j) + \sum_{k=1}^l \alpha_k \quad (26)$$

donde $K(x, x')$ es la función “kernel” que ejecuta la proyección no lineal en el espacio de rasgos, con las mismas restricciones señaladas por Ec. (25), $K(x, x')$ es de la forma.

$$K(x, x') = \langle \phi(x), \phi(x') \rangle \quad (26)$$

un SVC duro que implementa el HSO en el espacio de rasgos es:

$$f(x) = \text{sgn} \left(\sum_{i \in SVs} \alpha_i K(x_i, x) + b \right) \quad (27)$$

donde

$$\langle w^*, x \rangle = \sum_{i=1}^l \alpha_i y_i K(x_i, x) \quad (28)$$

$$b^* = -\frac{1}{2} \sum_{i=1}^l \alpha_i y_i [K(x_i, x_r) + K(x_i, x_r)]$$

El problema principal de aplicar o proyectar vectores de entrada a un espacio hiperdimensional se conoce como “la maldición de la dimensionalidad”, que consiste en un crecimiento potencial del tiempo de cálculo computacional. El objetivo de usar funciones “kernel” es permitir que las operaciones se ejecuten en el espacio de entrada, por tanto el producto escalar no necesita ser evaluado en el espacio de rasgos. Las funciones que califican como funciones “kernel” deben satisfacer ciertas condiciones (condiciones de Mercer) [3], [4], [5]

$$K(x, x') = \sum_m a_m \Phi_m(x) \Phi_m(x'), \quad a_m > 0 \quad (29)$$

$$\iint K(x, x') g(x) g(x') dx dx' > 0, \quad g \in L_2 \quad (30)$$

Clasificador v-SVC

La desventaja que tiene el clasificador de margen máximo (C-SVC) es que el margen de separación óptimo en el espacio de rasgos que se puede obtener es siempre uno. Es posible obtener márgenes de separación ρ menores a uno usando un clasificador con margen adaptable (no controlable directamente) a través de un parámetro (v) de control de la máquina (ver Figura 2), lo que permite tener un control entre compresión (menor número de vectores soporte con menor precisión en la clasificación) y precisión (mayor precisión en la clasificación con un mayor número de vectores soporte). Para este caso la función a minimizar es la siguiente:

$$\Phi(w, v, \rho, \xi, \gamma) = \frac{1}{2} \|w\|^2 - v\rho + \frac{1}{l} \sum_{i=1}^l \xi_i$$

Sujeta a las restricciones:

$$y^i \left[\langle w, \phi(x^i) \rangle + b \right] \geq \rho - \xi_i, i = 1, \dots, l$$

$$\xi_i \geq 0, i = 1, \dots, l.$$

$$\rho > 0$$

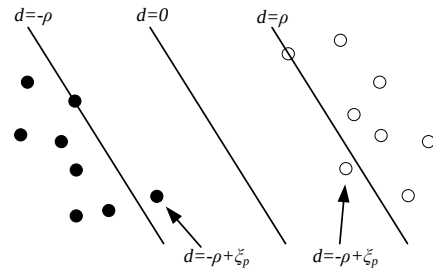


Figura 2. Clasificación de objetos con la v-SVC.

El problema dual (langragiano) de las ecuaciones anteriores es

$$\begin{aligned} \Phi(w, b, \alpha, \xi, \beta, \rho, \gamma) = & w^T \left(\frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha^i y^i w^T \phi(x^i) \right) - b \sum_{i=1}^l \alpha^i y^i \\ & + \sum_{i=1}^l \left(\frac{1}{l} - \alpha^i - \beta_i \right) \xi_i + \sum_{i=1}^l (\alpha^i - v - \gamma) \rho \end{aligned}$$

con $\alpha_i, \beta_i, \gamma_i \geq 0$

Derivando con respecto a las variables w, b, ρ, ξ se obtiene

$$w = \sum_{i=1}^l \alpha_i y_i \phi(x_i)$$

$$\sum_{i=1}^l \alpha_i = \nu + \lambda$$

$$\sum_{i=1}^l \alpha_i y_i = 0$$

$$\Phi(w) = -\frac{1}{2} w^T w$$

$$\alpha_i + \beta_i = \frac{1}{l}$$

Sustituyendo en la ecuación del langragiano:

$$\Phi(\alpha) = -\frac{1}{2} \sum_{i,j=1}^l \alpha^i \alpha^j y^i y^j \phi(x^i) \phi(x^j)$$

Esta ecuación tiene un máximo en las variables α .

Kernel de Cadenas

El kernel de cadenas está basado en la ponderación por un factor con decaimiento exponencial de subsecuencias de texto evaluado a lo largo del texto completo [2]. Una subsecuencia es cualquier secuencia ordenada de n caracteres (no necesariamente continúa) presentes en un texto.

Una de las grandes ventajas de los kernels es que no están limitados a objetos vectoriales [3], sino que son aplicables a cualquier tipo de objeto susceptible de ser representado vectorialmente, como es el caso de las cadenas. El kernel correspondiente se llama kernel de cadenas cuya aplicación fundamental es en el análisis de secuencias de ADN (cadenas compuesta por los símbolos A, T, G, C) y para la clasificación y categorización de texto.

Una de las principales desventajas de otros kernels para el análisis de texto (*bag of words*) es que no consideran el orden de las palabras en el documento [2],[3].

Definiciones: Sean Σ un alfabeto finito, Σ^n el conjunto de todas

las cadenas de longitud n , y $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$ el conjunto de todas

las cadenas finitas. La longitud de una cadena $s \in \Sigma^*$ es $|s|$, y sus elementos son $s(1), s(2), \dots, s(|s|)$; la concatenación de dos cadenas s y $t \in \Sigma^*$ se escribe st . Ahora se formarán subsecuencias u de cadenas. Dado un índice de secuencia $i = (i_1, i_2, \dots, i_{|u|})$ con $1 \leq i_1 < \dots < i_{|u|} \leq |s|$ la subsecuencia se define como $u = s(i) = s(i_1), \dots, s(i_{|u|})$. La longitud de la subsecuencia en s se define como $i_{|u|} - i_1 + 1$, si i no es contigua entonces $l(i)$ es mayor que la longitud de u ($|u|$).

El espacio de rasgos generado a partir de cadenas de longitud n se define como $\mathcal{H}_n = \mathcal{R}(\Sigma^n)$, esto significa que dicho espacio tiene una dimensión o coordenada por cada uno de los elementos de Σ^n . La proyección de todas las coordenadas en el espacio de rasgos para cada subsecuencia $u \in \Sigma^n$ se describe como $[\Phi_n(s)]_u = \sum_{i:s(i)=u} \lambda^{l(i)}$. Lambda (λ) es un factor de

decaimiento entre cero (0) y uno (1), en la medida que la subsecuencia en s sea más larga menor será su contribución a $[\Phi_n(s)]_u$. La sumatoria se realiza sobre todas las subsecuencias de s iguales a u . Por ejemplo, sea el espacio $\mathcal{H}_3$ generado por `asd`, en este caso $[\Phi_3(\text{Nasdaq})]_{\text{asd}} = \lambda^3$, mientras que $[\Phi_3(1\underline{a}ss\underline{s} \underline{d}as)]_{\text{asd}} = \lambda^5$.

El kernel inducido por la proyección Φ_n toma la forma:

$$k_n(s, t) = \sum_{u \in \Sigma^n} [\Phi_n(s)]_u [\Phi_n(t)]_u = \sum_{u \in \Sigma^n} \sum_{(i,j): s(i)=t(j)=u} \lambda^{l(i)} \lambda^{l(j)}$$

Para el caso de cadenas de diferentes longitudes n , el kernel resultante es la combinación lineal: $k = \sum_n c_n k_n$, donde $c_n \geq$

0. El número y tamaño de los términos en la sumatoria anterior dependen fuertemente de las longitudes de las cadenas s y t , por lo tanto es recomendable normalizar la proyección, $\Phi(t)/\|\Phi(t)\|$. Para las cadenas s y t el kernel normalizado será $k(s, t) / \sqrt{k(s, s)k(t, t)}$.

El cálculo de k_n se efectúa

$$k'_n(s, t) = \sum_{u \in \Sigma^i} \sum_{(i,j): s(i)=t(j)=u} \lambda^{|s|+|t|-i-j+2}, \text{ para } i = 1, 2, \dots, n-1$$

Usando la cadena de longitud uno, $x \in \Sigma^1$, con las siguientes recursiones permiten el cálculo de $k_n(s, t)$ para $n = 1, 2, \dots$:

$$k'_0(s, t) = 1 \text{ para } s, t$$

$$k'_i(s, t) = 0 \text{ si } \min(|s|, |t|) < i$$

$$k_i(s, t) = 0 \text{ si } \min(|s|, |t|) < i$$

$$k'_i(sx, t) = \lambda k'_i(s, t) + \sum_{j:t_j=x} k'_{i-1}(s, t[1, \dots, j-1]) \lambda^{|t|-j+2}, \text{ para } i = 1, 2, \dots, n-1$$

$$k_n(sx, t) = k_n(s, t) + \sum_{j:t_j=x} k'_{n-1}(s, t[1, \dots, j-1]) \lambda^2$$

3. PARTE EXPERIMENTAL

Diseño del clasificador de texto basado en SVM

Se seleccionó una lista de tesis de pregrado correspondientes a las carreras de Ingeniería Eléctrica e Ingeniería Civil de la Facultad de Ingeniería de la Universidad de Carabobo, las cuales fueron suministradas por la Dirección de Biblioteca de dicha facultad; esta lista de tesis corresponden al lapso de los años 1997 al 2006 (ambos años inclusive). Se procedió a crear un corpus de entrenamiento conformado por ciento sesenta palabras (160) con subconjuntos de ochenta (80) palabras relevantes, de cada clase, que mejor definieran las áreas (clases) respectivas de las tesis desarrolladas. Un primer conjunto de pruebas se formó con los doscientos cincuenta y nueve (259) títulos de tesis del área de ingeniería eléctrica y trescientas doce (312) títulos de tesis del área de ingeniería civil. Se eliminaron de los títulos de las tesis en el conjunto de prueba los caracteres: *dos puntos* (:), *coma* (,), *punto y coma* (;), *punto* (.), *comillas simples* (') y *dobles* ("), *guión* (-) ya que no ofrecen ningún tipo de información y aumentan la dimensión del espacio de rasgos (número de rasgos). Se sustituyeron las vocales acentuadas por vocales no acentuadas, con la finalidad de reducir la dimensión del espacio de rasgos. En este caso se mantuvieron las palabras copulativas, artículos definidos y no definidos y preposiciones.

La librería de SVM usada fue la LIBSVM [7]. Dicha librería es de código abierto y, no cuenta con un kernel de cadenas, por lo tanto, se le incorporó el algoritmo descrito más arriba.

TABLA I
DESEMPEÑO DEL CLASIFICADOR VARIANDO ν

Parámetro			Predicción (%)			N° Vect Sop		ρ
v-SVC	Kernel		Global	Eléctrica	Civil	nSV	nBSV	ρ
nu	λ	k						
0.3	0.7	3	63.92	96.53	36.86	137	5	0.2389
0.5	0.7	3	57.27	97.3	24.04	139	31	0.3438
0.7	0.7	3	48.51	98.84	6.73	144	72	0.5033
0.2	0.7	3	64.97	95.37	39.74	137	3	0.2129
0.1	0.7	3	65.32	94.6	41.03	136	0	0.2103
0.05	0.7	3	65.15	94.98	40.38	136	0	0.2132
0.01	0.7	3	64.97	96.91	38.46	125	0	0.2340
Valores Promedios								
			61.44	96.36	32.46			

Como se observa en la Tabla I, el desempeño con el conjunto de prueba global (título de tesis de ambas áreas) el desempeño fue regular (promedio de 61.44%), sin embargo, el promedio de precisión del conjunto de pruebas con tesis solo del área de ingeniería eléctrica fue bastante alto (96.36%). Por el contrario la predicción del clasificador con un conjunto de prueba de tesis del área de civil solamente fue de un 32.46%, esto se debe a que el seleccionador maneja ampliamente los términos del área de ingeniería eléctrica y no así los términos del área de civil. No obstante, el mejor valor de nu para este ensayo fue de 0.1.

El siguiente ensayo se varió lambda fijando el mejor valor de nu obtenido en la fase de prueba anterior.

TABLA II
DESEMPEÑO DEL CLASIFICADOR VARIANDO λ

Parámetro			Predicción (%)			N° Vect Sop		ρ
v-SVC	Kernel		Global	Eléctrica	Civil	nSV	nBSV	ρ
nu	λ	k						
0.1	0.7	3	65.32	94.6	41.03	136	0	0.2103
0.1	0.9	3	65.32	94.6	41.03	137	0	0.2076
0.1	0.5	3	62.87	96.14	35.26	127	0	0.2457
0.1	0.4	3	55.52	95.75	22.11	79	1	0.2323
0.1	0.95	3	65.32	94.6	41.03	136	0	0.2076
0.1	0.99	3	65.32	94.6	41.03	136	0	0.2075
0.1	0.8	3	65.32	94.6	41.03	136	0	0.2084
0.1	0.75	3	65.32	94.6	41.03	136	0	0.2089
Valores Promedios								
			61.24	96.50	32.80			

La Tabla II demuestra en este experimento el impacto que tiene λ sobre el desempeño del clasificador. Prácticamente valores de lambda mayores a 0.7 producen una precisión (65.32%) constante en la predicción de las clases. Valores de lambda menores 0.5 producen una mala clasificación de las áreas. En promedio el desempeño global fue prácticamente igual al primer ensayo, con una ligera mejora en la predicción de la clase de ingeniería civil. Se tomó como valor óptimo de lambda de 0.7 para realizar la prueba siguiente variando la longitud de la subcadena (k).

La Tabla III muestra unos resultados bien interesantes, ya que a pesar de haber obtenido una precisión global del 60.28% en promedio, la precisión global para k igual a 4 fue alta (73.56%) y para el conjunto de prueba con tesis de civil solamente, la precisión mejoró notablemente (56.10%). La precisión en la predicción de la clase de eléctrica en promedio fue siempre alta. Se observa que la longitud de la subcadena k tiene un peso bien importante en el desempeño del clasificador de texto, prácticamente k “compensa” la debilidad presente en el corpus de entrenamiento debido a la falta de conocimiento en

la selección de las palabras o frases más adecuadas correspondientes a la clase de títulos pertenecientes a ingeniería civil.

TABLA III
DESEMPEÑO DEL CLASIFICADOR VARIANDO k

Parámetro			Predicción (%)			N° Vect Sop		ρ
v-SVC	Kernel		Global	Eléctrica	Civil	nSV	nBSV	ρ
nu	λ	k						
0.1	0.7	3	65.32	94.6	41.03	136	0	0.2103
0.1	0.7	4	73.56	94.6	56.10	140	3	0.1720
0.1	0.7	2	45.36	99.61	0.32	103	0	0.6894
0.1	0.7	5	61.82	96.14	33.33	118	12	0.6858
0.1	0.7	6	55.34	81.47	33.65	42	14	0.1956
Valores Promedios								
			60.28	93.28	32.89			

En el siguiente ensayo se creó un nuevo conjunto de pruebas a partir del conjunto anterior, al cual se le eliminó el siguiente conjunto de palabras: *el, las, los, y, la, un, en, a, e, del, de, para, por*. Las Tablas IV, V y VI muestran los resultados obtenidos. Hay que recalcar que no se hizo un nuevo entrenamiento, únicamente se probaron los clasificadores obtenidos en el conjunto de ensayos anteriores.

TABLA IV
DESEMPEÑO DEL CLASIFICADOR VARIANDO ν

Parámetro			Predicción (%)			N° Vect Sop		ρ
v-SVC	Kernel		Global	Eléctrica	Civil	nSV	nBSV	ρ
nu	λ	k						
0.3	0.7	3	66.90	-	-	137	5	0.2389
0.5	0.7	3	64.10	-	-	139	31	0.3438
0.7	0.7	3	53.77	-	-	144	72	0.5033
0.2	0.7	3	68.48	-	-	137	3	0.2129
0.1	0.7	3	68.30	-	-	136	0	0.2103
0.05	0.7	3	68.13	-	-	136	0	0.2132
0.01	0.7	3	67.60	-	-	125	0	0.2340
Valores Promedios								
			65.32	-	-			

TABLA V
DESEMPEÑO DEL CLASIFICADOR VARIANDO λ

Parámetro			Predicción (%)			N° Vect Sop		ρ
v-SVC	Kernel		Global	Eléctrica	Civil	nSV	nBSV	ρ
nu	λ	k						
0.1	0.7	3	66.90	-	-	136	0	0.2103
0.1	0.9	3	68.48	-	-	137	0	0.2076
0.1	0.5	3	67.95	-	-	127	0	0.2457
0.1	0.4	3	57.10	-	-	79	1	0.2323
0.1	0.95	3	68.48	-	-	136	0	0.2076
0.1	0.99	3	68.48	-	-	136	0	0.2075
0.1	0.8	3	68.48	-	-	136	0	0.2084
0.1	0.75	3	68.48	-	-	136	0	0.2089
Valores Promedios								
			64.16	-	-			

TABLA VI
DESEMPEÑO DEL CLASIFICADOR VARIANDO k

Parámetro			Predicción (%)			N° Vect Sop		ρ
v-SVC	Kernel		Global	Eléctrica	Civil	nSV	nBSV	ρ
nu	λ	k						
0.1	0.7	3	66.90	-	-	136	0	0.2103
0.1	0.7	4	77.06	-	-	140	3	0.1720
0.1	0.7	2	47.99	-	-	103	0	0.6894
0.1	0.7	5	64.27	-	-	118	12	0.6858
0.1	0.7	6	61.30	-	-	42	14	0.1956
Valores Promedios								
			63.50	-	-			

Los resultados de las Tablas IV, V y VI muestran que el clasificador mejoró ligeramente al eliminar el conjunto de palabras que cumplen funciones de enlace y preposición en los títulos. En promedio la mejora de la predicción fue de 3.34 puntos porcentuales.

4. CONCLUSIONES

Esta investigación demostró el excelente desempeño del clasificador de texto basado en SVM y el kernel de cadenas.

Las pruebas demostraron que a pesar de no haber tenido un conjunto de palabras o frases adecuadas para definir el área de ingeniería civil en el corpus de entrenamiento, se logró mejorar la predicción (con pocos vectores de soporte de error) ajustando los parámetros de la SVM y del kernel de cadenas.

La eliminación de palabras que se utilizan como conectores en el idioma español y las preposiciones, presentes en el conjunto de datos de prueba, mejoró ligeramente el desempeño del clasificador, lo que conlleva a decidir si vale la pena o no eliminar dichas palabras.

5. REFERENCIAS

- [1] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, Vol. 18 No. 11, 1975, pages 613-620.
- [2] Huma Lodhi, Craig Saunders, John Shawe Taylor, Nello Cristianini and Chris Watkins, "Text Classification using String Kernels," in *Journal of Machine Learning Research*, 2002, pages 419-444.
- [3] Ralf Herbrich, *Learning Kernel Clasifiers*, The MIT Press Cambridge, Massachusetts London, England, 2002.
- [4] Bernhard Schölkopf and Alexander J. Smola, *Learning With Kernels, Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press Cambridge, Massachusetts London, England, 2002.
- [5] V. N. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer-Verlag, 1995.
- [6] Nello Cristianini and John Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge: Cambridge University Press, 2000.
- [7] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a Library for Support Vector Machines, 2001. [Online]. Software available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.