APRENDIZAJE ACTIVO PARA MÁQUINAS DE VECTORES DE RELEVANCIA

RESUMEN

Este artículo presenta una técnica de aprendizaje activo para máquinas de vectores de relevancia. Uno de los métodos más utilizados para entrenar clasificadores activos utiliza máquinas de vectores de soporte, las cuales poseen algunas desventajas desde el punto de vista práctico. La novedad del método propuesto consiste en mantener todas las ventajas del aprendizaje activo, reemplazando la máquina de vectores de soporte por un clasificador con propiedades similares que no sufra de sus desventajas, esto es, la máquina de vectores de relevancia, obteniendo de esta manera un método de aprendizaje activo mucho más robusto. Los resultados obtenidos sobre bases de datos estándar y de bioseñales empleando el método propuesto, muestran desempeños satisfactorios y comparables con relación al aprendizaje activo para máquinas de vectores de soporte.

PALABRAS CLAVES: Aprendizaje activo, Máquinas de Vectores de relevancia, Máquinas de vectores de soporte, Clasificación, Espacio versión.

ABSTRACT

This paper introduces an active learning technique for relevance vector machines. One of the most used methods for training active learners makes use of support vector machines as classifiers. However, despite its success, several disadvantages can be advised in the support vector methodology. The novelty of the proposed method consists on keeping all of the advantages of active learning, but replacing support vector machines with another classifier that does not suffer of its disadvantages, aiming to obtain an active learning method even more robust. Results over standard and biosignal datasets show satisfactory and comparable performances of the proposed method in contrast to classic support vector machines active learning.

KEYWORDS: *Active learning, Relevance vector machines, support vector machines, classification, version space.*

RICARDO HENAO*
Ingeniero Electrónico, Ms.Eng.
Profesor Auxiliar
Universidad Tecnológica de Pereira
rhenao@utp.edu.co

JORGE HUMBERTO SANZ**
Ingeniero Eléctrico
Profesor Asociado
Universidad Tecnológica de Pereira
jsanz@utp.edu.co

EDISON DUQUE*
Ingeniero Electrónico
Profesor Asistente
Universidad Tecnológica de Pereira
eduque@utp.edu.co

* **Grupo de investigación "LIDER"**
** **Grupo de investigación Sistemas de puesta a tierra**

1. INTRODUCCIÓN

Uno de los inconvenientes en las tareas de aprendizaje supervisado es que en muchos de los casos etiquetar las observaciones para construir los conjuntos de entrenamiento resulta ser muy costoso y usualmente requiere de mucho tiempo por parte de especialistas en el tema que se está considerando. Por lo tanto, encontrar maneras de minimizar la cantidad de observaciones etiquetadas requeridas para entrenar un clasificador puede resultar beneficioso. En el aprendizaje supervisado la estrategia que se utiliza convencionalmente es tomar aleatoriamente un subconjunto de la totalidad de pares observación-etiqueta para entrenar algún clasificador y los restantes para efectuar un proceso de validación. En el aprendizaje activo de otro lado, se obtiene cierta flexibilidad en el sentido que las observaciones necesarias para entrenar el clasificador se incrementan de acuerdo a un criterio de validación, con esto por ende, se reduce la necesidad de grandes cantidades de observaciones etiquetadas.

En particular, el aprendizaje activo es un procedimiento en el cual se seleccionan las observaciones entre un conjunto sin etiquetar para entrenar el clasificador, lo cual es bastante conveniente si se cuenta con una gran cantidad de datos no etiquetados. El problema reside en la manera como se seleccionan las observaciones adecuadas. Una de las primeras soluciones a este problema se presenta en [1], se denomina muestreo de incertidumbre, está motivado por la teoría de aprendizaje computacional [2] y utiliza un clasificador probabilístico convencional. Recientemente en [3] se presenta un algoritmo de aprendizaje activo para máquinas de vectores de soporte (SVM) basado en la noción de espacio versión, mostrando experimentalmente que las observaciones etiquetadas requeridas para entrenar el clasificador se reducen considerablemente.

Si bien las SVM han demostrado un gran potencial en tareas de clasificación principalmente por su buena capacidad de generalización, debido a que están fundamentadas en la teoría de aprendizaje estadístico [2], poseen varias desventajas desde el punto de vista práctico (ver sección 4). Las máquinas de vectores de relevancia (RVM) son un tratamiento bayesiano de una función de decisión similar a la de una SVM, pero que no sufre de

Fecha de Recepción: 30 Enero de 2006
Fecha de Aceptación: 12 Julio de 2006

algunas de sus desventajas [4]. En este trabajo se presenta un algoritmo de aprendizaje activo basado en β] que emplea una máquina de vectores de relevancia en vez de una máquina de vectores de soporte con miras a eliminar las desventajas de esta última y proporcionar una herramienta de aprendizaje activo mucho más robusta.

Este artículo está organizado como sigue: las secciones 2 y 3 contienen revisiones de la teoría de RVM y aprendizaje activo respectivamente. En la sección 4, se presenta el método propuesto. En la sección 5 se reportan los resultados experimentales obtenidos para bases de datos estándar y de bioseñales; y finalmente en la sección 6 las conclusiones y discusión del trabajo.

2. MÁQUINAS DE VECTORES DE RELEVANCIA

Partiendo de una formulación probabilística convencional para un conjunto de entrenamiento compuesto por pares observación-etiqueta $\{\mathbf{x}_n, t_n\}_{n=1}^N$, el modelo de una máquina de vectores de soporte [5], corresponde a una familia de funciones de la forma

$$y(\mathbf{x}_n, \mathbf{w}) = \mathbf{w} \cdot f(\mathbf{x}_n) \quad (1)$$

donde $x_n \in \mathbb{R}^d \equiv \mathcal{L}$ denota un vector de características, $\mathbf{w} = \{w_0, ..., w_N\}$ representa un vector de pesos que es calculado a partir del conjunto de entrenamiento y $f(\mathbf{x}_n)$ es una función que mapea las observaciones del espacio de entrada $\mathcal{L}$ de dimensión baja a un espacio de Hilbert $\mathcal{H}$ generalmente de dimensión alta, $f(\mathbf{x}_n): \mathcal{L} \to \mathcal{H}$ donde se supone que el conjunto de entrenamiento es linealmente separable. Una vez la SVM está entrenada, se espera que la evaluación del signo de (1) para un $\mathbf{x}$ dado suministre la etiqueta de la clase, asumiendo que el problema de clasificación es binario.

El entrenamiento de una SVM consiste en ajustar los valores de $\mathbf{w}$. Asumiendo que $t_n \in \{-1,1\}$, el vector $\mathbf{w}$ se puede escribir como una combinación lineal [5]

$$\mathbf{w} = \sum_{i=0}^{N} b_i y_i f(\mathbf{x}_i)$$

de tal manera que los b_i pueden ser encontrados minimizando la función cuadrática

$$L(b) = \sum_{i=0}^{N} b_i - \frac{1}{2} \sum_{i,j=0}^{N} b_i b_j f(\mathbf{x}_i) \cdot f(\mathbf{x}_j) \quad (2)$$

La expresión en (2) está sujeta a $0 \le b_i \le C$ donde C es una constante que hace el clasificador tolerante ante la no separabilidad de las observaciones en $\mathcal{H}$. Por ejemplo, cuando $C \to \infty$ el algoritmo de entrenamiento asume que los datos son separables, a este caso se le conoce como margen rígido. Aquellos puntos $\mathbf{x}_i$ para los cuales $b_i > 0$ se les denomina vectores de soporte y corresponden a los puntos más cercanos al hiperplano de decisión en $\mathcal{H}$. El producto interno implícito $f(\mathbf{x}_i) \cdot f(\mathbf{x}_j)$ se calcula explícitamente utilizando una función $k(x_i, x_j)$ conocida como kernel, la cual debe satisfacer las condiciones de Mercer [6]. Una de las funciones kernel más utilizadas es la gaussiana RBF definida como

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-g \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2) \quad (3)$$

donde el parámetro g controla el ancho del kernel y debe ser ajustado para un adecuado desempeño de la SVM.

Partiendo de una formulación probabilística convencional, el modelo de una máquina de vectores de relevancia corresponde a [4]

$$t_n = y(\mathbf{x}_n, \mathbf{w}) + \mathbf{e}_n$$

donde los $\mathbf{e}_n$ son muestras independientes de un proceso de ruido asumido $\mathcal{N}\left(0, s_n^2\right)$ y $y(\mathbf{x}_n, \mathbf{w})$ es la función de predicción de una máquina de vectores de soporte [5] que a partir de (1) se puede escribir como

$$y(\mathbf{x}_n, \mathbf{w}) = \sum_{i=1}^{N} w_i K(x_i, x_n) + w_0 \quad (4)$$

Teniendo en cuenta que el modelo en (1) es lineal, para el caso de clasificación se puede generalizar aplicando la función logística sigmoidal $s(y) = 1/(1+e^{-y})$ a $y(\mathbf{x}_n, \mathbf{w})$ y adoptando una distribución de Bernoulli para $P(t|\mathbf{x})$, con esto, la verosimilitud se puede escribir como

$$p(t|\mathbf{w}) = \prod_{n=1}^{N} s\{y(\mathbf{x}_n, \mathbf{w})\}^{t_n} [1 - s\{y(\mathbf{x}_n, \mathbf{w})\}]^{1-t_n} \quad (5)$$

donde $t_n = \{0,1\}$, siguiendo la especificación probabilística.

Ya que el modelo posee tantos parámetros como pares observación-etiqueta en el conjunto de entrenamiento, es de esperarse que la estimación de máxima verosimilitud para $\mathbf{w}$ a partir de (2) conduzca a un sobreentrenamiento severo. Para prevenir esto último, desde la perspectiva bayesiana, se pueden restringir los parámetros definiendo una distribución de probabilidad a priori explícita sobre éstos, utilizando una distribución gaussiana con media cero sobre $\mathbf{w}$ como

$$p(\mathbf{w}|\mathbf{a}) = \prod_{i=1}^{N} \mathcal{N}(0, a_i^{-1}) \quad (6)$$

donde $\mathbf{a}$ es un vector de $N+1$ hiperparámetros. Esta formulación es común en la determinación automática de relevancia (ARD) [7] y tiene como finalidad hacer que la probabilidad posterior del modelo se concentre en valores muy grandes de algunos de los a_i haciendo por consiguiente que sus correspondientes w_i se hagan cero, con lo cual la RVM resulta ser rala, esto es, que sólo unos cuantos valores en $\mathbf{w}$ son diferentes de cero. Intuitivamente, introducir $N+1$ parámetros adicionales al modelo sabiendo que ya se tienen demasiados puede parecer una contrariedad, sin embargo, desde la perspectiva bayesiana, con una formulación correcta del

problema no existe ningún inconveniente desde el punto de vista metodológico [8].

Debido a que no se puede obtener una expresión analítica para $p(\mathbf{w}|\mathbf{t},\mathbf{a})$, se puede utilizar un procedimiento de aproximación basado en el método de Laplace [9]. Dados los valores actuales de a_i, los pesos más probables $\mathbf{w}^{MP}$ se pueden encontrar para el punto de localización del modo de la distribución posterior, esto es, considerando que $p(\mathbf{w}|\mathbf{t},\mathbf{a}) \propto P(\mathbf{t}|\mathbf{w})\,p(\mathbf{w}|\mathbf{a})$ a partir de (5) y (6), corresponde a encontrar el máximo sobre $\mathbf{w}$ de

$$\log\{P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\mathbf{a})\} = \sum_{n=1}^{N}[t_n \log y_n + (1-t_n)\log(1-y_n)] - \frac{1}{2}\mathbf{w}'\mathbf{A}\mathbf{w} \quad (7)$$

donde $y_n = s\{y(\mathbf{x}_n, \mathbf{w})\}$ y $\mathbf{A} = \mathrm{diag}(a_0,...,a_N)$. Dado que el método de Laplace es una aproximación cuadrática del posterior logarítmico alrededor de su modo, se requiere diferenciar (7) dos veces para obtener

$$\nabla_w \nabla_w \log\{P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\mathbf{a})\}|_{\mathbf{w}^{MP}} = -(\mathbf{F}'\mathbf{B}\mathbf{F} + \mathbf{A}) \quad (8)$$

donde $\mathbf{B} = diag(n_1,...,n_N)$ es matriz diagonal con elementos $n_n = y_n(1-y_n)$ y $\mathbf{F} = [f(\mathbf{x}_1),...,f(\mathbf{x}_N)]'$. Luego, el término de la derecha en (8) se puede negar e invertir para obtener la covarianza $\mathbf{S}$ para una aproximación gaussiana del posterior sobre los pesos centrada en $\mathbf{w}^{MP}$, esto es,

$$\mathbf{S} = (\mathbf{F}'\mathbf{B}\mathbf{F} + \mathbf{A})^{-1} \quad (9)$$

En el modo de $p(\mathbf{w}|\mathbf{t},\mathbf{a})$, usando (8) y el hecho de que el gradiente de $\log p(\mathbf{w}|\mathbf{t},\mathbf{a})$ es igual a cero en $\mathbf{w}^{MP}$ [4],

$$\mathbf{w}^{MP} = \mathbf{S}\mathbf{F}'\mathbf{B}\mathbf{t} \quad (10)$$

Utilizando $\mathbf{S}$ y $\mathbf{w}^{MP}$ de (9) y (10), los hiperparámetros $\mathbf{a}$ se pueden actualizar usando [4]

$$\tilde{a}_i = \frac{1 - a_i \mathbf{S}_{ii}}{\mathbf{w}_i^{MP}} \quad (11)$$

El algoritmo procede calculando repetidamente (11) luego de las correspondientes actualizaciones de (9) y (10) hasta que algún criterio de convergencia se satisfaga. Cabe anotar que no es necesario actualizar el parámetro de varianza s^2 del proceso de ruido $\mathbf{e}_n$ debido a que $s_n^2 = 1/n_n$ [4].

3. APRENDIZAJE ACTIVO

Partiendo de la función de predicción de una SVM en (4) y teniendo en cuenta que el vector $\mathbf{w}$ tiene norma unitaria utilizando el kernel RBF en (3), la familia de puntos $\mathbf{w}$ tiene la forma de una hiperesfera de radio unitario, $\mathcal{W}$. El subespacio de dicha hiperesfera, cuyos elementos separan efectivamente los datos de entrenamiento dadas sus etiquetas se denomina espacio versión $\mathcal{V}$ [10]. A partir de (4) es posible observar que los elementos en $\mathcal{W}$ corresponden a hiperplanos en $\mathcal{H}$ y viceversa. El proceso de entrenamiento de una SVM puede ser asociado a la búsqueda de la hiperesfera más grande que se pueda inscribir en el espacio versión. Dicha esfera tiene radio $1/\|\mathbf{w}\|$ y los planos que representan los puntos de entrenamiento, los cuales son tangentes a dicha esfera, se dominan vectores de soporte.

Se puede demostrar que la minimización del área de $\mathcal{V}$, conlleva a disminuir el número de clasificadores que pueden separar de manera efectiva el conjunto de entrenamiento, reduciendo además, la posibilidad de sobreentrenamiento de la SVM [3].

El aprendizaje activo es una técnica empleada en el entrenamiento de clasificadores de manera transductiva, esto es, que a partir de un conjunto de vectores sin etiquetar, se puedan escoger la menor cantidad de puntos posibles para la construcción del clasificador de modo que para un conjunto etiquetado se obtenga una alta capacidad discriminante.

Partiendo de un conjunto sin etiquetar U, una máquina de aprendizaje activo consta de tres componentes $(\mathbf{X}, f, q)$: la primera es un conjunto de observaciones etiquetadas (también puede contener observaciones sin etiquetar de U), la segunda el clasificador $f: \mathbf{X} \rightarrow \{0,1\}$ entrenado sobre el conjunto actual de datos etiquetados $\mathbf{X}$ y la última componente q es una función oráculo que decide cuál elemento de U debe ser incluido a continuación en $\mathbf{X}$. Con esto último se tiene que una máquina de aprendizaje activo retorna un clasificador f luego de cada evaluación de q. Para el caso en que la SVM se utiliza como clasificador de la máquina de aprendizaje activo, lo que se quiere es reducir tan pronto como sea posible el área del espacio versión, de modo que también se reduzcan los posibles $\mathbf{w}$ que clasifiquen efectivamente los datos de entrenamiento. En este caso, dado el conjunto de datos sin etiquetar U, se conforma $\mathbf{X}$ con una observación de cada clase. A partir de este conjunto inicial se entrena una SVM que se puede denominar como SVM_2. El espacio versión $\mathcal{V}_2$ generado por SVM_2 sirve como base para que la función oráculo q seleccione el siguiente punto a etiquetar e incluir en $\mathbf{X}$ que corresponde a la observación en U que divida en cuanto sea posible a la mitad el área del espacio versión $\mathcal{V}_2$. Con este nuevo $\mathbf{X}$, ahora con tres observaciones se entrena un nuevo clasificador SVM_3 con espacio versión asociado $\mathcal{V}_3$. El proceso se continua iterativamente seleccionando un nuevo punto para $\mathbf{X}$ de modo que luego de cada consulta a la función q el área del espacio versión resultante se haga cada vez más pequeña, esto es $\mathcal{V}_1 \supset \mathcal{V}_2 \supset ... \supset \mathcal{V}_i$ con $\mathcal{V}_i$ el espacio versión resultante luego de i consultas a la función q e i clasificadores SVM entrenados.

Debido a que dada una observación sin etiquetar $\mathbf{x}$ de U, no es práctico calcular explícitamente el tamaño de los nuevos espacios versión $\mathcal{V}^0$ y $\mathcal{V}^1$, esto es, los espacios versión obtenidos asumiendo que $\mathbf{x}$ es etiquetado como 0 y 1 respectivamente, existen principalmente dos formas de hacer una aproximación a este tamaño [3]:

Margen Simple: Este criterio consiste en seleccionar la observación que se encuentre más cerca al hiperplano $\mathbf{w}$ en $\mathcal{H}$, esto es la observación para la cual $|\mathbf{w} \cdot f(\mathbf{x})|$ sea mínimo, con lo cual se espera que entre más central esté un hiperplano en $\mathcal{W}$, más cercano estará del centro del espacio versión, por lo tanto dividirá mejor en dos partes el espacio versión actual. El inconveniente de este criterio reside en que asume que $\mathbf{w}_i$ está ubicado aproximadamente en el centro de $\mathcal{V}_i$ y que este es aproximadamente simétrico, por lo tanto depende de la forma que tenga el espacio versión que en muchos de los casos tiende a ser oblonga [3].

Margen MaxMin: Este criterio parte del hecho de que $\mathbf{w}_i$ es el centro de la hiperesfera más grande que se puede inscribir en $\mathcal{V}_i$ y que el radio m_i de dicha hiperesfera es $1/\|\mathbf{w}\|$, por lo tanto m_i se puede utilizar como indicador del tamaño del espacio versión. Dado un punto $\mathbf{x}$ se pueden obtener dos espacios versión $\mathcal{V}^0$ y $\mathcal{V}^1$ como resultado de entrenar dos SVM asumiendo que $\mathbf{x}$ pertenece a la clase 0 y a la clase 1 respectivamente. Debido a que se quiere dividir el espacio versión en dos partes de la manera más equitativa posible, se requiere que las áreas de $\mathcal{V}^0$ y $\mathcal{V}^1$ sean similares. Como el área de $\mathcal{V}$ es proporcional a m, es de esperarse que $\min(m_0, m_1)$ sea pequeño si sus áreas correspondientes son muy diferentes. Con esto, el criterio MaxMin consiste en calcular m_0 y m_1 para cada $\mathbf{x}$ en U, para luego escoger la observación sin etiquetar para la cual $\min(m_0, m_1)$ es máximo.

Los experimentos presentados en [3] muestran resultados superiores en cuanto a tasas de clasificación y estabilidad utilizando margen MaxMin, en lugar de margen simple; sin embargo, cabe anotar que el costo computacional de calcular el margen MaxMin es mucho más alto que el requerido por el margen simple, considerando que en el primero es necesario entrenar dos SVM por cada punto en U y para cada iteración del algoritmo de aprendizaje activo.

4. MÉTODO PROPUESTO

Anteriormente se mencionó que las RVM no sufrían de algunas de las desventajas prácticas de la SVM, siendo esta la razón principal por la cual se eligieron las primeras para desarrollar un nuevo método de aprendizaje activo más robusto. En particular, las desventajas de las SVM son las siguientes:

- Aunque son relativamente ralas, las SVM utilizan deliberadamente una cantidad arbitraria de funciones base, considerando que el número de vectores de soporte crece linealmente con el tamaño del conjunto de entrenamiento; además, se puede demostrar que el número de vectores de soporte es una cota superior del riesgo del clasificador [11].
- La predicción del clasificador no tiene significado probabilístico.
- Las funciones kernel deben satisfacer las condiciones de Mercer [6].

El problema de utilizar una RVM como clasificador activo reside en que el criterio de margen MaxMin no puede ser utilizado debido a que en las RVM no existe noción tal, sin embargo, partiendo de la idea de margen simple, para el caso de las RVM una forma de aproximar el espacio versión sería seleccionar la observación con menos probabilidad de pertenecer a alguna de las dos clases, esto es, la observación que esté más cerca de la mitad de la trayectoria de $s\{y(\mathbf{x}_n, \mathbf{w})\}$, que en el caso de la función logística sigmoidal sería 1/2. Esto último puede ser visto como una aproximación al criterio de margen simple, con la diferencia que dada la propiedad de las RVM de ser ralas se puede esperar que los resultados sean mejores en cuanto a precisión de clasificación y estabilidad.

5. RESULTADOS EXPERIMENTALES

Con la finalidad de comprobar el funcionamiento del método propuesto, se realizaron pruebas sobre cinco diferentes bases de datos, tres de ellas estándar tomadas del catálogo UCI de aprendizaje de máquina [12] y las dos restantes reales para la identificación de estados funcionales en bioseñales, una para la identificación de patologías en voz [13] y la otra para la identificación de cardiopatía Isquémica [14]. En todos los casos, de la totalidad de cada base de datos se tomaron 60 muestras para el entrenamiento del clasificador activo. A modo de marco comparativo, los resultados del método propuesto se comparan con el método clásico de aprendizaje activo para SVM. Para la selección del modelo se realizó validación cruzada sobre un conjunto de valores para el parámetro g del kernel gaussiano, en los resultados sólo se presentan aquellos con mejor precisión de validación. Teniendo en cuenta que el algoritmo puede ser sensible a la inicialización, cada experimento se ejecutó cuatro veces, por lo cual, además de la precisión de validación se presenta su desviación estándar.

5.1 Conjuntos Estándar

Las tres bases de datos estándar utilizadas hacen parte de UCI, un catálogo ampliamente utilizado para la prueba de

algoritmos de aprendizaje de máquina. En la tabla (1) se muestra la estructura de cada una de ellas.

Nombre	Observaciones	Características
Thyroid	215	5
Herat	270	13
Digit	100	64

Tabla 1. Estructura de los conjuntos estándar

Los resultados para las bases de datos estándar se presentan en la tabla (2), incluyendo: porcentaje en la precisión de validación con desviación estándar, número de vectores de soporte o de relevancia según sea el caso (RV/SV) y el valor de g del kernel gaussiano.

	Método	Precisión	RV/SV	g
Thyroid	SVM	99.07±0.00	44	0.4
	RVM	**100.00±0.00**	**8**	2.2
Heart	SVM	**88.70±0.55**	60	0.1
	RVM	87.38±0.23	**6**	6.0
Digit	SVM	99.00±0.00	60	0.08
	RVM	**100.00±0.00**	**9**	10.0

Tabla 2. Resultados bases de datos estándar

De la tabla (2) se puede observar que para las tres bases de datos bajo prueba, la RVM obtuvo una considerable menor cantidad de vectores de relevancia con respecto a la SVM, de lo cual se puede aseverar que en términos de generalización, las RVM son un mejor clasificador. Sólo en el caso de Heart, las SVM obtuvieron una mejor precisión de clasificación, aunque la diferencia con respecto a la RVM no es substancial.

5.2 Identificación de Patologías en Voz

Las muestras de voz pertenecientes a esta base de datos de la Universidad Nacional de Colombia sede Manizales fueron obtenidos de 91 pacientes, 40 de ellos con voz normal y los restantes con voces presentando alguna patología. El conjunto de características fue extraído utilizando las dos componentes más grandes de cada uno de los 6 niveles de descomposición de un análisis wavelet *db8* empleando fonaciones de las cinco vocales del idioma español [13]. El conjunto finalmente cuenta con 91 observaciones y 60 características, los resultados para esta base de datos se presentan en la figura (1) y en la tabla (3).

Método	Precisión	RV/SV	g
SVM	**100.00±0.00**	60	0.1
RVM	**100.00±0.00**	**8**	12.0

Tabla 3. Resultados bases de datos de voz

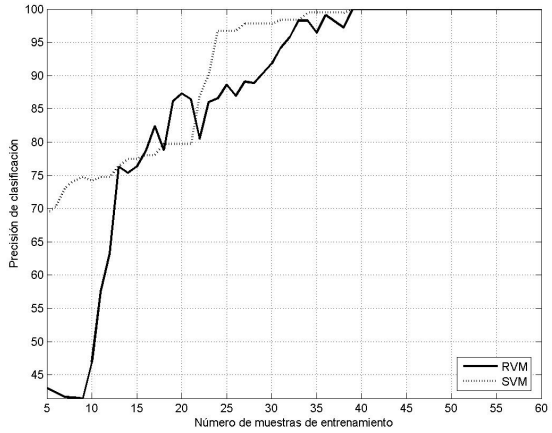

Figura 1. Resultados base de datos de voz

Los resultados obtenidos muestran una precisión de clasificación similar para ambos métodos, sin embargo, en el caso de las RVM se obtuvo una menor cantidad de vectores de relevancia con respecto a los 60 requeridos por la SVM. Además cabe anotar, que la velocidad de convergencia del algoritmo para el caso de SVM es relativamente mejor que para RVM. Resultados similares a los presentados en este trabajo para la misma base de datos, diferentes formas de extracción de características y clasificadores activos con SVM pueden ser encontrados en [15].

5.3 Identificación de Cardiopatía Isquémica

Para este experimento se tomaron 900 latidos normales y 900 patológicos de la base de datos ST-T europea [16]. El vector de características para cada latido se obtuvo mediante la agrupación de parámetros heurísticos, coeficientes wavelet y latidos originales tal y como se presenta en [14]. El conjunto total cuenta entonces con 1800 observaciones y 30 características. Los resultados para esta base de datos se presentan en la figura (2) y la tabla (4).

Método	Precisión	RV/SV	g
SVM	**100.00±0.00**	60	0.08
RVM	**100.00±0.00**	**52**	2.0

Tabla 4. Resultados bases de datos de ECG

Los resultados obtenidos muestran valores de precisión de clasificación iguales para ambos métodos, con un menor número de vectores de relevancia necesarios para construir el clasificador en el caso de RVM. En [17] se pueden encontrar resultados para esta base de datos utilizando clasificadores activos con SVM, a modo de referencia.

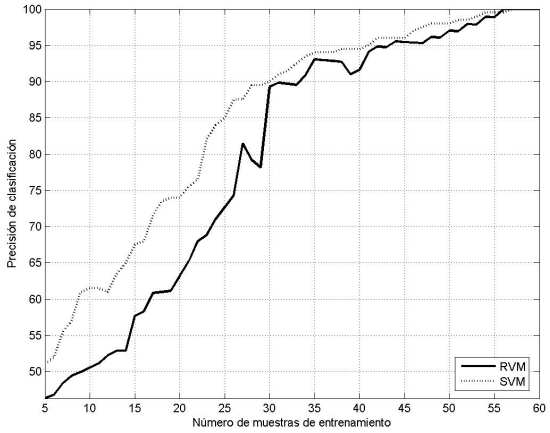

Figura 2. Resultados base de datos de ECG

6. DISCUSIÓN

En este trabajo se ha mostrado que por medio de la utilización de un clasificador activo basado en máquinas de vectores de relevancia se pueden obtener resultados comparables a aquellos obtenidos con el método en [3] para máquinas de soporte vectorial, resultando en un método más robusto considerando que elimina algunas de las desventajas de las SVM.

Considerando que el número de vectores de soporte o número de vectores de relevancia es una cota superior del riesgo del clasificador, los resultados obtenidos muestran que el método propuesto es menos sensible al sobreentrenamiento y por lo tanto debe tener mejor capacidad de generalización, teniendo en cuenta que para varios casos, el clasificador activo con SVM obtuvo un número de vectores de soporte igual a la cantidad de observaciones de entrenamiento.

Si bien en este trabajo no se mostraron resultados del método con funciones kernel diferentes a la gaussiana, existen muchas aplicaciones en las cuales se hace necesario utilizar funciones diferentes y más aun funciones que no cumplen con las condiciones de Mercer, para las cuales el método propuesto podría resultar de gran utilidad.

Aunque en este trabajo se obtuvieron resultados satisfactorios en términos de precisión de clasificación, queda todavía la pregunta de cómo seleccionar adecuadamente los parámetros del kernel y posiblemente una forma diferente de aproximar el tamaño del espacio versión para RVM con miras a mejorar la velocidad de convergencia del algoritmo.

7. AGRADECIMIENTOS

Los autores desean agradecer al Grupo de Control y Procesamiento Digital de Señales (GCPDS) de la Universidad Nacional de Colombia sede Manizales por proporcionar la base de datos para la identificación de patologías en voz.

8. BIBLIOGRAFÍA

[1] D.D. Lewis y W.A. Gale, A Sequential Algorithm to Train Text classifiers, Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval, Springer Verlag, Heidelberg, DE, pp. 3-12, 1994.

[2] V. Vapnik , Statistical Learning Theory, Wiley, NY, 1998.

[3] S. Tong and D. Koller, Support Vector Machine Active Learning with Applications to Text Classification, Journal of Machine Learning Research, vol. 2, pp. 45-66, 2001.

[4] M.E. Tipping, Sparse Bayesian Learning and the Relevance Vector Machine, Journal of Machine Learning Research, vol. 1, pp. 211-244, 2001.

[5] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Knowledge Discovery and Data Mining, vol. 2, no. 2, pp. 121-167, 1998.

[6] B. Schölkopf y A. Smola, Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond, MIT Press, Cambridge, MA, 2002.

[7] D.J.C. MacKay, Bayesian Interpolation, Neural Computation, vol. 4, no. 3, pp. 415-447, 1992.

[8] R.M. Neal, Bayesian Learning for Neural Networks, Ph.D. Thesis, Dept. of Computer Science, University of Toronto, 1994.

[9] D.J.C. MacKay, The Evidence Framework Applied to Classification Networks, Neural Computation, vol. 4, no. 5, pp. 720-736, 1992.

[10] R. Herbrich, Learning Kernel Classifiers – Theory and Algorithms, The MIT Press, 2002.

[11] V. Vapnik y O. Chapelle, Bounds on Error Expectation for Support Vector Machines, Neural Computation, vol. 12, no. 9, pp. 2013-2036, 2000.

[12] C.L. Blake y C.J. Merz., UCI Repository of Machine Learning Databases, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[13] F. Ojeda, Extracción de Características usando Transformada Wavelet en la Identificación de Voces Patológicas, Tesis de Pregrado, Universidad Nacional de Colombia, Manizales, 2003.

[14] G. Guarín y V. Montes, Extracción de Parámetros ECG en Tiempo Real Basados en Transformaciones no Lineales y Wavelets sobre DSP, Tesis de Pregrado, Universidad Nacional de Colombia, Manizales, 2004.

[15] D.A. Alvarez, R. Henao, C.G. Castellanos, J.E. Hurtado y C.L. Rengifo, Active Learning on the Classification of Voice Pathologies, Odyssey 2004: The Speaker and Language Recognition Workshop, International Speech Communication Association, Toledo, España, Mayo-Junio, 2004.

[16] A. Taddei, G. Distante, M. Emdin, P. Pisani, G.B. Moody, C. Zeelenberg y C. Marchesi, The European ST-T Database: Standard for Evaluating Systems for the Analysis of ST-T Changes in Ambulatory Electrocardiography, European Heart Journal, vol. 13, pp. 1164-1172, 1992.

[17] C.L. Rengifo, C.G. Castellanos y R. Henao, Aprendizaje Activo en la Identificación de Cardiopatía Isquémica. VIII Simposio de Tratamiento de Señales, Imagen y Visión Artificial. IEEE Signal Processing y Sociedad Colombiana de Procesamiento de Señales, Manizales, Septiembre, 2004.