MODELOS DE SELECCIÓN DE ATRIBUTOS PARA SVMS

Sebastián Maldonado
Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes
San Carlos de Apoquindo 2200, Santiago, Chile
smaldonado@uandes.cl

Richard Weber
Departamento de Ingeniería Industrial, Universidad de Chile
República 701, Santiago, Chile
rweber@dii.uchile.cl

Resumen

Recientemente los datos se han incrementado en todas las áreas del conocimiento, tanto en el número de instancias como en el de atributos. Bases de datos actuales pueden contar con decenas e incluso cientos de miles de variables con un alto grado de información tanto irrelevante como redundante. El objetivo de la selección de atributos es triple: mejorar el desempeño predictivo, implementar soluciones rápidas y menos costosas, y proveer de un mejor entendimiento del proceso subyacente que generó los datos. El método de clasificación llamado *Support Vector Machines* (SVMs) ha ganado popularidad gracias a su capacidad de generalización frente a nuevos objetos y de construir complejas funciones no lineales. Sin embargo, este método no está diseñado para identificar los atributos importantes para construir la regla discriminante. El presente trabajo busca desarrollar técnicas que permitan incorporar la selección de atributos en la formulación de SVMs, aportando eficiencia y comprensibilidad al método.
Palabras Clave: Minería de Datos, Selección de atributos, SVMs.
Main Area: OA - Otras aplicaciones en IO

Abstract

Feature selection is of considerable importance in classification. The reason for being so is threefold: to reduce the computational complexity, to gain knowledge about the process that generated the data and to improve the classifier's generalization ability. We introduce a novel algorithm for feature selection, using Support Vector Machines with kernel functions. Our method is based on a *sequential backward selection*, using the number of errors in a validation subset as the measure to decide which feature to remove in each iteration. We compare our approach with other algorithms to demonstrate its effectiveness and efficiency.
Keywords: Data mining, feature selection, SVMs.
Main Area: OA - Other applications in OR

1. Introducción

En el escenario actual, las empresas participan en un mercado muy competitivo, donde los clientes se encuentran adecuadamente informados al momento de elegir entre distintas compañías. En mercados donde esto ocurre, la empresa que posea una mayor cantidad de información relevante podrá ejecutar estrategias comerciales efectivas, sobresaliendo del resto de las compañías. Adicionalmente, la información disponible permite tomar diversas decisiones estratégicas, tales como: definir políticas de asignación de créditos en base al comportamiento histórico de clientes, diseño de nuevos productos a partir de preferencias declaradas, definir campañas que eviten que los clientes se fuguen de la empresa, etc.

Actualmente existen técnicas que permiten analizar patrones de conducta, nichos de mercado, y muchos otros tipos de información no trivial mediante la utilización de sofisticados modelos que combinan métodos estadísticos, aprendizaje de máquinas y optimización. Estas técnicas se engloban bajo el concepto de minería de datos (*data mining*) [4]. La investigación en estos modelos ha sido un tema relevante en estas últimas dos décadas, habiéndose han logrado avances significativos en términos de eficiencia y desempeño predictivo [15].

La estructura de este trabajo es la siguiente: La sección 2 presenta la derivación del método de clasificación Support Vector Machines. Técnicas recientes de selección de atributos para Support Vector Machines se presentan en la sección 3. La sección 4 describe la metodología propuesta. La sección 5 presenta los principales resultados del trabajo. Finalmente, la sección 6 muestra las conclusiones del trabajo.

2. Support Vector Machines

Dado los ejemplos de entrenamiento $\mathbf{x}_i \in \Re^m$, $i = 1, ..., m$ y un vector de etiquetas binarias $\mathbf{y} \in \Re^m$, $y_i \in \{-1, +1\}$, SVMs construye un hiperplano de la forma $f(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x} + b$ que separa ambas clases, maximizando el *margen*, que es la distancia entre las dos envolturas convexas, medida a lo largo de una línea perpendicular al hiperplano. Esta maximización se consigue minimizando la norma Euclideana de los coeficientes $\mathbf{w}$ [17].

El método SVMs en su versión no lineal proyecta el conjunto de datos a un espacio de mayor dimensión $\mathscr{H}$ utilizando una función $\mathbf{x} \to \phi(\mathbf{x}) \in \mathscr{H}$, donde se construye un hiperplano separador de máximo margen. El siguiente problema de optimización cuadrática debe resolverse:

$$\underset{\mathbf{w}, b, \xi}{\text{Min}} \qquad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \tag{1}$$

sujeto a

$$y_i \cdot (\mathbf{w}^T \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \qquad i = 1, ..., m,$$
$$\xi_i \geq 0 \qquad i = 1, ..., m.$$

Bajo esta proyección los únicos valores que deben calcularse son productos escalares de la forma $\phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ [16]. La proyección es realizada por una función de kernel $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$, que define un producto interno en $\mathscr{H}$. La función de clasificación $f(\mathbf{x})$ dada por SVM corresponde a:

$$f(\mathbf{x}) = signo(\sum_{i=1}^{m} y_i \alpha_i^* K(\mathbf{x}, \mathbf{x}_i) + b^*) \tag{2}$$

La formulación dual puede plantearse de la siguiente manera:

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \tag{3}$$

sujeto a

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq C \qquad i = 1, ..., m.$$

Dentro de las distintas funciones de kernel existentes, las funciones polinomiales y la *radial basis function* (RBF) son más frecuentemente utilizadas en diversas aplicaciones [16]:

1. función polinomial: $K(\mathbf{x}_i, \mathbf{x}_s) = (\mathbf{x}_i \cdot \mathbf{x}_s + 1)^d$, donde $d \in \mathbb{N}$ es el grado del polinomio.
2. *Radial basis function* (RBF): $K(\mathbf{x}_i, \mathbf{x}_s) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_s||^2}{2\rho^2}\right)$, donde $\rho > 0$ es el parámetro que controla el ancho del kernel.

3. Selección de Atributos para SVMs

Para la construcción de modelos de clasificación se desea utilizar la menor cantidad de atributos posibles de manera de obtener un resultado considerado aceptable por el investigador. Sin embargo, el problema radica en la elección y el número de atributos a seleccionar, debido a que esta elección determina la efectividad del modelo de discriminación construido. Este problema se conoce como *selección de atributos* y es combinatorial en el número de atributos originales [1].

De acuerdo a Guyon et al. [5], existen tres estrategias principales para la selección de atributos: los métodos de filtro, los métodos *wrapper* o envolventes, y los métodos *embedded* o empotrados. La primera estrategia utiliza

propiedades estadísticas para "filtrar" aquellos atributos que resulten poco informativos antes de aplicar el algoritmo de aprendizaje, mirando sólo propiedades intrínsecas de los datos. Un método de filtro univariado utilizado comúnmente es el criterio de Fisher (F), el cual calcula la importancia de cada atributo en forma de puntaje al estimar la correlación de cada atributo con respecto a la variable objetivo en un problema de clasificación binaria. El puntaje $F(j)$ para un atributo particular j viene dado por:

$$F(j) = \left| \frac{\mu_j^+ - \mu_j^-}{(\sigma_j^+)^2 + (\sigma_j^-)^2} \right| \tag{4}$$

donde μ_j^+ (μ_j^-) es la media del atributo j para la clase positiva (negativa) y σ_j^+ (σ_j^-) su respectiva desviación estándar.

Los métodos wrapper o envolventes exploran el conjunto completo de atributos para asignarles un puntaje de acuerdo a su poder predictivo en base a la función de clasificación utilizada, lo cual es computacionalmente demandante, sin embargo, puede traer mejores resultados que la utilización de métodos de filtro. Dado que la búsqueda de subconjuntos de atributos crece de forma exponencial con el número de atributos, heurísticas de búsqueda son utilizadas [5]. Estrategias wrapper frecuentemente utilizadas son la Selección Secuencial hacia Adelante (*Sequential forward selection* o SFS) y la Eliminación Secuencial hacia Atrás (*Sequential backward elimination* o SBE) [9]. Para el primer caso, el modelo parte sin considerar variables, para luego probar cada una de ellas e incluir la más relevante en cada iteración. De la misma manera, SBE parte con todas las variables candidatas a formar parte del modelo, eliminando de forma iterativa aquellas variables irrelevantes para la clasificación.

Una estrategia wrapper para selección de atributos utilizando SVMs que surge de manera natural es considerar los coeficientes $\boldsymbol{w}$ asociados a los atributos como medida para la contribución de ellos a la clasificación. Una estrategia SBE podría ser aplicada eliminando de forma iterativa los atributos irrelevantes, es decir, aquellos atributos j con un coeficiente w_j asociado cercano a cero en magnitud (considerando datos normalizados), utilizando la formulación primal de SVMs. La limitación de este método es que la formulación de SVMs no lineal no cuenta con un vector de coeficientes de forma explícita, por lo el método anterior se encuentra limitado a funciones de clasificación lineales. Un popular método wrapper para SVMs basado en la estrategia SBE fue propuesto por Guyon et al. [7] y se conoce como SVM-RFE (SVM- *Recursive Feature Elimination*). El objetivo de este método es encontrar un subconjunto de tamaño r entre las n variables disponibles ($r < n$) que maximice el desempeño de la función de clasificación con SVMs. El atributo que se elimina en cada iteración es aquel cuya extracción minimiza la variación de $W^2(\boldsymbol{\alpha})$, la cual es una medida de la capacidad predictiva del modelo y es inversamente proporcional al margen:

$$W^2(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \tag{5}$$

Ventajas de los métodos wrapper incluyen la interacción entre la búsqueda de subconjuntos de atributos y la selección del modelo, y la capacidad de considerar la dependencia entre atributos. Sus principales desventajas son su alto costo computacional y un mayor riesgo de sobre-ajuste del modelo [5]. Dado que los algoritmos de búsqueda wrapper son por lo general de naturaleza *greedy*, existe un riesgo de quedar estancado en un óptimo local y llegar a un subconjunto de atributos insatisfactorio. Para solucionar este problema, una serie de algoritmos de naturaleza aleatoria en la búsqueda han sido creados [6]. Si bien estos algoritmos permiten encontrar un subconjunto más cercano al óptimo, son más costosos aún en términos computacionales.

El tercer y último enfoque de selección de atributos corresponde a las técnicas empotradas o *embedded*. Estos métodos realizan la búsqueda de un subconjunto óptimo de atributos durante la construcción de la función de clasificación. Al igual que los métodos wrapper, estrategias embedded son específicas para un algoritmo de clasificación.

Existen diferentes estrategias para realizar selección de atributos embedded. Por un lado, la selección de atributos puede ser vista como un problema de optimización. Generalmente, la función objetivo cumple con dos objetivos: maximización de la bondad de ajuste y minimización del número de atributos [5]. Un método que utiliza esta estrategia fue presentado por Bradley y Mangasarian [2] y minimiza una aproximación de la "norma" cero: $\|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|$. Esta formulación no puede considerarse una norma ya que la desigualdad triangular no se cumple [2]. La aproximación utilizada por este método, conocido como FSV (*Feature Selection ConcaVe*), es la siguiente:

$$\|\mathbf{w}\|_0 \approx \mathbf{e}^T(\mathbf{e} - \exp(-\beta|\mathbf{w}|) \tag{6}$$

donde $\beta \in \Re_+$ es un parámetro de aproximación y $\mathbf{e} = (1, ..., 1)^T$. El problema se resuelve finalmente con un algoritmo iterativo.

4. Metodología propuesta

En esta sección se propone un nuevo método de selección de atributos para SVM. La estrategia se basa en una eliminación secuencial hacia atrás y determina la contribución de cada atributo considerando aquel que impacta menos en el desempeño de clasificación en un conjunto de validación independiente. Comenzando con todos los atributos disponibles, cada iteración eliminará aquellos atributos que afectan el desempeño predictivo hasta que se alcance el criterio de parada [11].

4.1. Notación y Aspectos Preliminares

El operador de producto vectorial por componentes $*$ se define como [18]:

$$\mathbf{a} * \mathbf{b} = (a_1 b_1, ..., a_n b_n) \tag{7}$$

El vector $\boldsymbol{\sigma}, \boldsymbol{\sigma} \in \{0,1\}^n$, actúa como un indicador de los atributos que están participando en la construcción de la función de clasificación. La función de Kernel toma la siguiente forma:

$$K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \equiv K(\boldsymbol{\sigma} * \mathbf{x}_i, \boldsymbol{\sigma} * \mathbf{x}_s) \tag{8}$$

El método propuesto utiliza el vector $\boldsymbol{\sigma}$ como parámetro y, para un $\boldsymbol{\sigma}$ dado, se resuelve la formulación dual de SVM:

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \tag{9}$$

sujeto a

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

$$0 \leq \alpha_i \leq C \qquad i = 1, ..., m.$$

4.2. Hold-out Support Vector Machines (HO-SVM)

La idea básica del método propuesto es la de eliminar aquellos atributos cuya eliminación implique un menor número de errores en un conjunto de validación independiente. El método recibe el nombre de Hold-out Support Vector Machines (HO-SVM), ya que en cada iteración el algoritmo genera una nueva partición del conjunto de datos en dos subconjuntos: uno de entrenamiento, donde se construye la función de clasificación, y otro de validación, donde se evalúa el desempeño predictivo de la función construida y se seleccionan los atributos a eliminar. Este paso se conoce en la literatura como *hold-out*. A continuación se presenta el algoritmo iterativo para la eliminación de atributos:

A continuación se detallan los pasos presentados en el algoritmo anterior:

Selección del modelo: El primer paso corresponde a determinar los parámetros de SVM (el parámetro de control del error de clasificación C, el grado del polinomio d o el ancho de un kernel Gaussiano ρ) cuando todos los atributos son seleccionados.

Inicialización: Se define $\boldsymbol{\sigma} = (1, ..., 1)$, es decir, se comienza con todos los atributos disponibles.

Partición de los datos: Se divide el conjunto de datos en dos subconjuntos: entrenamiento, con aproximadamente el 70 % de los ejemplos, y validación, con el 30 % restante.

Algorithm 1 Algoritmo HO-SVM para Selección de Atributos

1. Selección del Modelo
2. Inicialización
3. **repetir**
   - *a*) Partición aleatoria del conjunto de datos (hold-out)
   - *b*) entrenamiento del modelo (ecuación (9))
   - *c*) **para todo** atributo p con $\sigma_p = 1$, calcular $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, el número de errores de clasificación cuando el atributo p es removido.
   - *d*) eliminar atributo j con menor valor de $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$
4. **hasta que** el menor valor de $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ sea mayor que $E(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, el error de validación con todos los atributos seleccionados que cumplen $\boldsymbol{\sigma} = 1$.

Entrenamiento: se entrena un clasificador SVM (ecuación (9)) en el conjunto de entrenamiento con los atributos indicados por el vector $\boldsymbol{\sigma}$.

Calcular $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$: **para todo** atributo p con $\sigma_p = 1$, calcular:

$$E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \sum_{l \in VAL} \left| y_l^v - sgn\left(\sum_{i \in TRAIN} \alpha_i y_i K_{\boldsymbol{\sigma}}(\mathbf{x}_i^{(-p)}, \mathbf{x}_l^{v(-p)}) + b \right) \right| \quad (10)$$

donde VAL es el conjunto de validación y $\mathbf{x}_l^v$ y y_l^v son las observaciones y etiquetas en este conjunto, respectivamente. $\mathbf{x}_i^{(-p)}$ ($\mathbf{x}_l^{v(-p)}$) indica el objeto de entrenamiento i (ejemplo de validación l) con el atributo p removido. $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ es finalmente el número de errores en el conjunto de validación cuando el atributo p es eliminado.

Con el objetivo de reducir la complejidad computacional utilizamos la misma aproximación propuesta por Guyon et al. [7]: el vector $\boldsymbol{\alpha}$ utilizado en (10) se supone igual al de la solución de la formulación (9), incluso cuando se ha removido un atributo.

Criterio para Eliminación de Atributos: Eliminar el atributo j ($\sigma_j = 0$) con el menor valor de $E_{(-j)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$. El atributo j con el menor valor de $E_{(-j)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ es aquel cuya eliminación implica el menor número de errores de validación. En caso de empates en el número de errores se puede seleccionar un atributo al azar o eliminar todos estos atributos para acelerar el algoritmo.

Criterio de Parada: El algoritmo se detiene cuando el menor valor de $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ es mayor o igual a $E(\boldsymbol{\alpha}, \boldsymbol{\sigma})$. De manera alternativa, se puede modificar el criterio para remover más atributos considerando desigualdad estricta.

La figura 1 ilustra el proceso del algoritmo:

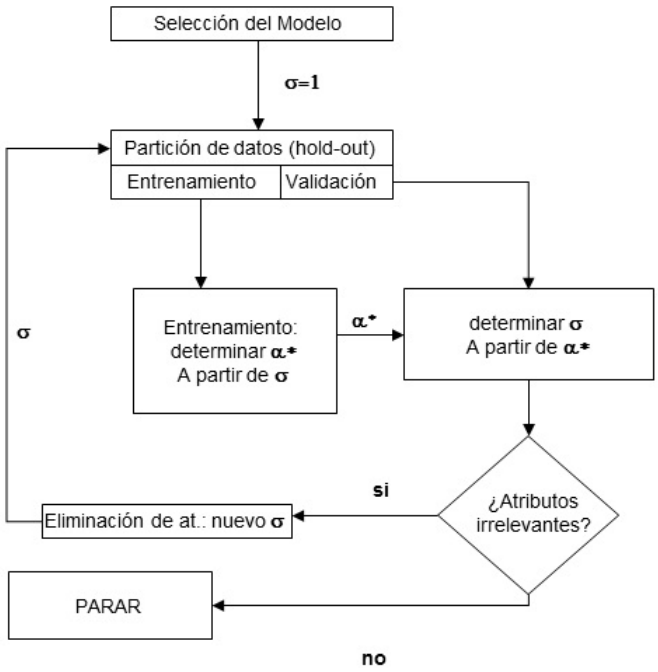


Figura 1: Selección de atributos utilizando HO-SVM

5. Resultados

El enfoque propuesto fue aplicado en cuatro bases de datos de clasificación: dos bases de *benchmark* utilizadas por la comunidad de aprendizaje computacional [13, 14] y dos de proyectos realizados para entidades financieras chilenas. La metodología aplicada consiste en (1) selección del modelo para obtener la mejor configuración de parámetros (2) ranking de variables y (3) medir el desempeño en un conjunto de test para un número creciente de atributos rankeados. Se obtiene un error promedio de 100 realizaciones de los métodos [13, 14]. Para este procedimiento se utilizó Spider Toolbox para Matlab [19]. A continuación se describen las bases de datos utilizadas.

5.1. Descripción de las bases de datos

Wisconsin Breast Cancer (WBC): Esta base de datos del UCI *data repository* [8] contiene 569 observaciones (212 tumores malignos y 357 benignos) descritos por 30 atributos. La base de datos no contiene valores perdidos y sus atributos fueron escalados entre cero y uno.

Colorectal Microarray (CRMA): La base de datos contiene la expresión de 2000 genes para 62 muestras de tejido (40 con tumor y 22 normales).

INDAP: La tercera base de datos proviene de un proyecto realizado para la organización chilena INDAP y se basa en una muestra balanceada de

49 variables descritas por 1,464 observaciones (767 clientes "buenos" y and 697 clientes "malos") [3]. INDAP es el servicio más importante provisto por el gobierno que apoya financieramente a pequeños agricultores. Fue fundado en 1962 y cuenta con más de 100 oficinas a lo largo de Chile sirviendo a sus más de 100,000 clientes.

BDDM: Un sistema de asignación de créditos fue desarrollado para la división de micro-empresas del banco chileno Banco del Desarrollo, el cual pertenece al grupo Scotiabank. Esta división se especializa en créditos para micro-empresarios y tuvo una participación de mercado de un 30 % el 2007. La base contiene una muestra balanceada de los créditos aceptados entre los años 2004 y 2006. Para cada uno de los 3,003 créditos disponibles se tomó una decisión para clasificar el comportamiento del cliente entre "buenos" y "malos" considerando 24 atributos pre-seleccionados mediante métodos univariados.

5.2. Resultados

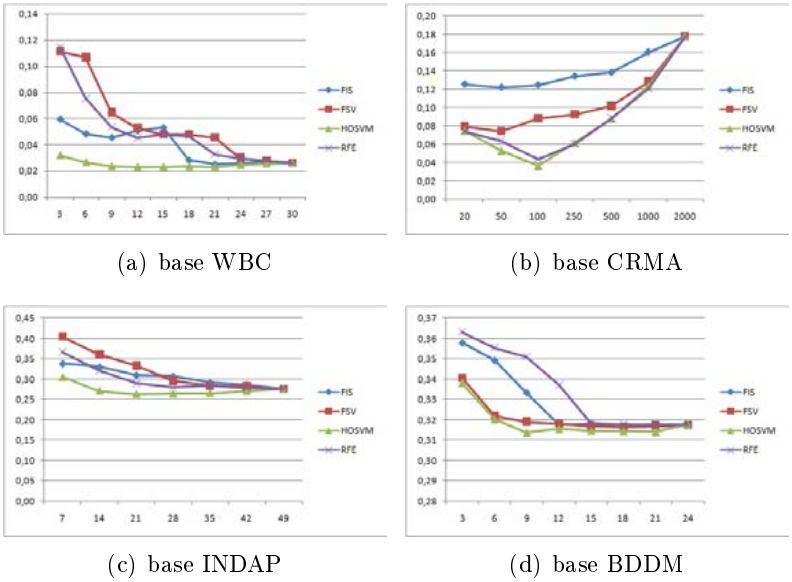
Primero se comparan los resultados para los mejores modelos obtenidos para diferentes funciones de Kernel. La Tabla 1 presenta la media y desviación estándar del desempeño (tasa de acierto) de testeo utilizando validación cruzada para los parámetros:
$C = \{0{,}1, 0{,}5, 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 1000\}$;
$d = \{2, 3, 4, 5, 6, 7, 8, 9\}$ and $\rho = \{0{,}1, 0{,}5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 100\}$.
En esta etapa se demuestra para nuestros datos que la mejor alternativa es el Kernel Gaussiano o RBF.

	SVM lineal	SVM polinomial	SVM RBF
WBC	94.55±2.4	96.49±2.2	98.25±2.0
CRMA	80.30±6.4	80.30±6.4	85.70±5.6
INDAP	71.10±4	75.27±3.3	75.54±3.6
BDDM	68.70±0.7	69.26±1.0	69.33±1.0

Cuadro 1: Desempeño para las cuatro bases de datos considerando diferentes funciones de Kernel.

Como segunda etapa se compara el desempeño de clasificación para diferentes estrategias de selección de atributos presentados en este trabajo (Fisher, RFE-SVM, FSV y nuestro enfoque HO-SVM). Las figuras 2(a), 2(b), 2(c), y 2(d) representan el error promedio para un número creciente de atributos rankeados. Las figuras muestran que HO-SVM consigue un desempeño consistentemente superior en las cuatro bases de datos estudiadas.

Para enfatizar la importancia del criterio de parada del método HO-SVM, se estudia el desempeño de cada algoritmo de selección de atributos para un número fijo de atributos, obtenido cuando el método HO-SVM alcanza el criterio de parada.



(a) base WBC (b) base CRMA

(c) base INDAP (d) base BDDM

Figura 2: Error promedio vs. número de atributos seleccionados para las cuatro bases de datos estudiadas.

	n	Fisher+SVM	FSV	RFE-SVM	HO-SVM
WBC	12	94.91±1.2	94.70±1.3	95.47±1.1	**97.69±0.9**
CRMA	100	87.55±7.5	91.17±6.7	95.61±5.4	**96.36±5.3**
INDAP	21	69.02±1.5	66.70±1.7	71.07±1.8	**73.65±1.5**
BDDM	9	66.66±1.2	68.09±1.0	64.89±1.2	**68.63±1.0**

Cuadro 2: Número de atributos seleccionados, media y desviación de la efectividad para cuatro bases de datos.

De la Tabla 2 se concluye que el método HO-SVM consigue un desempeño significativamente mejor en todas las bases. El segundo mejor método es RFE-SVM, pero éste obtiene un mal desempeño para la base BDDM.

6. Conclusiones

El trabajo presenta un nuevo método iterativo de selección de atributos para SVM. Este método realiza una eliminación secuencial hacia atrás, utilizando el número de errores en un conjunto independiente como criterio para eliminar atributos en cada iteración. Una comparación con otras técnicas muestra las ventajas de nuestro enfoque:

- Consigue un mejor desempeño predictivo que otras estrategias de filtro y wrapper, debido a su habilidad para ajustarse mejor a los datos,

gracias a la medida de desempeño en validación, pero evitando caer en sobreajuste.

- Presenta un criterio de parada explícito, indicando claramente cuando la eliminación de atributos comienza a afectar negativamente el desempeño del método.
- Se puede utilizar con cualquier función de Kernel.
- se puede extender de forma simple a variaciones de SVM, como SVM multiclase, y a otros métodos de clasificación.

El algoritmo se basa en una estrategia de búsqueda iterativa, lo cual es computacionalmente costoso si el número de atributos es muy alto. Para mejorar el desempeño de este tipo de métodos es recomendable aplicar métodos de filtro de forma previa al algoritmo iterativo [10]. De esta forma es posible identificar de forma rápida atributos claramente irrelevantes de forma menos costosa. En nuestros proyectos de asignación de créditos utilizamos test Chi-cuadrado para variables categóricas y Kolmogorov-Smirnov para variables continuas con muy buenos resultados [12].

Como trabajo futuro se proponen las siguientes directrices. Primero, resulta interesante la adaptación del método para variaciones de SVM y otros métodos de clasificación. Segundo, el método puede ser útil para seleccionar atributos relevantes en problemas de bases desbalanceadas mediante una adaptación de la función de error considerando los costos de equivocarse (Error Tipo I y Tipo II). Este tipo de problemas es frecuente en aplicaciones de análisis de negocios, tales como riesgo financiero, detección de fraude y predicción de fuga de clientes.

Referencias

[1] Blum, A., P. Langley, P. (1997): Selection of relevant features and examples in machine learning. Artificial Intelligence 97, 245-271.

[2] Bradley, P., Mangasarian, O. (1998): Feature selection vía concave minimization and support vector machines. Machine Learning proceedings of the fifteenth International Conference (ICML'98) 82 -90, San Francisco, California, Morgan Kaufmann.

[3] Coloma, P., Guajardo, J., Miranda, J., Weber, R. (2006): Modelos analíticos para el manejo del riesgo de crédito. Trend Management 8, 44-51.

[4] Fayyad, U., Piatetsky-shapiro, G., Smyth, P. (1996): From Data Mining to Knowledge Discovery in Databases. AI Magazine 17, 37-54.

[5] Guyon, I., Elisseeff, A.(2003): An Introduction to Variable and Feature Selection. Journal of Machine Learning research 3, 1157-1182.

[6] Guyon, I., Gunn, S., Nikravesh, M. , Zadeh, L. A. (2006): Feature extraction, foundations and applications. Springer, Berlin.

[7] Guyon, I., Weston, J., Barnhill, S. ,Vapnik, V. (2002): Gene selection for cancer classification using support vector machines, Machine Learning 46 (1-3), 389-422.

[8] Hettich, S., Bay, S. D.(1999): The UCI KDD Archive `http://kdd.ics.uci.edu`. Irvine, CA: University of California, Department of Information and Computer Science.

[9] Kittler, J. (1978): Pattern Recognition and Signal Processing, Chapter Feature Set Search Algorithms Sijthoff and Noordhoff, Alphen aan den Rijn, Netherlands, 41-60.

[10] Liu, Y., Zheng, Y. F. (2006). FS-SFS: A novel feature selection method for support vector machines. Pattern Recognition 39, 1333-1345.

[11] Maldonado, S., Weber, R. (2009): A wrapper method for feature selection using Support Vector Machines. Information Sciences 179 (13), 2208-2217.

[12] Maldonado, S., Weber, R. (2010): Feature Selection for Support Vector Regression via Kernel Penalization. Proceedings of the 2010 International Joint Conference on Neural Networks, Barcelona, Spain, 1973-1979.

[13] Rakotomamonjy, A. (2003): Variable Selection Using SVM-based Criteria. Journal of Machine Learning research 3, 1357-1370.

[14] Rätsch, G., Onoda, T., and Müller, K-R (2001). Soft margins for AdaBoost. Machine Learning 42(3), 287-320.

[15] Ruiz Sánchez, R. (2006): Heurísticas de selección de atributos para datos de gran dimensionalidad. Tesis Doctoral, Sevilla, Universidad de Sevilla. Mimeografiada.

[16] Schölkopf, B. and Smola, A. J.(2002). Learning with Kernels. Cambridge, MA, USA: MIT Press.

[17] Vapnik, V. (1998): Statistical Learning Theory. John Wiley and Sons, New York.

[18] Weston, J., Mukherjee, S., Chapelle, O., Ponntil, M., Poggio, T., Vapnik, V.(2001): Feature selection for SVMs, Advances in Neural Information Processing Systems 13, MIT Press, Cambridge, MA.

[19] Weston, J., Elisseeff, A., Bakir, G., Sinz, F.: The spider. `http://www.kyb.tuebingen.mpg.de/bs/people/spider/`.