Modelos de Selección de Atributos para Support Vector Machines

Sebastián Maldonado*
Richard Weber**

Resumen

Recientemente los volúmenes de 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. Esta gran cantidad de datos causa serios problemas a muchos algoritmos de minería de datos en términos de escalabilidad y rendimiento. Dentro de las áreas de investigación en selección de atributos se incluyen el análisis de chips de ADN, procesamiento de documentos provenientes de internet y modelos de administración de riesgo en el sector financiero. El objetivo de la selección de atributos es triple: mejorar el desempeño predictivo de los modelos, implementar algoritmos más rápidos y menos costosos, y proveer de un mejor entendimiento del proceso subyacente que generó los datos. Dentro de las técnicas de minería de datos, el método 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 para clasificación o regresión. En muchas aplicaciones, estas características permiten obtener mejores resultados que otros métodos predictivos. Sin embargo, una limitación de este método es que no está diseñado para identificar los atributos importantes para construir la regla discriminante. El presente trabajo tiene como objetivo desarrollar técnicas que permitan incorporar la selección de atributos en la formulación no lineal de SVMs, aportando eficiencia y comprensibilidad al método.

Palabras Clave: Selección de Atributos, Support Vector Machines, Clasificación Supervisada

*Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Santiago, Chile.
**Departamento de Ingeniería Industrial, Facultad de Ciencias Físicas y Matemáticas. Universidad de Chile, Santiago, Chile.

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, diagnóstico temprano de tumores cancerígenos mediante el análisis de chips de ADN, etc.

Si bien obtener información potencialmente útil es cada vez más simple, gracias al importante aumento de la capacidad de almacenaje y la disponibilidad de mejores herramientas para el manejo de datos, el proceso de extracción de información relevante a partir de los datos disponibles sigue siendo complejo y costoso. Dada esta mayor disponibilidad de información, el proceso de generación de conocimiento a partir de los datos cobra además una mayor importancia.

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*) [7]. La investigación en estos modelos ha sido un tema relevante en estas últimas dos décadas, habiéndose logrado avances significativos en términos de eficiencia y desempeño predictivo. En esta misma línea, el presente trabajo busca el desarrollo de algoritmos que combinen la clasificación mediante SVMs y la selección de atributos, robusteciendo los modelos en términos predictivos y generando conocimiento mediante la interpretación de la solución obtenida.

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

En esta sección se describe la derivación matemática de SVMs como técnica de clasificación binaria. Se comenzará con la descripción del enfoque para funciones de discriminación lineales, para luego extender el método a funciones no lineales.

2.1. Clasificación Lineal para Problemas Linealmente Separables

Para el caso linealmente separable, SVMs determina el hiperplano óptimo que separa el conjunto de datos. Para este propósito, "linealmente separable" requiere encontrar el par $(\mathbf{w}, b)$ tal que clasifique correctamente los vectores de ejemplos $\mathbf{x}_i$ en dos clases y_i, es decir, para un espacio de hipótesis dado por un conjunto de funciones $f_{\mathbf{w},b} = signo(\mathbf{w}^T \cdot \mathbf{x}_i + b)$ se impone la siguiente restricción:

$$\begin{array}{c} \text{Min} \\ i = 1, ..., m \end{array} \qquad |\mathbf{w}^T \cdot \mathbf{x}_i + b| = 1 \tag{1}$$

Los hiperplanos que satisfacen (1) se conocen como *hiperplanos canónicos* [18]. El objetivo de SVM es encontrar, entre todos los hiperplanos canónicos que clasifican correctamente los datos, aquel con menor norma, o, equivalentemente, con mínimo $||\mathbf{w}||^2$. Es interesante notar que la minimización de $||\mathbf{w}||^2$ es equivalente a encontrar el hiperplano separador para el cual la distancia entre dos envolturas convexas (las dos clases del conjunto de datos de entrenamiento, asumiendo que son linealmente separables), medida a lo largo de una línea perpendicular al hiperplano, es maximizada. Esta distancia se conoce como *margen* [25]. La figura 1 ilustra la construcción del margen de separación con dos atributos predictivos:

El problema de maximización del margen se formula de la siguiente manera:

$$\begin{array}{c} \text{Min} \\ \mathbf{w}, b \end{array} \qquad \frac{1}{2} \|\mathbf{w}\|^2 \tag{2}$$

sujeto a

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

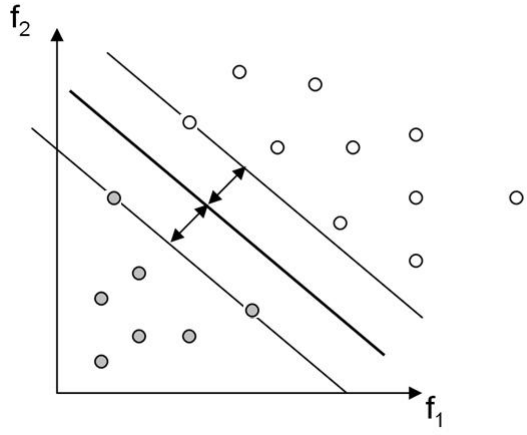


Figura 1: Hiperplano Óptimo de SVM para un Problema Bidimensional

A partir de esta formulación se construye el dual mediante la técnica de los multiplicadores de Lagrange. La formulación dual permitirá construir funciones de clasificación no lineales, lo que usualmente lleva a un mayor poder predictivo, como se presentará en la sección 2.3. La formulación dual de (2) corresponde a:

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

sujeto a

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

$$\alpha_i \geq 0 \qquad\qquad i = 1, ..., m.$$

donde α_i representan los multiplicadores de Lagrange asociados a las restricciones de (2). Los multiplicadores que cumplen con $\alpha_i > 0$ son llamados *Support Vectors*, ya que son los únicos que participan en la construcción del hiperplano de clasificación. Se tiene además que $\mathbf{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \mathbf{x}_i$ y $b^* = y_i - \mathbf{w}^* \cdot \mathbf{x}_i$ para cada Support Vector $\mathbf{x}_i$. La función de decisión puede escribirse como:

$$f(\mathbf{x}) = signo(\mathbf{w}^* \cdot \mathbf{x} + b^*) = signo(\sum_{i=1}^{m} y_i \alpha_i^* (\mathbf{x} \cdot \mathbf{x}_i) + b^*) \tag{4}$$

2.2. Clasificación Lineal para Problemas Linealmente no Separables

Ahora se considera el caso en que no existe un hiperplano separador, es decir, no es posible satisfacer todas las restricciones del problema (2).

Con el fin de considerar un costo por observación mal clasificada, se introduce un conjunto adicional de variables ξ_i, $i = 1, ..., m$. SVMs resuelve el siguiente problema de optimización:

$$\begin{array}{ll} \underset{\mathbf{w}, b, \boldsymbol{\xi}}{\text{Min}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \end{array} \tag{5}$$

sujeto a

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

$$\xi_i \geq 0 \qquad i = 1, ..., m.$$

La función de clasificación se mantiene: $f(\mathbf{x}) = signo(\sum_{i=1}^{m} y_i \alpha_i^* (\mathbf{x} \cdot \mathbf{x}_i) + b^*)$, donde $b^* = y_i - \mathbf{w}^* \cdot \mathbf{x}_i$ para cada *Support Vector* $\mathbf{x}_i$ tal que $0 < \alpha_i < C$.

2.3. Clasificación no Lineal

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

$$\begin{array}{ll} \underset{\mathbf{w}, b, \xi}{\text{Min}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \end{array} \tag{6}$$

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 la solución obtenida aplicando SVM toma la siguiente forma:

$$f(\mathbf{x}) = signo(\mathbf{w}^* \cdot \phi(\mathbf{x}) + b^*) = signo(\sum_{i=1}^{m} y_i \alpha_i^* \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i) + b^*) \tag{7}$$

Notar que los únicos valores que deben calcularse son productos escalares de la forma $\phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ [21]. 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:

$$f(\mathbf{x}) = signo(\sum_{i=1}^{m} y_i \alpha_i^* K(\mathbf{x}, \mathbf{x}_i) + b^*) \tag{8}$$

La formulación dual puede reformularse de la siguiente manera:

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

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, el kernel lineal (equivalente a la formulación dual de la formulación (5)), las funciones polinomiales y la *radial basis function* (RBF) o Kernel Gaussiano son más frecuentemente utilizadas en diversas aplicaciones [22]:

1. kernel lineal: $K(\mathbf{x}_i, \mathbf{x}_s) = \mathbf{x}_i^T \cdot \mathbf{x}_s$
2. 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.
3. *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.

La selección del mejor kernel para una aplicación es todavía un tema de investigación [1]. El procedimiento más común es el de seleccionar los parámetros de kernel (el grado del polinomio d para las funciones polinomiales o el ancho del kernel ρ para la función Gaussiana) calibrando estos parámetros en conjunto con el proceso de selección del modelo (parámetro C que controla la generalización del modelo) mediante una búsqueda de grilla [13]. En el presente trabajo se utiliza validación cruzada de 10 iteraciones para encontrar la mejor combinación de parámetros. Existen además diversos enfoques que buscan una selección más eficiente utilizando medidas estadísticas clásicas a partir de la distribución de los datos [1].

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 [2].

Una desventaja del método SVMs es que no está diseñado para identificar los atributos importantes para construir la regla discriminante [16]. La utilización de la norma euclidiana en la formulación primal de SVMs (5) para el cálculo del margen en la función objetivo no busca anular componentes del vector $\boldsymbol{w}$. Por ejemplo, sean los vectores $\boldsymbol{w}_1 = (0,5; 0,5; 0,5; 0,5)$ y $\boldsymbol{w}_2 = (1; 0; 0; 0)$; ambos poseen la misma norma euclidiana ($||\boldsymbol{w}_1||^2 = ||\boldsymbol{w}_2||^2 = 1$), y por ende ambas soluciones tienen el mismo valor en el problema de minimización que formula SVMs. Sin embargo, el primer caso plantea una solución con cuatro atributos, mientras que el segundo caso utiliza sólo un atributo, siendo los tres restantes irrelevantes para la clasificación. Dado que SVMs no distingue entre ambas soluciones, su diseño podría considerarse no adecuado para lograr una clasificación efectiva y a la vez eficaz en identificar los atributos que no contribuyen a ésta.

De acuerdo a Guyon et al. [8], 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. En muchos casos un puntaje o *score* de relevancia es calculado para cada atributo, eliminando aquellos con bajo puntaje. Esta estrategia es independiente del algoritmo predictivo, lo que implica ventajas y desventajas:

- Son computacionalmente simples y rápidos de ejecutar.
- Son fácilmente escalables a bases de datos de alta dimensionalidad, ya que la selección de atributos sólo necesita ser aplicada una vez, para luego evaluar el desempeño de diferentes métodos de clasificación.
- Estos métodos ignoran las interacciones con el método predictivo, y, por ende, las relaciones entre las distintas variables.

El último punto es particularmente relevante ya que ignorar las interacciones entre las variables puede afectar negativamente el desempeño de clasificación. Atributos presumiblemente redundantes de acuerdo a medidas informativas pero correlacionados entre sí pueden aportar a la clasificación de forma significativa. Los siguientes dos ejemplos ilustran este efecto [8]: La figura 2.a muestra la distribución condicional de dos variables con matrices de covarianza idénticas y direcciones principales diagonales. Se observa que una de las variables (arriba, izquierda en la figura 2.a) presenta su distribución condicional completamente traslapada con respecto a la variable objetivo (distinción entre barras negras y blancas), mientras la segunda (abajo, derecha) presenta un poder discriminante importante, sin embargo no alcanza una separación perfecta por sí sola. La utilización de ambas variables en conjunto permite lograr una clasificación perfecta en este caso (arriba, derecha y abajo, izquierda), mejorando significativamente el desempeño de clasificación.

Un ejemplo aún más ilustrativo se presenta en la figura 2.b: en este caso se tienen ejemplos de dos clases utilizando cuatro distribuciones normales en las coordenadas (0;0), (0;1), (1;0), and (1;1). Las etiquetas para estos cuatro grupos se distribuyen de acuerdo a la función lógica XOR: f(0;0)=1, f(0;1)=-1, f(1;0)=-1; f(1;1)=1. Notar que las proyecciones en los ejes no entregan separación entre clases (diagonales en Fig. 2.b), sin embargo, ambas variables en conjunto permiten obtener una clasificación perfecta con algún clasificador no lineal sencillo.

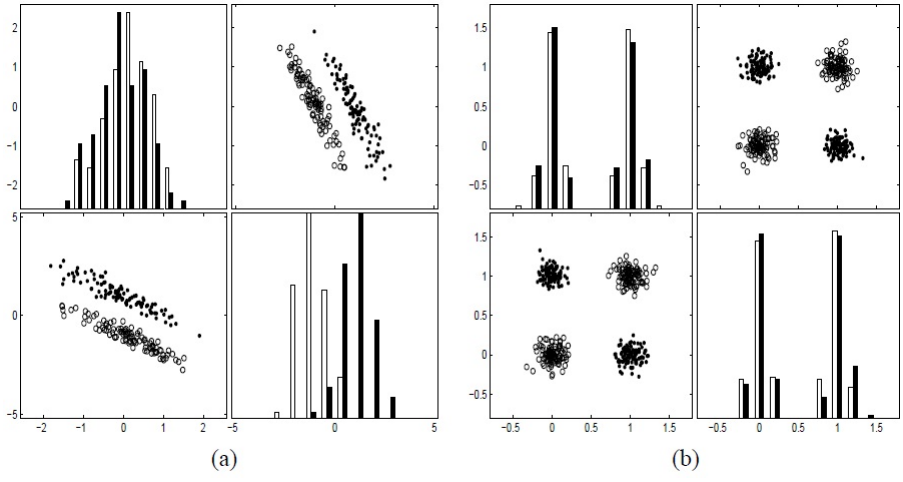


Figura 2: Variables Irrelevantes Por Sí Solas Pero Relevantes Junto con Otras

Una serie de métodos de filtro multivariados han sido introducidos para estudiar la interacción entre variables. Estas metodologías, sin embargo, suelen ser menos rápidas y escalables que los métodos de filtro univariados [9].

Un método de filtro univariado utilizado comúnmente es el criterio de Fisher (F), el cual calcula la importancia de cada atributo en forma de score 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:

$$F(j) = \left| \frac{\mu_j^+ - \mu_j^-}{(\sigma_j^+)^2 + (\sigma_j^-)^2} \right| \tag{10}$$

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. Otras medidas de filtro son el estadístico χ^2, el cual mide la independencia entre la distribución de los ejemplos y clases; y la Ganancia de la Información (*Information Gain*), medida comúnmente utilizada para la construcción de árboles de decisión como método de clasificación, que mide la entropía o "desorden" en el sistema de acuerdo a la Teoría de la Información [24].

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 [8]. 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) [14]. 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 $\mathbf{w}$ 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 (5). 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 que 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. [11] 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:

$$W^2(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \tag{11}$$

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 [8]. 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 [9]. 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 [8]. Un método que utiliza esta estrategia fue presentado por Bradley y Mangasarian [3] 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 [3]. La aproximación utilizada por este método, conocido como FSV (*Feature Selection ConcaVe*), es la siguiente:

$$\|\mathbf{w}\|_0 \approx \mathbf{e}^T(\mathbf{e} - \exp(-\beta|\mathbf{w}|) \tag{12}$$

donde $\beta \in \Re_+$ es un parámetro de aproximación y $\mathbf{e} = (1, ..., 1)^T$. El problema se resuelve finalmente con un algoritmo iterativo. Weston et al. [27] demuestra

que la minimización de la norma cero para SVM (l_0-SVM) puede aproximarse con una modificación simple del algoritmo *vanilla* SVM:

Algorithm 1

1. Entrenar una SVM lineal de acuerdo a (5).
2. Re-escalar las variables multiplicándolas por el valor absoluto de los componentes del vector de pesos $\mathbf{w}$.
3. Iterar los primeros dos pasos hasta convergencia.

end

Weston argumenta que, en la práctica, esta estrategia permite una mejor generalización que la minimización de la norma cero [27]. Una limitación de estas estrategias iterativas basadas en el vector de coeficientes es que se encuentran limitadas a funciones de clasificación lineales [9, 16].

Existen varios enfoques propuestos que utilizan estrategias de selección de atributos que se basan en estrategias de selección hacia adelante o hacia atrás para identificar los atributos relevantes, y de esta manera construir un *ranking*, el cual puede utilizarse a modo de filtro antes de aplicar SVM. Uno de estos métodos es el ya presentado SVM-RFE. Otro método de esta naturaleza que permite la utilización de funciones de kernel son los presentados en Rakotomamonjy [19], que utiliza una cota para el error de clasificación *leave-one-out* (LOO) de SVM, el *radius margin bound* [25]:

$$LOO \leq 4R^2||\mathbf{w}||^2 \tag{13}$$

donde R denota el radio de la menor esfera inscrita que contiene los datos de entrenamiento. Esta cota también es utilizada en Weston et al. [28] mediante la estrategia conocida como la optimización de factores de escalamiento (*scaling factors*). La selección de atributos con *scaling factors* se realiza mediante el escalamiento de las variables de entrada por un vector $\boldsymbol{\sigma} \in [0,1]^n$. Valores grandes de σ_j indican una mayor relevancia. El problema consiste en encontrar el mejor kernel de la siguiente forma:

$$K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \equiv K(\boldsymbol{\sigma} * \mathbf{x}_i, \boldsymbol{\sigma} * \mathbf{x}_s) \tag{14}$$

donde $*$ es el operador para el producto vectorial por componentes. El método presentado por Weston et al. utiliza un algoritmo para actualizar $\boldsymbol{\sigma}$ mediante

el Método del Gradiente. Enfoques que utilizan otras cotas para el mismo propósito se presentan en Chapelle et al. [5]. Canu y Grandvalet [4] proponen reducir la utilización de atributos restringiendo los factores de escalamiento en la formulación de SVM mediante un parámetro σ_0 que controla la norma de $\boldsymbol{\sigma}$:

$$\underset{\boldsymbol{\sigma}}{\text{Min}} \quad \underset{\boldsymbol{\alpha}}{\text{Max}} \qquad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \tag{15}$$

sujeto a

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

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

$$||\boldsymbol{\sigma}||_p = \sigma_0,$$

con $K_{\boldsymbol{\sigma}}$ definido en (14). Mientras más cercano a cero sea el parámetro p, más estricta será la selección de atributos, sin embargo, la optimización será también más compleja [9].

A modo general, los métodos *embedded* presentan importantes ventajas como la interacción entre las variables y el modelo de clasificación (en este caso SVMs), la modelación de las dependencias entre variables y ser computacionalmente menos costosos que los métodos *wrapper* [8]. Sin embargo, estos métodos tienden a ser conceptualmente más complejos, y muchas veces las modificaciones impuestas alteran la naturaleza convexa del problema planteado por SVMs, requiriendo algoritmos no lineales que pueden caer en óptimos locales. Adicionalmente, muchos métodos empotrados se encuentran restringidos sólo para SVMs lineal, limitando el potencial predictivo que otorgan las funciones de kernel.

Este estudio del estado del arte de la selección de atributos para SVM proporciona una guía general en los diversos aspectos que comprende esta tarea. Además de definir el concepto de selección y de analizar su proceso, se ha clasificado y descrito una gran cantidad de algoritmos existentes. Si bien la investigación en el área de selección de atributos para SVMs tuvo su *peak* en el año 2003, cuyos trabajos se resumen en el libro de Guyon et al. [9], el importante crecimiento de la capacidad de almacenaje, sumado a nuevas aplicaciones de alta dimensionalidad en el mundo de las ciencias de la vida (tales como el estudio del genoma humano) justifican la investigación en esta área. Las últimas publicaciones del estado del arte consideran algoritmos híbridos, que combinan ventajas de distintos modelos de algoritmos (filtros, wrappers,

ranking, etc) [23]. Otros trabajos apuntan a abordar el problema de selección de atributos desde el punto de vista de la selección del modelo y no en forma de ranking, independiente de la construcción del modelo predictivo final [10]. El enfoque del presente trabajo es precisamente éste, desarrollar modelos que lleguen a una solución final de clasificación en conjunto con la determinación de los atributos relevantes para el modelo, identificando cuándo la eliminación de atributos comienza a afectar el desempeño de los modelos en el entrenamiento del mismo. Esto trae consigo dos ventajas: primero, es posible establecer un criterio de parada para los métodos, identificando claramente cuando la eliminación de atributos comienza a afectar negativamente el desempeño de los modelos. Segundo, reduce el esfuerzo computacional de construir un modelo final a partir de un ranking de atributos, debiendo posteriormente realizar la selección del modelo mediante algún tipo de evaluación (comúnmente validación cruzada), lo cual es computacionalmente demandante y se corre el riesgo de caer en sobreajuste [10]. Guyon [10] plantea que la unificación del proceso de selección de atributos y selección del modelo es uno de los tópicos relevantes para la investigación en aprendizaje computacional hoy en día.

4. Metodología Propuesta

En esta sección se propone un método de selección de atributos para SVMs presentado en Maldonado y Weber [16]. 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.

4.1. Notación y Aspectos Preliminares

El operador de producto vectorial por componentes $*$ se define como [28]:

$$\mathbf{a} * \mathbf{b} = (a_1 b_1, ..., a_n b_n) \tag{16}$$

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:

$$K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \equiv K(\boldsymbol{\sigma} * \mathbf{x}_i, \boldsymbol{\sigma} * \mathbf{x}_s) \tag{17}$$

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

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \tag{18}$$

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:

Algorithm 2

1.Selección del Modelo

2.Inicialización

3.**repetir**

a)Partición aleatoria del conjunto de datos (hold-out)

b)entrenamiento del modelo (ecuación (18))

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 $\boldsymbol{\sigma} = 1$.

end

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.

Entrenamiento: se entrena un clasificador SVM (ecuación (18)) 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:

$$E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \sum_{l \in VAL} \left| y_l^v - sgn\left(\sum_{i \in TRAIN} \alpha_i y_i K_{\boldsymbol{\sigma}}(\mathbf{x}_i^{(-p)}, \mathbf{x}_l^{v(-p)}) + b \right) \right| \quad (19)$$

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. [11]: el vector $\boldsymbol{\alpha}$ utilizado en (19) se supone igual al de la solución de la formulación (18), 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 3 ilustra el proceso del algoritmo:

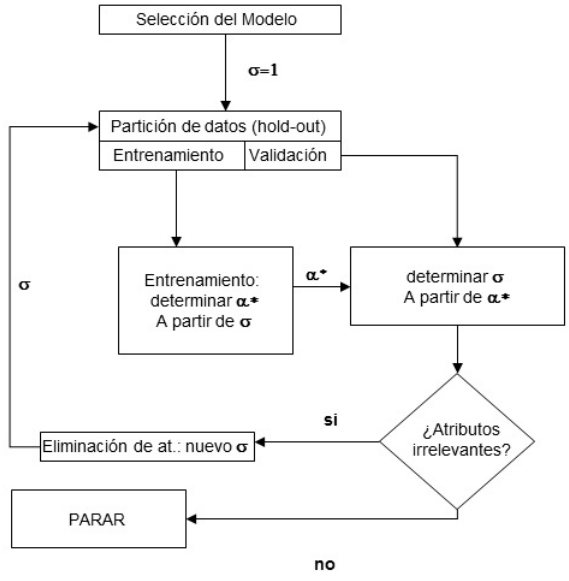


Figura 3: Selección de Atributos Utilizando HO-SVM

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 [19, 20] 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 [19, 20]. Para este procedimiento se utilizó Spider Toolbox para Matlab [26]. 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* [12] 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): Esta 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") [6]. 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.

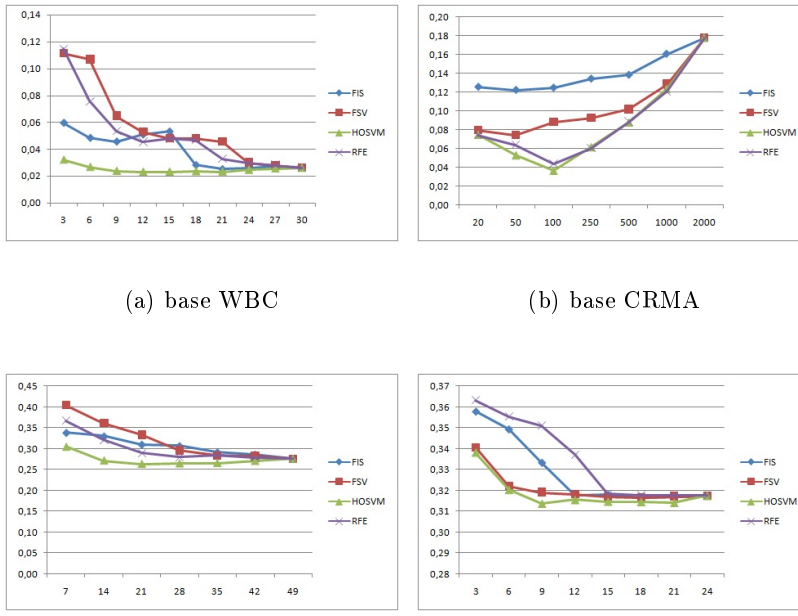
5.2. Resultados

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

Tabla 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 4(a), 4(b), 4(c) y 4(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.



(a) base WBC (b) base CRMA

(c) base INDAP (d) base BDDM

Figura 4: Error promedio vs. número de atributos seleccionados para 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.

	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**

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

Se puede concluir de la Tabla 2 que el método propuesto consigue un desempeño significativamente mejor en todas las bases de datos. 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 [15]. 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 [17].

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.

Agradecimientos: Este trabajo fue parcialmente financiado por el Instituto Sistemas Complejos de Ingeniería (ICM: P-05-004-F, CONICYT: FBO16) (www.sistemasdeingenieria.cl). El primer autor también agradece el financiamiento por parte de CONICYT para su estudio en el Doctorado en Sistemas de Ingeniería de la Universidad de Chile.

Referencias

[1] S. Ali and K. A Smith-Miles. A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing*, 70(1–3):173–186, 2006.

[2] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:254–271, 1997.

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

[4] S. Canu and Y. Grandvalet. Adaptive scaling for feature selection in svms. advances in neural information processing systems. *Cambridge, MA, USA, MIT Press*, 15:553–560., 2002.

[5] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

[6] P. Coloma, J. Guajardo, J. Miranda, and R. Weber. Modelos analíticos para el manejo del riesgo de crédito. *Trend Management*, 8:44–51., 2006.

[7] U. Fayyad, G. Piatetsky-shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54., 1996.

[8] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning research*, 3:1157–1182, 2003.

[9] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature extraction, foundations and applications.* Springer, Berlin., 2006.

[10] I. Guyon, A. Saffari, G. Dror, and G. Cawley. Model selection: Beyond the bayesian frequentist divide. *Journal of Machine Learning research*, 11:61–87, 2009.

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

[12] S. Hettich and S. D. Bay. The uci kdd archive irvine, ca *http:\\kdd.ics.uci.edu. University of California, Department of Information and Computer science.*

[13] C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification., 2003.

[14] J. Kittler. Pattern recognition and signal processing. *Chapter Feature Set Search Algorithms Sijthoff and Noordhoff, Alphen aan den Rijn, Netherlands*, pages 41–60, 1978.

[15] Y. Liu and Y. F. Zheng. Fs-sfs: A novel feature selection method for support vector machines. *Pattern Recognition*, 39:1333–1345, 2006.

[16] S. Maldonado and R. Weber. A wrapper method for feature selection using support vector machines. *Information Sciences*, 179(13):2208–2217, 2009.

[17] S. Maldonado and R. Weber. Feature selection for support vector regression via kernel penalization. *Proceedings of the 2010 International Joint Conference on Neural Networks, Barcelona, Spain*, pages 1973–1979, 2010.

[18] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. *MIT Artificial Intelligence Laboratory , A. I. Memo AIM-1602.*, 1997.

[19] A. Rakotomamonjy. Variable selection using svm-based criteria. *Journal of Machine Learning research*, 3:1357–1370, 2003.

[20] G. Rätsch, T. Onoda, and K-R Müller. Soft margins for adaboost. *Machine Learning*, 42(3):287–320, 2001.

[21] B. Schölkopf and A. J. Smola. Learning with kernels. *Cambridge, MA, USA: MIT Press.*, 2002.

[22] J. Shawe-Taylor and N. Cristianini. Kernel methods for pattern analysis. *Cambridge University Press, Cambridge.*, 2004.

[23] ö. Uncu and I.B. Türksen. A novel feature selection approach: Combining feature wrappers and filters. *Information Sciences*, 177:449–466., 2007.

[24] J. Van Hulse, T.M. Khoshgoftaar, A. Napolitano, and R. Wald. Feature selection with high-dimensional imbalanced data. *Proceedings of the 2009 IEEE International Conference ICDMW '09*, pages 507–514, 2009.

[25] V. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, New York., 1998.

[26] J. Weston, A. Elisseeff, G. Bakir, and F. Sinz. The spider. *http:\\www.kyb.tuebingen.mpg.de\debspeople \spider*.

[27] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. The use of zero-norm with linear models and kernel methods. *Journal of Machine Learning research*, 3:1439–1461, 2003.

[28] J. Weston, S. Mukherjee, O. Chapelle, M. Ponntil, T. Poggio, and V. Vapnik. Feature selection for svms. *Advances in Neural Information Processing Systems , MIT Press, Cambridge, MA.*, 13, 2001.