Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A wrapper method for feature selection using Support Vector Machines

Sebastián Maldonado, Richard Weber *

Department of Industrial Engineering, University of Chile, P.O. Box 2777, República 701, Santiago de Chile, Chile

ARTICLE INFO

Article history:
Received 6 June 2008
Received in revised form 16 February 2009
Accepted 21 February 2009

Keywords:
Feature selection
Wrapper methods
Classification
Support Vector Machines
Mathematical programming

ABSTRACT

We introduce a novel wrapper 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 like a filter method or Recursive Feature Elimination SVM to demonstrate its effectiveness and efficiency.

1. Introduction

Feature selection [2,7,10] is of considerable importance in classification. The reason for being so is twofold: to reduce the computational complexity and to improve the classifier's generalization ability on one side. This reason is quite evident, since high-dimensional feature vectors impose a high computational cost and a high cost of data acquisition. On the other side a low-dimensional representation reduces the risk of *overfitting* [6,11]. Feature selection addresses the dimensionality reduction problem by determining a subset of available features to build a good model for classification or prediction, which is a combinatorial problem in the number of original features [7,13].

Support Vector Machines (SVMs) [20] is an effective classification method with significant advantages such as the absence of local minima, an adequate generalization to new objects, and a representation that depends on few parameters [5,17,20]. This method, however, does not directly determine the importance of the features used. In the present paper we introduce a new feature selection method for binary classification using SVM and compare it to existing approaches.

This paper is organized as follows. In Section 2 we briefly introduce SVM for binary classification. Section 3 provides an overview on recent developments for feature selection using SVM. Section 4 introduces the proposed feature selection method based on SVM. Experimental results using four real-world data sets are given in Section 5. Section 6 summarizes this paper by providing its main conclusions and addresses future developments.

2. Support Vector Machines for binary classification

In this section we describe the mathematical derivation of SVMs developed by Vapnik [20]. This technique is introduced in the following three steps. We first consider the simplest case, a linear classifier for a linearly separable problem. Then we

* Corresponding author. Tel.: +56 2 9784072; fax: +56 2 678 7895.
E-mail addresses: semaldon@ing.uchile.cl (S. Maldonado), rweber@dii.uchile.cl (R. Weber).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.02.014

look at linear classifiers for linearly non-separable problems. Finally a non-linear classifier for linearly non-separable problems, which is the most interesting and useful case, is presented. This non-linear classifier builds the basis for the approach proposed in this paper.

2.1. Linear classifier for linearly separable problems

For the linearly separable case, SVM determines the optimal hyperplane that separates the training patterns. The optimal hyperplane maximizes the sum of its distances to the closest positive and negative training patterns, respectively. This sum is called *margin*. To construct the maximum margin or optimal separating hyperplane, we need to correctly classify the vectors $\mathbf{x}_i$ of the training set into two different classes y_i, using the smallest norm of coefficients [20]. This problem can be formulated as follows:

$$\underset{\mathbf{w},b}{\text{Min}} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{1}$$

subject to

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

In order to explain the extension to non-linear classifiers (which will be described in Section 2.3) easily, we look at the dual formulation of the problem, using the technique of Lagrange multipliers. We construct the Lagrangian:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i \left[y_i\left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) - 1\right], \tag{2}$$

where $\boldsymbol{\alpha}$ is the vector of Lagrange multipliers corresponding to the constraints in (1). Applying the Karush–Kuhn–Tucker (KKT) conditions we obtain:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0, \tag{3}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{4}$$

The complementary slackness conditions take the following form:

$$\alpha_i^*[y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1] = 0, \quad i = 1, \ldots, m. \tag{5}$$

Notice that complementary slackness conditions of this form imply that $\alpha_i > 0$ are called *Support Vectors*. From Eqs. (3) and (5) it follows that $\mathbf{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \mathbf{x}_i$ and $b^* = y_i - \mathbf{w}^* \cdot \mathbf{x}_i$ for any Support Vector $\mathbf{x}_i$. The decision function can be written as follows:

$$f(\mathbf{x}) = sign(\mathbf{w}^* \cdot \mathbf{x} + b^*) = sign\left(\sum_{i=1}^{m} y_i \alpha_i^* (\mathbf{x} \cdot \mathbf{x}_i) + b^*\right). \tag{6}$$

Finally, the dual formulation of (1) becomes:

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s \mathbf{x}_i \cdot \mathbf{x}_s \tag{7}$$

subject to

$$\sum_{i=1}^{m} \alpha_i y_i = 0,$$
$$\alpha_i \geqslant 0, \quad i = 1, \ldots, m.$$

2.2. Linear classifier for linearly non-separable problems

We now consider the case in which a linear separating hyperplane does not exist, i.e. it is not possible to satisfy all the constraints in problem (1).

In order to weight the cost of misclassification an additional set of variables $\xi_i, i = 1, \ldots, m$ is introduced. The SVM procedure aims at solving the following optimization problem:

$$\underset{\mathbf{w},b,\xi}{\text{Min}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \xi_i \tag{8}$$

subject to

$$\begin{aligned} &y_i \cdot (\mathbf{w}^T \cdot \mathbf{x}_i + b) \geqslant 1 - \xi_i, \quad && i = 1, \ldots, m, \\ &\xi_i \geqslant 0, && i = 1, \ldots, m. \end{aligned}$$

The decision function remains $f(\mathbf{x}) = sign(\sum_{i=1}^{m} y_i \alpha_i^* (\mathbf{x} \cdot \mathbf{x}_i) + b^*)$, where $b^* = y_i - \mathbf{w}^* \cdot \mathbf{x}_i$ for any Support Vector $\mathbf{x}_i$ such that $0 < \alpha_i < C$ (a Support Vector which is correctly classified).

2.3. Non-linear classifier

For the non-linear case, SVMs map the data points into a higher dimensional space $\mathscr{H}$ where a separating hyperplane with maximal margin is constructed. The following quadratic optimization problem has to be solved.

$$\underset{\mathbf{w},b,\xi}{\text{Min}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \tag{9}$$

subject to

$$\begin{aligned} &y_i \cdot (\mathbf{w}^T \cdot \phi(\mathbf{x}_i) + b) \geqslant 1 - \xi_i, \quad && i = 1, \ldots, m, \\ &\xi_i \geqslant 0, && i = 1, \ldots, m. \end{aligned}$$

where training data are mapped to the higher dimensional space $\mathscr{H}$ by the function $\mathbf{x} \to \phi(\mathbf{x}) \in \mathscr{H}$ and C is a penalty parameter on the training error [4]. Under this mapping the solution obtained by applying SVM has the form:

$$f(\mathbf{x}) = sign(\mathbf{w}^* \cdot \phi(\mathbf{x}) + b^*) = sign\left(\sum_{i=1}^{m} y_i \alpha_i^* \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i) + b^*\right). \tag{10}$$

The only values one needs to compute are scalar products of the form $\phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ [16]. The mapping is performed by a kernel function $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ which defines an inner product in $\mathscr{H}$. The decision function $f(\mathbf{x})$ given by SVM is thus:

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{m} y_i \alpha_i^* K(\mathbf{x}, \mathbf{x}_i) + b^*\right). \tag{11}$$

The optimal hyperplane is the one with maximal distance (in $\mathscr{H}$) to the closest image $\phi(\mathbf{x}_i)$ from the training data. The dual formulation can be reformulated as follows:

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,s=1}^{m}\alpha_i\alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \tag{12}$$

subject to

$$\begin{aligned} &\sum_{i=1}^{m}\alpha_i y_i = 0, \\ &0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, m. \end{aligned}$$

Among a variety of existing kernel functions, the polynomial and the radial basis function are chosen in many applications [17]:

(1) Polynomial function: $K(\mathbf{x}_i, \mathbf{x}_s) = (\mathbf{x}_i \cdot \mathbf{x}_s + 1)^d$, where $d \in \mathbb{N}$ is the degree of the polynomial.
(2) Radial basis function (RBF): $K(\mathbf{x}_i, \mathbf{x}_s) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_s\|^2}{2\rho^2}\right)$, where $\rho > 0$ is the parameter controlling the width of the kernel.

3. Feature selection with SVM criterion

Three main directions have been developed for feature selection: filter, wrapper, and embedded methods [7,19]. Subsequently, we provide a brief overview on each one of these directions and present the techniques that have been compared with our approach proposed in this paper. The first scheme (*filter methods*) uses statistical properties of the features to filter out poorly informative ones. This is done before applying any classification algorithm.

The Fisher Criterion Score (F) is such a filter method which computes the importance of each feature independently of the other features by comparing that feature's correlation to the output labels. The respective score $F(j)$ of feature j is given by:

$$F(j) = \left|\frac{\mu_j^+ - \mu_j^-}{(\sigma_j^+)^2 + (\sigma_j^-)^2}\right|, \tag{13}$$

where $\mu_j^+(\mu_j^-)$ is the mean value for the jth feature in the positive (negative) class and σ_j^+ (σ_j^-) is the respective standard deviation.

A second approach (*wrapper methods*) is computationally demanding, but often provides more accurate results than filter methods. A wrapper algorithm explores the feature space to score feature subsets according to their predictive power, optimizing the subsequent induction algorithm that uses the respective subset for classification.

A wrapper method called Recursive Feature Elimination (RFE-SVM) is a feature selection algorithm described by Guyon et al. [8]. In this paper we use the version that includes Kernel functions as described in [7,10] and in [18] for multi-class. The goal is to find a subset of size r among n variables ($r < n$) which maximizes the performance of the predictor, using an SVM classifier. The method, given that one wishes to employ only $r < n$ input variables in the final decision rule, attempts to find the best subset of r features. It operates by trying to choose the r features which lead to the largest margin of class separation. This problem is based on a sequential backward selection, removing one feature at a time until r features remain. The removed feature is the one whose elimination minimizes the variation of $W^2(\boldsymbol{\alpha})$:

$$W^2(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s). \tag{14}$$

This is a measure of the model's predictive ability and is inversely proportional to the margin. Feature elimination is done applying the following procedure:

(1) Given a solution $\boldsymbol{\alpha}$, calculate for each feature p:

$$W^2_{(-p)}(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K\left(\mathbf{x}_i^{(-p)}, \mathbf{x}_s^{(-p)}\right), \tag{15}$$

where $\mathbf{x}_i^{(-p)}$ means training object i with feature p removed.

(2) Remove the feature with smallest value of $\left|W^2(\boldsymbol{\alpha}) - W^2_{(-p)}(\boldsymbol{\alpha})\right|$.

The third approach (*embedded methods*) performs feature selection in the process of model building. For example, [12] adds an extra term that penalizes the size of the selected feature subset to the standard cost function of SVM, and optimizes the new objective function to select features. These approaches are, however, limited to linear kernels.

Another approach for feature penalization is the so-called l_0-SVM or Concave Feature Selection (FSV) [3], based on the minimization of the "zero norm": $\|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|$. Note that $\|\cdot\|_0$ is not a norm because, unlike l_p-norms with $p > 0$, the triangle inequality does not hold [3]. Since the l_0-"norm" is non-smooth, it was approximated by the concave function:

$$\|\mathbf{w}\|_0 \approx \mathbf{e}^T(\mathbf{e} - \exp(-\beta|\mathbf{w}|), \tag{16}$$

with approximation parameter $\beta \in \Re_+$ and $\mathbf{e}$ the vector $(1, \ldots, 1)^T$. The l_0-SVM (FSV) formulation follows:

$$\underset{\mathbf{w},\mathbf{v},b,\xi}{\text{Min}} \quad \sum_{j=1}^{n} [1 - exp(-\beta v_j)] + C \sum_{i=1}^{m} \xi_i \tag{17}$$

subject to

$$\begin{aligned} & y_i \cdot (\mathbf{w}^T \cdot \mathbf{x}_i + b) \geqslant 1 - \xi_i, && i = 1, \ldots, m, \\ & -v_j \leqslant w_j \leqslant v_j, && j = 1, \ldots, n, \\ & \xi_i \geqslant 0, && i = 1, \ldots, m. \end{aligned}$$

This embedded method can be used to establish a feature ranking in order to compare its feature selection performance with other wrapper methods [3].

4. The proposed method for feature selection using SVM

We propose a method for feature selection using SVM and a specific kernel function. It starts with all available features and determines each feature's contribution to the respective classifier. The one with the least impact on the classification performance in an independent validation subset will be removed in each iteration until a stopping criterion indicates that a good solution has been found.

After providing the relevant notation we introduce the proposed method for feature selection which subsequently will be related to alternative techniques at the end of this section.

4.1. Notation and preliminaries

The componentwise vector product operator $*$ is defined as [21]:

$$\mathbf{a} * \mathbf{b} = (a_1 b_1, \ldots, a_n b_n). \tag{18}$$

The binary vector $\boldsymbol{\sigma}, \boldsymbol{\sigma} \in \{0,1\}^n$, acts as an indicator for feature selection that will be multiplied componentwise with the input objects.

Thus the kernel function as explained in Section 2.3 becomes:

$$K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \equiv K(\boldsymbol{\sigma} * \mathbf{x}_i, \boldsymbol{\sigma} * \mathbf{x}_s). \tag{19}$$

We use this vector $\boldsymbol{\sigma}$ as a parameter for feature selection and, for a given $\boldsymbol{\sigma}$, we solve (15), the dual formulation of SVM, whose mathematical derivation is shown in Section 2.2.4:

$$\operatorname*{Max}_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \tag{20}$$

subject to

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

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, m.$$

4.2. Hold-out Support Vector Machines (HO-SVM): a novel wrapper method for feature selection

The basic idea of the proposed method is to remove features whose elimination implies only a small number of errors in a validation subset independent of the training data. This is achieved by the following iterative algorithm:

Algorithm 1. HO-SVM Algorithm for Feature Selection

```
(1) Model selection
(2) Initialization
(3) **repeat**
    (a) Random split of the training data
    (b) SVM Training
    (c) **for** each feature $p$ with $\sigma_p = 1$, **do** determine $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, the number of classification errors when feature $p$ is removed
    (d) remove feature $j$ with the smallest value of $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$
(4) **until** the smallest value of $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is greater than $E(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, which is the number of errors in the Validation subset using all features as indicated by the current vector $\sigma$, i.e. without removing any further feature.
```

In order to give a complete description of the methodology, we detail each step of the previous algorithm:

Model selection: The first step is to determine the parameters for SVM (C for error penalization, the polynomial degree d or the Gaussian kernel parameter ρ) when all features are selected. In our experiments we perform SVM without feature selection in order to identify the best Kernel function for the algorithm.

Initialization: We set $\boldsymbol{\sigma} = (1, \ldots, 1)$, which means we start with all features and in each iteration we remove the feature with the smallest contribution to the respective model.

Random split of the data: We split the training data set into two subsets: a Training Subset (with approximately 70% of the observations) and a Validation Subset (with the remaining 30% of the observations). We perform SVM on the Training subset for the current features obtaining a certain solution. Using this solution we then evaluate each feature's contribution on the Validation Subset. The percentage of observations in each subset can be treated as an additional parameter of the algorithm.

SVM Training: We train a SVM classifier (Eq. (20)) with the training subset and the features as indicated by the vector $\boldsymbol{\sigma}$.

Calculate $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$: **for** each feature p with $\sigma_p = 1$, **do** calculate:

$$E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \sum_{l \in VAL} \left| y_l^v - sgn\left(\sum_{i \in TRAIN} \alpha_i y_i K_{\boldsymbol{\sigma}}\left(\mathbf{x}_i^{(-p)}, \mathbf{x}_l^{v(-p)}\right) + b \right) \right|, \tag{21}$$

where VAL is the Validation subset and $\mathbf{x}_l^v$ and y_l^v are the objects and labels of this subset, respectively. $\mathbf{x}_i^{(-p)}$ ($\mathbf{x}_l^{v(-p)}$) means training object i (validation object l) with feature p removed. $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is the number of errors in the Validation Subset when feature p is removed, using the currently selected features as indicated by $\boldsymbol{\sigma}$.

To reduce computational complexity of the proposed algorithm, we use the same approximation as in [8]: the vector $\boldsymbol{\alpha}$ used in (21) is supposed to be equal to the solution of (20) even if a feature has been removed.

Criterion for feature elimination: Remove feature j (i.e. set $\sigma_j = 0$) with the smallest value of $E_{(-j)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$. Feature j with the smallest value $E_{(-j)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is the one whose elimination implies the least number of errors in the Validation Subset, so it is considered to be irrelevant.

Stopping criterion: The algorithm stops when the smallest value of the measure $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is greater or equal than $E(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, the number of errors in the Validation Subset without removing any feature. Alternatively, we can modify this criterion in order to remove more or less features. A stronger criterion would indicate stopping when the smallest value of $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is strictly greater than $E(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, that means, in case of a tie between the smallest value of $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ and $E(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, the algorithm will keep iterating and removing features.

The training and validation subsets change in each iteration due to the random split of the data. Since the output of the algorithm in each iteration is only the vector $\boldsymbol{\sigma}$ of selected features, it is not necessary to explicit the iteration index in the formula.

Fig. 1 illustrates the proposed process of feature selection for SVMs, which we call HO-SVM (Hold-out SVM).

4.3. Relation to other SVM-based feature selection methods

Several algorithms for feature selection based on SVM are already available. RFE-SVM and other wrapper methods presented in [1,14] differ regarding the measure to decide which feature to remove in each iteration and the stopping criterion. Our approach uses the number of generalization errors (applying the hold-out technique) instead of a measure based only on one data set ($W^2(\boldsymbol{\alpha})$ [8], a gradient-based measure [14]) or a measure based on the Fisher Correlation Score [1]. The intuition behind this proposed measure is that we can improve the classification performance by removing the features that directly affect on generalization (classification errors on an independent subset) of the classifier instead of a measure that only considers the training data. Additionally, our method presents an explicit stopping criterion, unlike the other wrapper algorithms cited.

Compared to the wrapper method RFE-SVM, our approach requires at least the same computational effort. In each iteration we train a SVM classifier with the training data subset and for each feature we evaluate a function on the test data subset. RFE-SVM does the same but with the entire data set, therefore the order of both algorithms is the same. Additionally, our approach splits the data set in each iteration. The explicit stopping criterion reduces computational efforts to determine when the elimination of features affects negatively the model's performance.

Embedded methods differ from other feature selection methods in the way feature selection and learning interact. In contrast to our approach, in embedded methods such as [3,12,23] the learning part and the feature selection part can not be separated. The method proposed by Weston et al. [21] differs from ours in the objective function (they minimize the R^2W^2 bound on the leave-one-out error *LOO* of a trained hard margin SVM classifier instead of the number of errors in an independent subset) and the variable space search algorithm: instead of using a greedy algorithm, they use a gradient descent to minimize this bound.

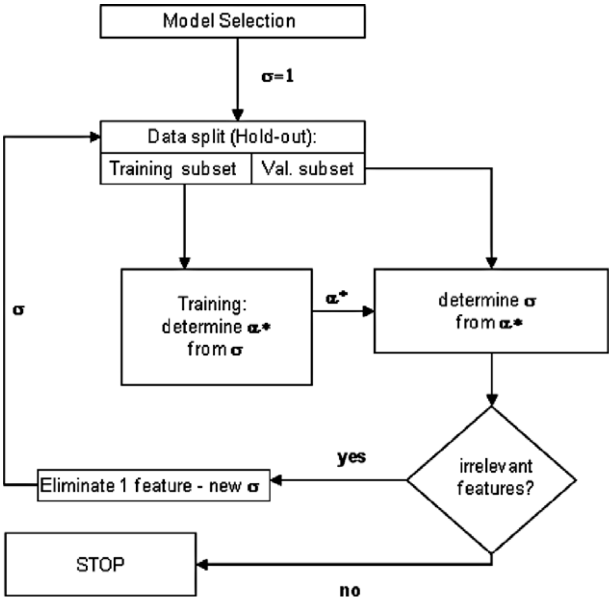

Fig. 1. Feature Selection using HO-SVM.

5. Experimental results

The proposed approach has been applied for feature selection on four data sets, two well-known benchmark data sets used in [14,15] and two from projects that have been performed for Chilean financial institutions. The methodology we followed consisted in each case in: (1) model selection in order to obtain the best Kernel and for parameter setting, (2) variable ranking, and (3) measuring the test error of an SVM classifier when this predictor is provided with an increasing number of ranked variables. A mean test error is then obtained by averaging the results over 100 realizations, as described in [14,15]. For this procedure we use the Spider Toolbox for Matlab [22].

Next, we describe briefly the mentioned data sets and provide then the classification results using different feature selection methods.

5.1. Description of data sets

Wisconsin Breast Cancer (WBC): This data set from the UCI data repository [9] contains 569 observations (212 malignant and 357 benign tumors) described by 30 continuous features. Wisconsin Breast Cancer was created by William H. Wolberg from the General Surgery Department of the University of Wisconsin and by W. Nick Street and Olvi L. Mangasarian from the Computer Sciences Department of the same University, and donated in November 1995. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. As preprocessing step the features were scaled between 0 and 1. The data set does not contain missing values.

Colorectal Microarray data set (CRMA): This data set contains the expression of the 2000 genes with highest minimal intensity across the 62 tissues (40 tumor and 22 normal). The genes are placed in order of descending minimal intensity. Following the procedure of [14,21], the training set and the test set are obtained by splitting the data set into 2 groups of 40 and 22 elements, respectively, while ensuring that the proportions of positive and negative classes are similar in both sets. For the other data sets we split the data into 2 groups of approximately 60% of the observations for the training set and the remaining 40% for the test subset.

In order to speed up the procedure of our approach, 20 variables are removed at each step until 100 variables remain still to be ranked. Then features are removed one at a time. We compare our results choosing the number of variables as mentioned in [21]: 20, 50, 100, 250, 1000 and 2000 (no variables removed).

INDAP data set: The third data set stems from a credit scoring project performed for the Chilean organization INDAP and is based on 49 variables describing 1464 observations (767 good and 697 bad customers) [4]. INDAP is the main service provided by the Chilean government that aims at supporting small agricultural enterprises; see www.indap.cl. It was founded in 1962 and has more than 100 offices all over Chile serving its more than 100,000 customers.

The available data set contained all credits that have been accepted between 2004 and 2006. In order to keep the results of our analysis unbiased, the classes of this data set were balanced using random sampling. The observations with missing values were deleted and some irrelevant features were filtered out using univariate feature selection.

BDDM data set: A credit scoring system has been developed for the Micro-Enterprises Division of the Chilean bank *Banco del Desarrollo* (*BDD*); see: www.bdd.cl. This bank belongs to the Scotiabank Group. The Micro-Enterprises Division (www.bddm.cl) is specialized on credits for micro-entrepreneurs and has a market coverage of approximately 30% in 2007. The goal of the mentioned project was to develop a system for automatic credit scoring that should reduce the time needed for revision of a credit.

The available data set contained all credits that have been accepted between 2004 and 2006. For each one of these 4780 credits a decision had to be taken so as to classify the customer's behavior as *good* or *bad*. Additionally, we had a set of 677 features describing the respective credits, customers, and their paying behavior in the past (in the case of customers who already had a credit previously).

After data cleaning, data set balancing and univariate feature selection using simple statistical tests as filters we obtained a data set with 3003 credits (rows of the data matrix) and 24 features (columns of the data matrix). To this data set we applied our proposed feature selection approach as well as other techniques described above.

5.2. Results

First, we compared the results of the best model found on the model selection procedure by each Kernel function. Table 1 presents mean and standard deviation of the test error using 10-fold cross-validation. We use the following set of values for the parameters:

$$C = \{0.1, 0.5, 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 1000\},\ d = \{2, 3, 4, 5, 6, 7, 8, 9\} \text{ and}$$
$$\rho = \{0.1, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 100\}.$$

In this step we prove that for our data sets the best Kernel is the Gaussian.

We compared the classification performance of the different feature selection methods presented in this paper (RFE-SVM, FSV and our approach HO-SVM). Furthermore, we applied the filter technique Fisher Criterion Score (Fisher) for feature selection prior to classifier design. Figs. 2–5 display the mean test error for an increasing number of ranked features used

Table 1
Mean and standard deviation of effectiveness on four data sets using three different SVM with different Kernel functions.

	SVM linear	SVM poly	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

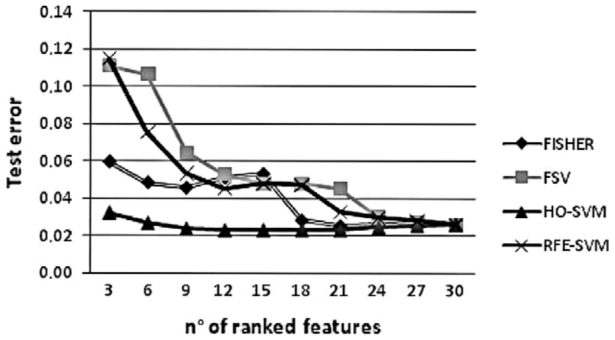

Fig. 2. Mean test errors for WBC vs. the number of ranked variables used for training.

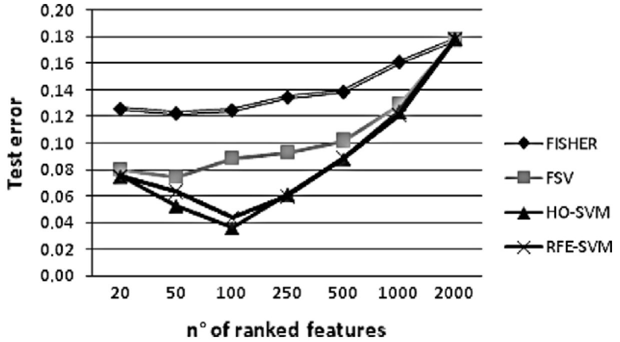

Fig. 3. Mean test errors for CRMA vs. the number of ranked variables used for training.

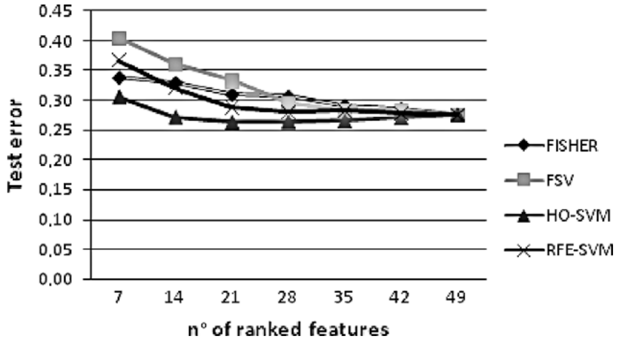

Fig. 4. Mean test errors for INDAP vs. the number of ranked variables used for training.

for learning. They show that HO-SVM outperforms the other methods in all 4 data sets in terms of mean classification error in the respective test sets for all analyzed numbers of selected features.

In order to emphasize the importance of HO-SVM's stopping criterion, we study the performance of each feature selection algorithm when it reaches this number of features. Table 2 shows the mean and standard deviation of the effectiveness at this point on our four data sets.

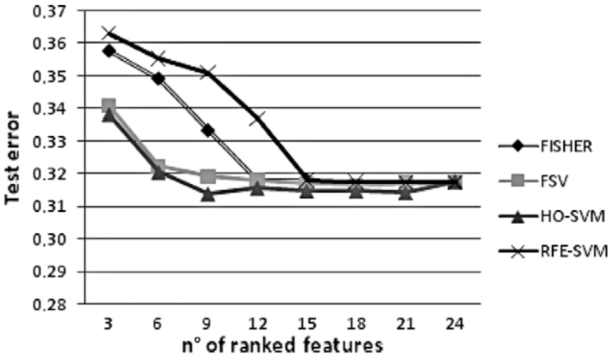

Fig. 5. Mean test errors for BDDM vs. the number of ranked variables used for training.

Table 2
Number of selected features, mean and standard deviation of effectiveness using four different feature selection methods on four data sets.

	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**

As can be concluded from Table 2, the proposed method outperforms all other approaches in terms of classification error for a given number of features (our stopping criterion). The gain in terms of effectiveness is significant in all cases. The second best method is RFE-SVM but it fails on the BDDM data set.

6. Conclusions and future work

We presented a novel wrapper approach for feature selection using SVM. This method performs a sequential backward elimination of features, using the number of errors in a validation subset as the measure to decide which feature to remove in each iteration.

A comparison with other techniques for feature selection and classification shows the advantages of our approach:

- It outperforms other filter and wrapper methods, based on its ability to adjust better to a data set because of the validation error measure, but avoiding overfitting doing a random split of the data set in each iteration.
- It presents an explicit stopping criterion, indicating clearly when removing further features begins to affect negatively the performance of the classifier.
- It can be used with any Kernel function.
- It can be easily generalized to variations of SVM, such as Support Vector Regression and multi-class SVM.

An important characteristic of our method is that different runs of the algorithm may select different features. This is due to the random data split in each iteration. An unfortunate split of the data set may also remove an important feature, affecting thus negatively the classifier's performance. To avoid this situation, we recommend to perform 3-4 runs of the algorithm, compare the eliminated features and remove them only if they have been discarded in more than one run. We can also check the performance of the algorithm by analyzing the number of errors and identifying incorrectly removed features, improving the method's effectiveness.

Empirically we prove the method's robustness regarding feature selection by verifying that most of the time the same features are selected in different runs providing high classifier performance. For example, after running the proposed method five times on the WBC data set, 9 from the original 30 features have been selected five times. We also recommend to order the features in terms of relevance, using a fast filter method for example, before running the algorithm, in order to decide which variable to remove in case of equal number of validation errors. This point is particularly important in high-dimensional data sets with a small number of observations.

Our algorithm relies on a backward feature elimination, which is computationally treatable but expensive if the number of input features is large. We could improve its performance by applying filter methods for feature selection before running our wrapper algorithm [11,19]. This way we can identify and eliminate irrelevant features at low cost. In our Credit Scoring projects we use univariate analysis (Chi-Square Test for categorical features and the Kolmogorov–Smirnov Test for continuous ones) as a first filter for features selection with excellent results.

Future work has to be done in various directions. First, it would be interesting to use the proposed wrapper technique for feature selection in combination with the variations of SVM, such as different Kernel functions and Support Vector Regression. Second, it would be attractive to apply the approach HO-SVM together with weighted Support Vector Machines to compensate for the undesirable effects caused by unbalanced data sets in model construction; an issue which occurs frequently e.g. in the domains of fraud and intrusion detection.

Acknowledgements

Support from the Millennium Science Institute on Complex Engineering Systems (www.sistemasdeingenieria.cl) and the Project Fondecyt 1040926 is greatly acknowledged. The first author also acknowledges a grant provided by CONICYT for his Ph.D. studies in Engineering Systems.

References

[1] M.E. Blazadonakis, M. Zervakis, Wrapper filtering criteria via linear neuron and kernel approaches, Computers in Biology and Medicine 38 (8) (2008) 894–912.

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

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

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

[5] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK, 2000.

[6] A. Famili, W.-M. Shen, R. Weber, E. Simoudis, Data preprocessing and intelligent data analysis, Intelligent Data Analysis 1 (1997) 3–23.

[7] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

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

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

[10] T.N. Lal, O. Chapelle, J. Weston, A. Elisseeff, Embedded methods, in: I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh (Eds.), Feature Extraction: Foundations and Applications. Studies in Fuzziness and Soft Computing, vol. 207, Springer, Berlin, Heidelberg, 2006, pp. 137–165.

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

[12] J. Miranda, R. Montoya, R. Weber, Linear penalization support vector machines for feature selection, in: S.K. Pal et al. (Eds.), PReMI 2005, LNCS, vol. 3776, Springer-Verlag, Berlin Heidelberg, 2005, pp. 188–192.

[13] G. Nemhauser, L. Wolsey, Integer and Combinatorial Optimization, John Wiley and Sons, New York, 1988.

[14] A. Rakotomamonjy, Variable selection using SVM-based criteria, Journal of Machine Learning Research 3 (2003) 1357–1370.

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

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

[17] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, Cambridge, 2004.

[18] M-D. Shieh, C-C. Yang, Multiclass SVM-RFE for product form feature selection, Expert Systems with Applications 35 (1-2) (2008) 531–541.

[19] Ö. Uncu, I.B. Türksen, A novel feature selection approach: combining feature wrappers and filters, Information Sciences 177 (2007) 449–466.

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

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

[22] J. Weston, A. Elisseeff, G. BakIr, F. Sinz, The spider. <http://www.kyb.tuebingen.mpg.de/bs/people/spider/>.

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