Intelligent Data Analysis 19 (2015) 1259–1273
DOI 10.3233/IDA-150781
IOS Press

An embedded feature selection approach for support vector classification via second-order cone programming

Sebastián Maldonado[a,*] and Julio López[b]
[a]*Facultad de Ingeniería, Universidad Diego Portales, Ejército, Santiago, Chile*
[b]*Universidad de los Andes, Mons. Álvaro del Portillo, Las Condes, Santiago, Chile*

Abstract. Feature selection is an important machine learning topic, especially in high dimensional applications, such as cancer prediction with microarray data. This work addresses the issue of high dimensionality of feature selection for linear and kernel-based Support Vector Machines (SVMs) considering second-order cone programming formulations. These formulations provide a robust and efficient framework for classification, while an adequate feature selection process avoids errors in the estimation of means and covariances. Our approach is based on a sequential backward elimination which uses different linear and kernel-based contribution measures to determine the feature relevance. Experimental results with microarray datasets demonstrate the effectiveness in terms of predictive performance and construction of a low-dimensional data representation.

Keywords: Second-order cone programming, Support Vector Machines, feature selection, kernel methods, data mining

1. Introduction

Feature selection is one of the most important machine learning tasks. An appropriate selection of the most relevant features reduces the risk of overfitting, improving model generalization by decreasing the model's complexity [16]. This is particularly important in small-sized high-dimensional datasets, where the *curse of dimensionality* is present and a significant gain in terms of performance can be achieved with a small subset of features [16,26]. Additionally, low-dimensional representation allows better interpretation of the classifier. This is particularly important in some application fields like business analytics, since machine learning approaches are considered to be *black boxes* by practitioners, who therefore tend to be hesitant to use these techniques [10,12]. The understanding of the process that generates the data is also of crucial importance in the life sciences, e.g., for selection of the relevant genes that lead to better discrimination in cancer prediction.

Support Vector Machine (SVM) has shown to be a very powerful machine learning method. SVM provides several advantages such as adequate generalization to new objects, a flexible non-linear decision boundary, absence of local minima, and representation that depends on only a few parameters [35].

*Corresponding author: Sebastián Maldonado, Facultad de Ingeniería, Universidad Diego Portales, Ejército 441, Santiago, Chile. E-mail: smaldonado@uandes.cl.

Recently, second-order cone programming (SOCP) formulations have been proposed as a robust optimization scheme for SVMs. These formulations have several theoretical advantages that may lead to superior predictive performance, compared to the standard SVM method [23,27].

This work presents an embedded feature selection strategy for SOCP-SVM, proposing novel contribution measures to assess feature relevance. Two algorithms are described: one for the linear version of SOCP-SVM, and one for the kernel-based SOCP-SVM formulation. Based on robust optimization, our proposals combine feature selection and classification via Second-order cone programming, achieving better classification performance than other embedded feature selection techniques for SVM.

The paper is structured as follows: Section 2 introduces Support Vector Machines for binary classification, and its robust formulation with second-order cones. Recent developments for feature selection are reviewed in Section 3. The proposed feature selection approach is presented in Section 4. Section 5 provides experimental results using real-world datasets. A summary of this paper can be found in Section 6, where we provide its main conclusions and address future developments.

2. Support Vector Machine

In this section we describe the mathematical derivation of SVM developed by Vapnik [35], and SOCP-based Support Vector Machine formulation based on the first two moments of each class, the mean and covariance [27].

2.1. l_2 Support Vector Machine

Considering training examples $\mathbf{x}_i \in \Re^n$, $i = 1, \ldots, m$ and their respective labels $y_i \in \{-1, 1\}$, SVM determines a hyperplane $f(\mathbf{x}) = \mathbf{w}^\top \cdot \mathbf{x} + b$ such that each vector has to be correctly classified into one of the two classes. This hyperplane maximizes the *margin*, which is computed as the sum of the distances to the closest positive and negative training instances. To maximize this measure, we need to classify the training vectors $\mathbf{x}_i$ correctly into two different classes, y_i, using the smallest norm of coefficients $\mathbf{w}$ [35]. The primal SVM formulation balances the minimization of the Euclidean norm (structural risk) and the misclassification errors (empirical risk) by introducing an additional set of slack variables ξ_i, $i = 1, \ldots, m$ and a penalty parameter C that controls this trade-off:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i \cdot (\mathbf{w}^\top \cdot \mathbf{x}_i + b) \geqslant 1 - \xi_i, \qquad i = 1, \ldots, m, \\ & \xi_i \geqslant 0, \qquad i = 1, \ldots, m. \end{aligned} \tag{1}$$

A non-linear classifier can be obtained by mapping the data instances into a higher dimensional space $\mathscr{H}$, where a separating hyperplane with maximal margin is constructed. The mapping is performed by a kernel function $K(\mathbf{x}, \mathbf{y})$ which defines an inner product in $\mathscr{H}$. The kernel-based SVM formulation can be stated as follows:

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

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0, \\ & 0 \leqslant \alpha_i \leqslant C, \qquad i = 1, \ldots, m, \end{aligned} \tag{2}$$

where $\boldsymbol{\alpha}$ is the vector of Lagrange multipliers corresponding to the constraints in Eq. (1). We based our analysis on the *gaussian kernel*, which has the following form:

$$K(\mathbf{x}_i, \mathbf{x}_s) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_s||^2}{2\sigma^2}\right), \tag{3}$$

where $\sigma > 0$ is the parameter controlling the width of the kernel [30].

2.2. SOCP Support Vector Machine

Suppose that $\mathbf{X}_1$ and $\mathbf{X}_2$ are random vector variables that generate samples of the positive and negative classes respectively. In order to construct a maximum margin linear classifier such that the probability of a false-negative and a false-positive error does not exceed $1 - \eta_1$ and $1 - \eta_2$ respectively, with $\eta_1, \eta_2 \in (0, 1)$, let us consider the following quadratic chance-constrained programming (QCCP) problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_1 - b \leqslant 0\} \leqslant 1 - \eta_1, \\ & \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_2 - b \geqslant 0\} \leqslant 1 - \eta_2. \end{aligned} \tag{4}$$

In other words, we require that the random variable $\mathbf{X}_i$ lies on the correct side of the hyperplane with probability greater than η_i for $i = 1, 2$. Assume that for $i = 1, 2$ we *only know* the mean $\boldsymbol{\mu}_i \in \Re^n$ and covariance matrix $\Sigma_i \in \Re^{n \times n}$ of the random vector $\mathbf{X}_i$. In this case, for each $i = 1, 2$ we want to be able to classify correctly, up to the rate η_i, even for the *worst distribution* in the class of distributions which have common mean and covariance $\mathbf{X}_i \sim (\boldsymbol{\mu}_i, \Sigma_i)$. For this purpose we replace the probability constraints in Eq. (4) with their *robust* counterparts:

$$\sup_{\mathbf{X}_1 \sim (\boldsymbol{\mu}_1, \Sigma_1)} \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_1 - b \leqslant 0\} \leqslant 1 - \eta_1, \quad \sup_{\mathbf{X}_2 \sim (\boldsymbol{\mu}_2, \Sigma_2)} \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_2 - b \geqslant 0\} \leqslant 1 - \eta_2.$$

By virtue of an appropriate application of the multivariate Chebyshev inequality, this worst distribution approach leads to the following quadratic second-order cone programming (QSOCP) problem [3], which is the deterministic formulation of Eq. (4) (see [5,27] for all details):

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \mathbf{w}^\top \cdot \boldsymbol{\mu}_1 - b \geqslant 1 + \kappa_1 \sqrt{\mathbf{w}^\top \cdot \Sigma_1 \mathbf{w}}, \\ & b - \mathbf{w}^\top \cdot \boldsymbol{\mu}_2 \geqslant 1 + \kappa_2 \sqrt{\mathbf{w}^\top \cdot \Sigma_2 \mathbf{w}}, \end{aligned} \tag{5}$$

where $\kappa_i = \sqrt{\frac{\eta_i}{1-\eta_i}}$, for $i = 1, 2$.

Now, we denote the number of elements of the positive and negative class by m_1 and m_2 respectively; by $T_1 \in \Re^{n \times m_1}$, a data matrix for the positive class; by $T_2 \in \Re^{n \times m_2}$, a data matrix for the negative class; by $\mathbf{1}_{m_i}$, a vector of ones of dimension m_i; and by $\mathbf{I}_{m_i}$, the identity matrix in $\Re^{m_i \times m_i}$. Since $\mathbf{w} \in \Re^n$, it can be written as $\mathbf{w} = [T_1, T_2]\mathbf{s} + M\mathbf{r}$, where M is a matrix with its columns as vectors orthogonal to training data points and $\mathbf{s}, \mathbf{r}$ are vectors of combining coefficients. On the other hand, the empirical estimates of the mean and covariance are given by:

$$\boldsymbol{\mu}_i = \bar{\mathbf{x}}_i = \frac{1}{m_i} T_i \mathbf{1}_{m_i}, \quad \Sigma_i = \bar{\Sigma}_i = S_i S_i^\top \quad \text{with} \quad S_i = \frac{1}{\sqrt{m_i}}(T_i - \boldsymbol{\mu}_i \mathbf{1}_{m_i}^\top),$$

for $i = 1, 2$. Thus

$$\mathbf{w}^\top \cdot \boldsymbol{\mu}_i = \mathbf{s}^\top \cdot \mathbf{g}_i, \quad \mathbf{w}^\top \cdot \Sigma_i \mathbf{w} = \mathbf{s}^\top \cdot \mathbf{G}_i \mathbf{s}, \quad i = 1, 2,$$

where

$$\mathbf{g}_i = \frac{1}{m_i} \begin{bmatrix} \mathbf{K}_{1i} \mathbf{1}_i \\ \mathbf{K}_{2i} \mathbf{1}_i \end{bmatrix}, \quad \mathbf{G}_i = \frac{1}{m_i} \begin{bmatrix} \mathbf{K}_{1i} \\ \mathbf{K}_{2i} \end{bmatrix} \left(I_{m_i} - \frac{1}{m_i} \mathbf{1}_i \mathbf{1}_i^\top \right) \begin{bmatrix} \mathbf{K}_{1i}^\top \, \mathbf{K}_{2i}^\top \end{bmatrix},$$

with $\mathbf{K}_{11} = T_1^\top T_1$, $\mathbf{K}_{12} = \mathbf{K}_{21}^\top = T_1^\top T_2$, $\mathbf{K}_{22} = T_2^\top T_2$ matrices whose elements are inner products of data points. For instance, the entry (i, j) for the matrix $\mathbf{K}_{12}$ is

$$(\mathbf{K}_{12})_{ij} = \langle \mathbf{x}_i^1, \mathbf{x}_j^2 \rangle. \tag{6}$$

Hence, in order to design nonlinear classifiers, we replace the inner product, $\langle \cdot, \cdot \rangle$, by $K(\cdot, \cdot)$. For instance, the inner product Eq. (6) is replaced by $(\mathbf{K}_{12})_{i,j} = K(\mathbf{x}_i^1, \mathbf{x}_j^2)$. Thus, the non-linear formulation is given by

$$\begin{aligned} \min_{\mathbf{s},b} \quad & \frac{1}{2} \mathbf{s}^\top \cdot \mathbf{K} \mathbf{s} \\ \text{s.t.} \quad & \mathbf{s}^\top \cdot \mathbf{g}_1 - b \geqslant 1 + \kappa_1 \sqrt{\mathbf{s}^\top \cdot \mathbf{G_1} \mathbf{s}} \\ & b - \mathbf{s}^\top \cdot \mathbf{g}_2 \geqslant 1 + \kappa_2 \sqrt{\mathbf{s}^\top \cdot \mathbf{G_2} \mathbf{s}}, \end{aligned} \tag{7}$$

where $\mathbf{K} = [\mathbf{K}_{11}, \mathbf{K}_{12}; \mathbf{K}_{21}, \mathbf{K}_{22}]$.

3. Feature selection for machine learning

Three main strategies have been proposed for feature selection: filter, wrapper, and embedded methods [16]. Filter methods include algorithms that are independent of the classifier, filtering out irrelevant features based on statistical properties. In this work we use the Fisher Criterion Score as a benchmark approach. This method computes the correlation of each variable with the labels. The score $F(j)$ of feature j is given by:

$$F(j) = \left| \frac{\mu_j^+ - \mu_j^-}{(\sigma_j^+)^2 + (\sigma_j^-)^2} \right| \tag{8}$$

where μ_j^+ (μ_j^-) represents the mean for the j-th feature in the positive (negative) class and σ_j^+ (σ_j^-) is the respective standard deviation. Other common measures are Information Gain (also known as Mutual Information) [40], and the Hilbert-Schmidt Independence Criterion [32]. This strategy has advantages such as its simplicity, scalability, and a reduced computational effort; but it ignores the interactions between the variables, and the relationship between them and the classification algorithm.

Wrapper methods are search strategies which are wrapped around predictors to score feature subsets according to their predictive power (usually accuracy). Since the exhaustive search for an optimal subset of features grows exponentially with the number of original variables, heuristic approaches have been suggested [20]. Commonly used wrapper strategies are the Sequential forward selection (SFS), and the Sequential backward elimination (SBE) [20]. In the first case, each candidate variable is included in the current set, and the result is evaluated. The variable whose inclusion results in the best evaluation is inserted into the current set. Subsequently, SBE starts with the variable set that consists of all the candidate variables, and the variable whose exclusion results in the best evaluation is considered to be eliminated from the current set. An advantage of wrapper methods is the interaction between a subset of variables and the model. The main disadvantage is the high computational cost and the risk of overfitting [16].

Embedded methods encompass algorithms that perform feature selection in the training process, being specific to given machine learning methods [16]. Some approaches consider backward feature elimination in order to establish a ranking of features, using SVM-based contribution measures to evaluate their relevance. One popular method is known as Recursive Feature Elimination (RFE-SVM) [18]. The goal of this approach is to find a subset of size r among n variables ($r < n$), eliminating those features whose removal leads to the largest margin of class separation. Since the margin is inversely proportional to the Euclidean norm of the weight vector, this value can be rewritten in terms of the dual variables of SVM:

$$W^2(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s). \tag{9}$$

The feature to be removed in each iteration is the one whose removal minimizes the variation of $W^2(\boldsymbol{\alpha})$. While one could choose a single variable to remove at each iteration, this would be inefficient in many high dimensional applications (such as microarray data). In such datasets there are thousands of features, and the authors suggest removing half of the variables at each step [18].

Recent studies suggest that RFE-SVM achieves best (or close to best) performance in microarray datasets (see e.g. [8,36]), and therefore we base our proposal in this backward elimination strategy.

Embedded feature selection can also be seen as an optimization problem. This is generally done by enforcing feature selection on the parameter of the model directly, considering a sparsity term in the objective function. For instance, the squared Euclidean norm from the classical SVM ($\|\mathbf{w}\|^2$ from Formulation (1)) can be replaced by the l_1 norm $\Omega(\mathbf{w}) = \sum_i |w_i|$, as presented in the l_1 Support Vector Machine (l_1-SVM) approach of Bradley and Mangasarian [9]:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \sum_{i=1}^{n} |w_i| + C \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y_i \cdot (\mathbf{w}^\top \cdot \mathbf{x}_i + b) \geqslant 1 - \xi_i, \qquad i = 1, \ldots, m, \\ & \xi_i \geqslant 0, \qquad i = 1, \ldots, m. \end{aligned} \tag{10}$$

An alternative sparsity term is the minimization of the "zero norm" (l_0 norm): $\Omega(\mathbf{w}) = \|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|$. Note that, unlike the l_1 norm, $\|\cdot\|_0$ is not a norm because the triangle inequality does not

hold [9]. Since the l_0 norm is non-smooth, Bradley and Mangasarian [9] proposed the following concave approximation:

$$\|\mathbf{w}\|_0 \approx \mathbf{1}_n{}^{\top}(\mathbf{1}_n - \exp(-\beta|\mathbf{w}|), \tag{11}$$

where β controls the steepness of the penalty function. The l_0 norm can be considered instead of the Euclidean norm, as suggested by Bradley and Mangasarian [9] in their approach, Feature Selection ConcaVe (FSV). Since this function is not directly differentiable, the authors suggest an iterative approach based on a constrained gradient descent method. The sparsity terms can also be combined with the Euclidean norm to achieve both generalization and sparsity simultaneously. Weston [38] proposed an alternative approach for l_0 norm minimization (l_0-SVM), which scales the variables iteratively by the absolute value of the weight vector $\mathbf{w}$, and obtains from the primal SVM formulation (1), until convergence. A ranking of variables can be constructed by removing the features whose weights become zero during the iterative algorithm and computing the order of removal. This method considers the following approximation of the l_0 norm:

$$\Omega(\mathbf{w}) = \sum_{j=1}^{n} \log(\epsilon + |w_i|). \tag{12}$$

To the best of our knowledge, only one feature selection approach based on the SOCP formulation for SVM has been proposed to date. This work extends the ideas of l_1 Support Vector Machine (l_1-SVM) to second-order cones, by minimizing the l_1 norm [7] instead of the Euclidean norm used in formulation (5):

$$\begin{aligned} \min_{\mathbf{u},\mathbf{v},b} \quad & (\mathbf{u}+\mathbf{v})^{\top} \cdot \mathbf{1}_n \\ \text{s.t.} \quad & (\mathbf{u}-\mathbf{v})^{\top} \cdot \boldsymbol{\mu}_1 - b \geqslant 1 + \kappa_1 \|S_1^{\top}(\mathbf{u}-\mathbf{v})\|, \\ & b - (\mathbf{u}-\mathbf{v})^{\top} \cdot \boldsymbol{\mu}_2 \geqslant 1 + \kappa_2 \|S_2^{\top}(\mathbf{u}-\mathbf{v})\|, \\ & \mathbf{u} \geqslant 0,\ \mathbf{v} \geqslant 0. \end{aligned} \tag{13}$$

Formulation (13) addresses the l_1 norm by introducing two vectors $\mathbf{u}$ and $\mathbf{v}$, leading to a convex linear problem with second-order cone constraints.

An important drawback of these methods is that they are limited to linear classifiers [16,25]. Some approaches have been proposed for nonlinear feature selection via model optimization. For example, Maldonado et al. [26] proposed an SVM-based approach based on the dual version of SVMs (Formulation (2)) that penalizes the number of features using an anisotropic Gaussian kernel and a concave penalization of the zero norm for the scaling factors. An alternative approach presented by Chapelle et al. [13] automatically tunes the scaling factors in the dual formulation of Support Vector Machines using a gradient descent algorithm. In the following section we incorporate some of these strategies to propose a backward elimination algorithm for the robust version of SVMs, which can be extended to nonlinear classifiers.

Feature selection strategies can be categorized into supervised, unsupervised, or semi-supervised algorithms depending on the use of the respective label information [1]. Only few approaches have been proposed for unsupervised learning. This is a challenging task because methods that attempt to detect

irrelevant attributes by assessing the correlation between variables and labels are not suitable for clustering since such class labels are not available. One such method was presented by Dyer et al. [15] who provide results for feature selection in subspace clustering. Another family of unsupervised feature selection methods is based on spectral graph theory [14]. Such methods focus on a *target concept* rather than on class labels. The idea behind spectral feature selection is that observations that are close to each other in the feature space should belong to the same target concept [41]. Some approaches that follow this principle are SPEC [41], Laplacian Score [19], and MCFS [11].

4. Feature ranking approaches based on SOCP-SVM

In this section we propose a backward feature selection method for SVM based on the formulation with moments and using second-order cone programming. The approach starts with all available features and determines each feature's contribution to the respective classifier for ranking the features in terms of relevance. The reasoning behind this method is that we can improve classification performance by eliminating the features that adversely affect the generalization capacity of the classifier. Our strategy considers a robust model that makes no assumptions about the distribution of features, controlling misclassification probabilities in a worst-case setting, i.e., under all possible choices of class-conditional densities. The robustness of the SOCP classifier should minimize the risk of removing potentially relevant attributes during the backward elimination process, which is common in greedy approaches [16].

We first introduce the linear case, in which we construct a linear classifier using the SOCP formulation for SVM and perform backward selection based on the weights obtained by the model. Subsequently, a non-linear approach that considers Kernel functions is presented.

4.1. Feature ranking for linear SOCP-SVM

According to the notation used by Song et al. [32], $\mathcal{S}$ denotes the full set of features, and a backward algorithm generates a relevance rank of attributes $\mathcal{S}^{\dagger}$. At each iteration $\mathcal{S}^{\dagger}$ is appended by one or several features from $\mathcal{S}$ which are not yet contained in $\mathcal{S}^{\dagger}$ by selecting those features which are least important in the construction of the hyperplane, given by components of the weight vector obtained from the SOCP-SVM formulation. We propose the following backward algorithm (SOCPSVM-BFE$_l$, second-order cone programming SVM – Backward Feature Elimination, linear case):

Algorithm 1 SOCPSVM Backward Feature Elimination – linear case

Input: The full set of features $\mathcal{S}$
Output: An ordered set of features $\mathcal{S}^{\dagger}$

1. $\mathcal{S}^{\dagger} \leftarrow \emptyset$
2. **repeat**
3. $\quad \mathbf{w} \leftarrow$ SOCP-SVM Training (Formulation (5)).
4. $\quad \mathcal{I} \leftarrow \operatorname{argmin}_{\mathcal{I}} \sum_{p \in \mathcal{I}} |w_p|$, $\mathcal{I} \subset \mathcal{S}$.
5. $\quad \mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{I}$.
6. $\quad \mathcal{S}^{\dagger} \leftarrow (\mathcal{S}^{\dagger}, \mathcal{I})$.
7. **until** $\mathcal{S} = \emptyset$.

Algorithm 1 is concerned with the selection of a set $\mathcal{I}$ of features to eliminate. As indicated in the previous section, several elements of $\mathcal{S}$ can be removed at each iteration to speed up the training process. We remove half of the features at each step of the algorithm.

4.2. Feature ranking for kernel-based SOCP-SVM

For the nonlinear version of the algorithm, the weight vector is no longer available since it is related to the projection function, ϕ. In order to overcome this issue, we propose the robust version of the contribution measure $W^2(\boldsymbol{\alpha})$ presented in the previous section for the RFE-SVM method Eq. (9). For this approach, the margin is inversely proportional to the Euclidean norm of the weight vector, which can be rewritten using the coefficients $\mathbf{s}$ from Formulation Eq. (7) and the new kernel function. The contribution measure follows:

$$W^2(\mathbf{s}) = \mathbf{s}^\top \mathbf{K}\mathbf{s}. \tag{14}$$

The modified algorithm for kernel-based backward elimination (SOCPSVM-BFE$_{nl}$, second-order cone programming SVM – Backward Feature Elimination, nonlinear case) is presented as Algorithm 2:

Algorithm 2 SOCPSVM Backward Feature Elimination – nonlinear case

Input: The full set of features $\mathcal{S}$
Output: An ordered set of features $\mathcal{S}^\dagger$

1. $\mathcal{S}^\dagger \leftarrow \emptyset$
2. **repeat**
3. $\quad\boldsymbol{\alpha} \leftarrow$ SOCP-SVM Training (Formulation (7)).
4. $\quad\mathcal{I} \leftarrow \text{argmin}_{\mathcal{I}} \sum_{p\in\mathcal{I}} |W^2(\mathbf{s}) - W^2_{(-p)}(\mathbf{s})|,\ \mathcal{I} \subset \mathcal{S}$.
5. $\quad\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{I}$.
6. $\quad\mathcal{S}^\dagger \leftarrow (\mathcal{S}^\dagger, \mathcal{I})$.
7. **until** $\mathcal{S} = \emptyset$

In step 4 of Algorithm 2 we define $W^2_{(-p)}(\mathbf{s}) = \mathbf{s}^\top \mathbf{K}^{(-\mathbf{p})}\mathbf{s}$, which means, we construct the kernel function with attribute p removed from the training vectors $\mathbf{x}$, where $\mathbf{K} = [\mathbf{K}_{11}, \mathbf{K}_{12}; \mathbf{K}_{21}, \mathbf{K}_{22}]$, with $\mathbf{K}_{rs}$ represents the matrix with entry (i, j), $(\mathbf{K}_{rs})_{i,j} = K(\mathbf{x}_i^{r(-p)}, \mathbf{x}_j^{s(-p)})$, for $r, s = \{1, 2\}$ and $\mathbf{x}_i^{(-p)}$ means training object i with feature p removed.

To reduce computational complexity of the proposed algorithm, we use a similar approximation as that described in Guyon et al. [17], in which the vector $\mathbf{s}$ used in $W^2_{(-p)}(\mathbf{s})$ is set equal to the solution of Eq. (7) even if a feature has been removed.

5. Numerical experiments

In this section we briefly describe the benchmark datasets and provide the classification results using different feature selection methods.

5.1. Datasets and experimental settings

We applied the proposed approaches for feature selection and the alternative methods, Fisher Score, l_1-SVM and its extension to SOCP by Bhattacharyya [7] (l_1-SOCP), the method l_0-SVM, and SVM-RFE (for both linear and nonlinear feature selection) to three well-known DNA microarray benchmark datasets, which have already been used for benchmark feature selection algorithms (see, for example, [32,39]):

- *Colorectal Microarray (CoMA)* [4]: The CoMA dataset contains the expression of the 2000 genes with the highest minimal intensity across 62 samples (40 tumor and 22 normal).
- *Lymphoma Microarray (LyMA)* [2]: The LyMA dataset contains the gene expression of 96 samples (61 malignant and 35 normal) described by 4026 features.
- *Lung Microarray (LuMA)* [6]: The LuMA dataset contains the gene expression of 181 samples (31 malignant and 150 normal) described by 12533 features.

The first step of the evaluation is model selection. We compare the results of the best model found using a standard model selection procedure for linear SVM and kernel-based SVM (Gaussian kernel) without feature selection. We evaluate the effectiveness of the learner by its percentage of correct predictions (classification accuracy), and by the arithmetic mean of the true positive rate and the true negative rate which represents the Area Under the Curve (AUC) when only one run is available [31].

For feature selection we follow the procedure presented in Victo Sudha George and Cyril Raj [36]: training and test subsets are obtained using a leave-one-out cross-validation, which is a common procedure for tumor prediction with DNA microarray data [21]. Feature selection and classification are then performed on the training set and the classification performance (accuracy and AUC) is finally computed by averaging the test results. We consider four classifiers: linear SVM, nonlinear SVM with gaussian kernel, linear SOCP-SVM with $C = 1$, and nonlinear SOCP-SVM with gaussian kernel ($C = 1$ and $\sigma = \frac{n}{5}$, with n the number of selected features). The values for these parameters were selected based on a grid search performed for these data sets in previous works (see [22,26]). For the robust approaches we consider that $\eta = \eta_1 = \eta_2$ and we study the following values of $\eta = \{0.2, 0.4, 0.6, 0.8\}$. The values for parameter η are chosen based on the paper Bhattacharyya [7] to provide a fair comparison between our approach and l_1-SOCP. Among feature selection methods, l_1-SVM and l_1-SOCP are studied as both feature selection and classifier, while Fisher Score, l_1-SVM, l_0-SVM, SVM-RFE and the proposed SOCP-BFE for linear and non-linear classification are used as feature ranking. Then the respective classification method is trained for an increasing number of ranked features. We compare our results choosing the number of variables indicated in Rakotomamonjy [29]:

- CoMA: $n = \{10, 20, 50, 100, 250, 500, 1000, 2000\}$.
- LyMA: $n = \{10, 20, 50, 100, 250, 500, 1000, 2000, 4026\}$.
- LuMA: $n = \{10, 20, 50, 100, 250, 500, 1000, 2000, 4000, 12533\}$.

Notice that the last value represents the full set of variables, which is used to illustrate the performance of all methods considering the same starting point. For this procedure we used the Spider Toolbox for Matlab for standard SVM approaches [37] and the SeDuMi Matlab Toolbox for SOCP-based classifiers [33].

5.2. Classification performance summary

Table 1 summarizes the results obtained for each feature selection approach and for the three datasets. From these results we obtained the mean performance for each method by averaging all the different subsets of attributes, and the maximum performance in terms of AUC with the respective number of ranked features is displayed in Table 1. These metrics provide a good comparison point for all methods, balancing accuracy and stability.

Table 1 presents the best performance considering in the first group all the linear alternative approaches based on linear SVM: SVM-RFE$_l$ (recursive feature elimination SVM, linear elimination), l_0-SVM, Fisher Score (for feature ranking) and SVM classification, l_1-SVM for feature ranking (l_1-SVM$_r$) and for automatic feature selection (l_1-SVM$_e$), and the l_1 extension to second order cones (l_1-SOCP). The

Table 1
Performance summary for different feature selection approaches. All datasets

	CoMA dataset			LyMA dataset			LuMA dataset		
	Mean	Max	Feat.	Mean	Max	Feat.	Mean	Max	Feat.
SVM-RFE$_l$	83.1	86.9	10	92.7	95.5	100	96.6	96.7	10
Fisher+SVM	79.7	83.4	50	92.7	95.5	1000	97.8	98.4	20
l_0-SVM	79.4	82.4	100	93.5	95.5	50	96.4	96.7	20
l_1-SVMr	81.7	85.9	10	93.9	95.5	20	96.4	96.7	50
l_1-SVMe	*	85.9	217	*	93.9	62	*	95.0	41
l_1-SOCP	*	71.1	64	*	95.5	89	*	98.4	23
SVM-RFE$_l$	85.2	*90.5*	*250*	93.8	95.5	50	96.7	96.7	10
SVM-RFE$_{nl}$	86.5	90.5	100	93.0	95.5	100	96.6	96.7	20
Fisher+SVM	86.4	86.9	20	93.8	96.2	100	97.6	98.4	100
l_0-SVM	84.7	88.2	250	94.0	*97.0*	*50*	96.7	96.7	20
l_1-SVMr	85.3	89.2	10	93.4	95.5	100	96.7	96.7	20
SOCP-BFE$_l$	*87.2*	89.2	10	93.8	95.5	50	*98.2*	*98.4*	*10*
SOCP-BFE$_{nl}$	*87.2*	89.2	10	*94.3*	97.0	100	98.0	98.4	20

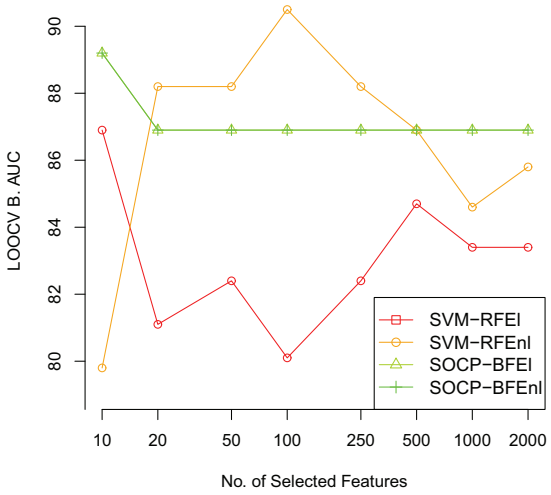
second block includes linear and nonlinear alternative approaches using a kernel-based SVM classifier: SVM-RFE$_l$ and SVM-RFE$_{nl}$ (recursive feature elimination SVM, linear and nonlinear elimination), l_0-SVM, Fisher Score (for feature ranking) and nonlinear SVM classification, and l_1-SVM for feature ranking (l_1-SVM$_r$). Finally, the third block presents our proposal: the backward feature elimination method based on linear SOCP-SVM (SOCP-BFE$_l$) and kernel-based SOCP-SVM (SOCP-BFE$_{nl}$).

Table 1 shows the following results:

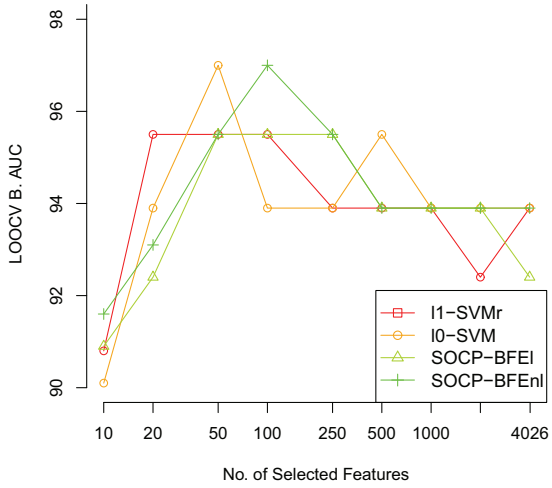
- The best overall performance (based on mean AUC) for all datasets is achieved with our backward elimination methods based on SOCP-SVM. This result confirms the theoretical advantages of the approaches. However, none of the methods based on their maximum performance performs best in all datasets.
- The best overall performance is obtained with the linear version for the LuMA dataset, with the kernel-based version for the LyMA dataset, and for the CoMA dataset both approaches behave similarly. From these results we conclude that there is no significant gain in terms of predictive performance from using the nonlinear version of our method compared to the linear version, although for alternative approaches we observed a clear benefit in using nonlinear SVM as the baseline classifier (second block of methods), compared to linear SVM (first block of methods).
- Embedded methods are very interesting alternatives when the desired number of attributes in the final classifier is unknown and we want to set it automatically. However, for high dimensional datasets, the risk of achieving poor results is higher compared with ranking methods. The l_1-SVM$_e$ classifier achieved great performance for the CoMA dataset, similar to the best performance with ranking methods. On the other hand, l_1-SOCP accomplished very good performance for the LyMA and LuMA datasets, but failed at identifying the relevant features for the CoMA dataset.

Next, a comparison between our approaches and the best linear and nonlinear alternative approaches is presented for each dataset. We consider this approach for visualization purposes, instead of presenting all methods together. The best approaches were selected based on their mean performance. For all robust formulations we consider the best solution for the different values of $\eta = \{0.2, 0.4, 0.6, 0.8\}$. Figure 1 presents a graphic representation of the predictive performance for an increasing number of ranked features for all three datasets.

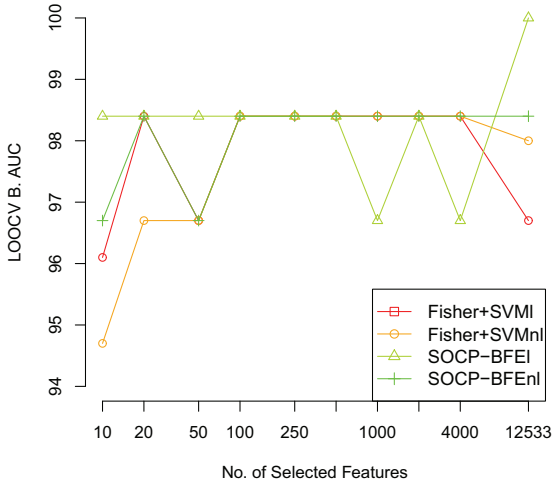
In Fig. 1 we first notice that the best alternative approach differs in all datasets: SVM-RFE$_l$ and SVM-RFE$_{nl}$, l_1-SVM (linear SVM) and l_0-SVM (nonlinear SVM), and Fisher Score with linear and nonlinear



(a) CoMA dataset



(b) LyMA dataset



(c) LuMA dataset

Fig. 1. LOO AUC versus the number of ranked variables for different feature selection approaches. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-150781)

SVM were considered for the CoMA, LyMA and LuMA datasets, respectively. Although the results may seem similar between the proposed and alternative methods and the gains may not appear to be significant, we observed clear differences in performance when comparing each alternative method one by one with our proposals.

In Fig. 1(a) we observe that SVM-RFE$_l$ is outperformed by SVM-RFE$_{nl}$ and our proposals, which behave similarly. While our methods are stable and achieve the best performance with 10 attributes, SVM-RFE$_{nl}$ has its peak with 100 variables and then decreases its performance significatively by 10 attributes. For the LyMA dataset Fig. 1(b), all methods behave relatively similar, improving their performance by reducing the dimensionality to 50–250 features, but then accuracy decreases significantly for 10–20 variables. The method l_0-SVM with a kernel-based classifier and SOCP-BFE$_{nl}$ achieved the best performance. Finally, Fig. 1(c) presents the results for the LuMA dataset, where more stable results

Table 2
Sensitivity for parameter η. ANOVA test for all datasets

	CoMA dataset	LyMA dataset	LuMA dataset
SOCP-BFE$_l$	0.011	0.00	0.07
SOCP-BFE$_{nl}$	0.17	0.00	0.15

Table 3
Sensitivity for parameter η. LyMA dataset

	10	20	50	100	250	500	1000	2000	4000
SOCP-BFE$_{l\ \eta=0.2}$	90	89.3	93.2	91.6	90	88.4	86.7	86.7	86.7
SOCP-BFE$_{l\ \eta=0.4}$	86.7	88.4	90.7	92.4	92.4	90	88.4	90	90
SOCP-BFE$_{l\ \eta=0.6}$	90.9	92.4	95.5	95.5	95.5	93.9	93.9	93.9	92.4
SOCP-BFE$_{l\ \eta=0.8}$	86.9	91.6	95.5	97	95.5	93.9	93.9	93.9	93.9
SOCP-BFE$_{nl\ \eta=0.2}$	89.3	90.1	93.2	91.6	91.6	90	88.4	86.7	88.4
SOCP-BFE$_{nl\ \eta=0.4}$	86.7	88.4	93.9	92.4	93.9	90	88.4	93.2	91.6
SOCP-BFE$_{nl\ \eta=0.6}$	90.9	92.4	96.2	95.5	95.5	93.9	93.9	93.9	93.9
SOCP-BFE$_{nl\ \eta=0.8}$	91.6	93.1	95.5	97	95.5	93.9	93.9	93.9	93.9

were obtained for all methods, but we consider SOCP-BFE$_l$ to be the best approach since it has perfect performance using all attributes, and the best feature selection results with 10 variables.

Regarding computational complexity of our proposal, the SOCP-BFE algorithm is similar to the backward procedure followed by SVM-RFE, whose complexity is of the order of $\max(n, m)m^2 = n * m^2$ considering the operations of SVM (computation of the kernel and its inversion) and the successive backward elimination steps with a decreasing number of variables (assuming a reduction by a factor of 2 at each iteration) [16]. These embedded methods computationally more expensive than filter techniques, but considerably more efficient than wrapper approaches that perform exhaustive search. Our proposal is suitable for large-size microarray datasets, such LuMA, since SOCP-BFE can be trained in less than one minute running time.

5.3. Influence of the hyperparameters and discussion

In this subsection we report the performance of the proposed feature selection methodologies by performing sensitivity analysis of parameter η, characterizing its influence on the final solution. Our goal was to assess whether the results are stable along different values of this parameter. If this is the case, a less rigorous validation strategy can be used. In contrast, high variance in the performance would require more exhaustive model selection in order to find the best combination of parameters.

We monitored the performance of our proposals by varying the different values of η. For these experiments we constructed ANOVA tests to assess the influence of η on each method and dataset. Table 2 presents their respective p values for these ANOVA tests.

In Table 2 we observe that the differences are not statistically significant for the case of the CoMA and LuMA datasets (p values above 0.01), while a stronger influence of parameter η is detected for the LyMA dataset (p values below 0.01). The detailed results for this dataset are presented in Table 3 as illustrative example.

From Table 3 we see better performance for higher values of η (0.6 and 0.8) compared to lower ones (0.2 and 0.4). We conclude that the models are relatively robust in the context of the parameter setting, although it is important to set parameter η using cross-validation and considering the values presented in this work (or a broader range of them) to achieve best results.

6. Conclusions

We presented two backward elimination approaches for feature ranking using the robust SVM formulation with second-order cones. A comparison with other feature selection and classification approaches in high dimensional applications (such as microarray datasets) showed the advantages of the proposed strategies:

- They achieve the best performance compared with other feature ranking techniques in terms of classification accuracy based on their ability to construct robust classifiers by assuming the worst distribution of the data and directly controlling the error rate via the parameter η.
- They result from extending the work of Bhattacharyya [7] on embedded feature selection and classification for SOCP-SVM, and proposing a backward elimination approach that considers the use of the Euclidean norm in the formulation, instead of sacrificing the structural risk minimization principle by replacing it with the l_1 norm.
- Our nonlinear approach allows the use of kernel functions for nonlinear feature selection and classification, giving flexibility to the model construction.
- The strategy presented here performs a ranking of features, and the model selection procedure becomes the assessment of different classification techniques for an increasing number of ranked features. This is in contrast to the construction of a single solution that may lead to poor classification performance.

In Section 5 we propose a strategy for avoiding uneven comparison between linear and nonlinear methods, considering similar efforts in both cases. Although no significant gain is obtained in our experiments by using kernel methods, the proposed strategy has the potential of achieving better results under a more exhaustive model selection, by embedding it in the feature selection process. Given the noteworthy results achieved while combining SOCP-SVM for feature ranking with the same approach for classification, we suggest performing a grid search for parameters C and σ at each iteration of the algorithm, and monitoring the performance in order to obtain a final solution during the process, without considering feature selection and classification as independent problems. According to our results, the success of a nonlinear backward elimination is strongly dependent on the right definition of the hyperplane, and in particular, on the correct setting of the parameter σ for high dimensional applications.

The experimental procedure also allows comparison between the traditional SVM classifiers and its robust versions for both linear and nonlinear cases. According to our results, the robust version of SVM leads to consistently better results in terms of AUC compared to the traditional SVM classifier, resulting in an attractive alternative. The main drawback, however, is the running time of the algorithm under SeDuMI toolbox.

There are several opportunities for future work aimed at improving this strategy. First, there is a pressing need for more efficient implementations of second-order cone programming formulations, and in particular applied to SOCP-SVM. We are currently working on efficient strategies for feature selection and model selection to reduce computational times, but faster implementations are necessary for the method to become a real alternative to traditional SVM. Secondly, given that feature selection can be cast to a non-convex optimization problem, based on the minimization of the "zero norm" (as presented in Section 3), extensions of the l_0 penalization strategy to SOCP-SVM represent an interesting research opportunity. Finally, SOCP-SVM presents interesting properties for classification on highly imbalanced datasets, a very relevant topic in pattern recognition given the vast field of applications [24,34]. Since the parameter η controls the Type I and Type II errors, a differentiated value of this parameter may help to construct better classification functions that consider the costs of both types of errors [28]. Subsequently, wrapper or embedded feature selection can be performed using these classifiers, selecting those attributes that are relevant for identifying rare cases from the target class [24,34].

Acknowledgements

The authors would like to thank Richard Weber and Richard LeBoeuf for their valuable comments and suggestions. The first author was supported by FONDECYT project 11121196, while the second author was supported by FONDECYT project 11110188 and by CONICYT, under Anillo project ACT1106. The work reported in this paper has been partially funded by the Complex Engineering Systems Institute (ICM: P-05-004-F, CONICYT: FB016).

References

[1] S. Alelyani, J. Tang and H. Liu, *Data Clustering: Algorithms and Applications*, chapter Feature Selection for Clustering: A Review, CRC Press, 2013.
[2] A. Alizadeh, M. Eisen, R. Davis et al., Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling, *Nature* **403** (2000), 503–511.
[3] F. Alizadeh and D. Goldfarb, Second-order cone programming, *Mathematical Programming* **95** (2003), 3–51.
[4] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack and A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligo-nucleotide arrays, in: *Proceedings of the National Academy of Sciences* (1999), 6745–6750.
[5] F. Alvarez, J. López and H.C. Ramírez, Interior proximal algorithm with variable metric for second-order cone programming: applications to structural optimization and Support Vector Machines, *Optimization Methods Software* **25**(6) (2010), 859–881.
[6] D.G. Beer, Kardia, S.L. Huang et al., Gene-expression profiles predict survival of patients with lung adenocarcinoma, *Nature Medicine* **8** (2002), 816–824.
[7] C. Bhattacharyya, Second order cone programming formulations for feature selection, *Journal of Machine Learning Research* **5** (2004), 1417–1433.
[8] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J.M. Benítez and F. Herrera, A review of microarray datasets and applied feature selection methods, *Information Sciences* **282**(0) (2014), 111–135.
[9] P. Bradley and O. Mangasarian, Feature selection vía concave minimization and support vector machines, in: *Machine Learning proceedings of the fifteenth International Conference (ICML'98), San Francisco, California, Morgan Kaufmann*, (1998), 82–90.
[10] B.D.E.K. Paas, G. Smith-Miles, L.C. Thomas, R. Weber, R. Baeza-Yates, C. Bravo, G. L'Huillier and S. Maldonado, Future trends in business analytics and optimization, *Intelligent Data Analysis* **15**(6) (2011), 1001–1017.
[11] D. Cai, C. Zhang and X. He, Unsupervised feature selection for multi-cluster data, in: *16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'10)* (2010).
[12] E. Carrizosa, B. Martín-Barragán and D. Romero-Morales, Detecting relevant variables and interactions in supervised classification, *European Journal of Operational Research* **213**(1) (2011), 260–269.
[13] O. Chapelle, V. Vapnik, O. Bousquet and S. Mukherjee, Choosing multiple parameters for Support Vector Machines, *Machine Learning* **46** (2002), 131–159.
[14] D.M. Cvetković, M. Doob and H. Sachs, *Spectra of Graphs: Theory and Application*, Pure and applied mathematics, Academic Press, 1980.
[15] E.L. Dyer, A.C. Sankaranarayanan and R.G. Baraniuk, Greedy feature selection for subspace clustering, *Journal of Machine Learning Research* **14** (2013), 2487–2517.
[16] I. Guyon, S. Gunn, M. Nikravesh and L.A. Zadeh, *Feature Extraction, Foundations and Applications*, Springer, Berlin, 2006.
[17] I. Guyon, A. Saffari, G. Dror and G. Cawley, Model selection: Beyond the bayesian frequentist divide, *Journal of Machine Learning Research* **11** (2009), 61–87.
[18] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, Gene selection for cancer classification using Support Vector Machines, *Machine Learning* **46**(1–3) (2002), 389–422.
[19] X. He, D. Cai and P. Niyogi, Laplacian score for feature selection, *In NIPS*, MIT Press, 2005.
[20] J. Kittler, *Pattern Recognition and Signal Processing*, chapter Feature Set Search Algorithms, Sijthoff and Noordhoff, Netherlands, 1978, pp. 41–60.
[21] T.C. Lin, R.S. Liu, C.Y. Chen, Y.T. Chao and S.Y. Chen, Pattern classification in DNA microarray data of multiple tumor types, *Pattern Recognition* **39**(12) (2006), 2426–2438.
[22] S. Maldonado, F. Famili and R. Weber, Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines, *Information Sciences* **286** (2014).

[23] S. Maldonado and J. López, Imbalanced data classification using second-order cone programming Support Vector Machines, *Pattern Recognition* **47** (2014), 2070–2079.
[24] S. Maldonado and C. Montecinos, Robust classification of imbalanced data using ensembles of one-class and two-class svms, *Intelligent Data Analysis Special Issue on Business Analytics and Intelligent Optimization* **18**(1) (2014), 95–112.
[25] S. Maldonado and R. Weber, A wrapper method for feature selection using Support Vector Machines, *Information Sciences* **179** (2009), 2208–2217.
[26] S. Maldonado, R. Weber and J. Basak, Kernel-penalized SVM for feature selection, *Information Sciences* **181**(1) (2011), 115–128.
[27] S. Nath and C. Bhattacharyya, Maximum margin classifiers with specified false positive and false negative error rates, in: *Proceedings of the SIAM International Conference on Data Mining* (2007).
[28] Y. Qu, L.G. Su and J. Chu, A novel svm modeling approach for highly imbalanced and overlapping classification, *Intelligent Data Analysis* **15** (2011), 319–341.
[29] A. Rakotomamonjy, Variable selection using SVM-based criteria, *Journal of Machine Learning Research* **3** (2003), 1357–1370.
[30] B. Schölkopf and A.J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, USA, 2002.
[31] M. Sokolova, N. Japkowicz and S. Szpakowicz, Beyond accuracy F-score and roc: A family of discriminant measures for performance evaluation, in: *Advances in Artificial Intelligence* Springer, Berlin Heidelberg, (2006), 1015–1021.
[32] L. Song, A. Smola, A. Gretton, J. Bedo and K. Borgwardt, Feature selection via dependence maximization, *Journal of Machine Learning Research* **13** (2012), 1393–1434.
[33] J.F. Sturm, Using sedumi 102 a matlab toolbox for optimization over symmetric cones, *Optimization Methods and Software* Special issue on Interior Point Methods (CD supplement with **11**(12) (soft), 625–653.
[34] J. Van Hulse, T.M. Khoshgoftaar and A. Napolitano, An exploration of learning when data is noisy and imbalanced, *Intelligent Data Analysis* **15** (2011), 215–236.
[35] V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, 1998.
[36] G. Victo, S. George and V. Cyril Raj, Review on feature selection techniques and the impact of svm for cancer classification using gene expression profile, *International Journal of Computer Science and Engineering Survey* **2**(3) (2011), 16–27.
[37] J. Weston, A. Elisseeff, G. BakIr and F. Sinz, The spider machine learning toolbox, 2005.
[38] 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** (2003), 1439–1461.
[39] J. Weston, S. Mukherjee, O. Chapelle, M. Ponntil, T. Poggio and V. Vapnik, Feature selection for SVMs, in: *Advances in Neural Information Processing Systems 13* **13** (2001).
[40] M. Zaffalon and M. Hutter, Robust feature selection using distributions of mutual information, in: *A Darwiche and N Friedman Editors Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence (UAI-2002)* San Francisco CA Morgan Kaufmann (2002), 577–584.
[41] Z. Zhao and H. Liu, Spectral feature selection for supervised and unsupervised learning, in: *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, New York, NY, ACM, (2007), 1151–1157.