Information Sciences 268 (2014) 328–341

Contents lists available at ScienceDirect

Information Sciences

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

Alternative second-order cone programming formulations for support vector classification

Sebastián Maldonado [a,*], Julio López [b]

[a] Universidad de los Andes, Mons. Álvaro del Portillo 12455, Las Condes, Santiago, Chile
[b] Facultad de Ingeniería, Universidad Diego Portales, Ejército 441, Santiago, Chile

ARTICLE INFO

Article history:
Received 29 May 2013
Received in revised form 31 October 2013
Accepted 26 January 2014
Available online 3 February 2014

Keywords:
Support Vector Machine
Second-order cone programming
Linear programming SVM

ABSTRACT

This paper presents two novel second-order cone programming (SOCP) formulations that determine a linear predictor using Support Vector Machines (SVMs). Inspired by the soft-margin SVM formulation, our first approach (ξ-SOCP-SVM) proposes a relaxation of the conic constraints via a slack variable, penalizing it in the objective function. The second formulation (r-SOCP-SVM) is based on the LP-SVM formulation principle: the bound of the VC dimension is loosened properly using the l_∞-norm, and the margin is directly maximized. The proposed methods have several advantages: The first approach constructs a flexible classifier, extending the benefits of the soft-margin SVM formulation to second-order cones. The second method obtains comparable results to the SOCP-SVM formulation with less computational effort, since one conic restriction is eliminated. Experiments on well-known benchmark datasets from the UCI Repository demonstrate that our approach accomplishes the best classification performance compared to the traditional SOCP-SVM formulation, LP-SVM, and to standard linear SVM.

1. Introduction

Binary classification is one of the most important data mining tasks since the research topics and application domains are vast, including business analytics [7] and credit scoring [23], computer vision [28], medical diagnosis [21], and document classification [27], to name a few. Among existing classification methods, Support Vector Machines (SVMs) [24] provide theoretical advantages, such as adequate generalization to new objects, thanks to the Structural Risk Minimization (SRM) principle [25], absence of local minima via convex optimization, and representation that depends on only a few parameters. These advantages usually lead to better empirical results compared to other statistical and machine learning approaches [12,14].

Recently, second-order cone programming (SOCP) formulations have been proposed as an alternative optimization scheme for SVMs. These consider all possible choices of class-conditional densities with a given mean and covariance matrix, i.e, in a worst-case setting. They therefore, avoid making assumptions about the class-conditional densities, which would cast the generality and validity of such approach in doubt [5,8]. Moreover, these formulations provide a cost-sensitive framework to handle uneven misclassification costs in binary classification intuitively [15], for instance, in the case of medical diagnosis. These special types of non-linear convex optimization problems can be solved efficiently by interior point algorithms [3,5].

* Corresponding author. Tel.: +56 2 26181874.
E-mail addresses: smaldonado@uandes.cl (S. Maldonado), julio.lopez@udp.cl (J. López).

http://dx.doi.org/10.1016/j.ins.2014.01.041
0020-0255/

In this work, two novel SOCP-based methods are introduced for binary classification. The first method controls the complexity of the classifier by introducing slack variables related to the conic constraints, while the second one maximizes the margin directly by replacing the l_2 norm by a decision variable, r, extending the ideas of the LP-SVM method [29].

This paper is organized as follows; in Section 2, we briefly introduce Support Vector Machines and SOCP-SVM for binary classification. Section 3 presents the proposed feature selection method based on SVM. Experimental results using benchmark data sets are given in Section 4. A summary of this paper can be found in Section 5, where we provide its main conclusions and address future developments.

2. Support Vector Machines for binary classification

In this section we describe the mathematical derivation of SVM developed by Vapnik [24], considering differing extensions that are relevant for this work. We consider the simplest case first, a linear classifier for a linearly separable problem, which leads to the hard-margin formulation. Then, we study linear classifiers for linearly non-separable problems [9]. Subsequently, the LP-SVM formulation [29] is described. Finally, we present the variation of SVM based on second-order cone programming [15,18].

2.1. Hard margin SVM

For the linearly separable case, the SVM determines the optimal hyperplane that separates the convex hulls of both training patterns. Given a set of instances with their respective labels $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \Re^n, i = 1,\ldots,m$ and $y_i \in \{-1,+1\}$, the hard-margin SVM aims at finding a classifier of the form $f(\mathbf{x}) = \mathbf{w}^\top \cdot \mathbf{x} + b$ that maximizes the distance from it to the nearest training point on each class (the margin). To maximize this measure, the SVM minimizes the Euclidean norm of coefficients $\mathbf{w}$ [24]. Additionally, we intend to classify the training vectors $\mathbf{x}_i$ correctly into two different classes y_i:

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

2.2. Soft margin SVM

Notice that if there is no hyperplane that can split both classes, formulation (1) becomes unfeasible. Cortes and Vapnik [9] suggested a modified formulation that allows misclassification by balancing the structural risk (minimization of the Euclidean norm), and the empirical risk (minimization of misclassification errors) by introducing slack variables $\xi_i, i = 1,\ldots,m$, which measure the degree of misclassification for an instance $\mathbf{x}_i$, and a penalty parameter C, which controls this trade-off. For a linear penalty function, the *soft margin* SVM formulation becomes:

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

One important advantage of formulation (2) is that these slack variables do not appear in the dual formulation of the problem, resulting only in an additional constraint on the Lagrange multipliers, upper bounding them with the parameter C.

Formulation (2) can be solved efficiently in the dual space using the Sequential Minimal Optimization (SMO) technique [17], among others. Some studies have also been proposed for solving the primal SVM formulation efficiently, using a Newton-based algorithm [13] or back-fitting strategies [16], for example.

2.3. Linear Programming SVM

The method LP-SVM attempts to improve training times by loosening the bound of the VC dimension using the l_∞-norm, resulting in a linear programming formulation that controls the margin maximization directly, by considering a margin variable r [29]. This variable is then maximized while assuring that each instance is on the right side of the hyperplane, and is at least at a distance r from it. The LP-SVM hard-margin formulation for linearly separable problems follows:

$$\begin{aligned} \min_{\mathbf{w},r,b} \quad & -r \\ \text{s.t.} \quad & y_i \cdot (\mathbf{w}^\top \cdot \mathbf{x}_i + b) \geqslant r, \quad i = 1,\ldots,m, \\ & -1 \leqslant w_j \leqslant 1, \qquad j = 1,\ldots,n, \\ & r \geqslant 0. \end{aligned} \tag{3}$$

Similar to the standard SVMs, a soft-margin strategy has been proposed for non-separable cases, where a set of slack variables is introduced and penalized in the objective function:

$$\begin{aligned} \min_{\mathbf{w},r,b,\xi} \quad & -r + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i \cdot (\mathbf{w}^\top \cdot \mathbf{x}_i + b) \geqslant r - \xi_i, \quad i = 1,\ldots,m, \\ & -1 \leqslant w_j \leqslant 1, \qquad j = 1,\ldots,n. \\ & \xi_i \geqslant 0, \qquad i = 1,\ldots,m. \\ & r \geqslant 0, \end{aligned} \tag{4}$$

where C is a positive hyperparameter that can be calibrated using cross-validation. The decision function of LP-SVM is also similar to that of the standard SVMs. The approach was tested on simulated and real data sets in Zhou et al. [29], leading to an improvement of at least an order magnitude in the training speed and making it especially suitable for complex machine learning tasks, such as large scale problems or feature selection. Even if the VC dimension of LP-SVM is larger than l_2-SVM, the generalization error obtained by the authors was smaller than when using SVMs in most cases, from which is was concluded that the loss, in terms of structural risk, is tolerable. The dual formulation of (3) can be stated as follows: (see Appendix A for details)

$$\begin{aligned} \min_{\mathbf{z}_1,\mathbf{z}_2} \quad & \|A\mathbf{z}_1 - B\mathbf{z}_2\|_1 \\ \text{s.t.} \quad & \mathbf{e}^\top\mathbf{z}_1 = 1, \ \mathbf{e}^\top\mathbf{z}_2 = 1, \\ & \mathbf{z}_1 \geq 0, \mathbf{z}_2 \geq 0. \end{aligned} \tag{5}$$

The dual problem (5) has an interesting geometrical interpretation, since it attempts to find the closest points of the training patterns using the 1-norm.

2.4. Second order cone programming SVM

The SOCP-SVM formulation provides a robust and efficient framework for classification since it considers all possible choices of class-conditional densities with a given mean and covariance matrix, achieving great predictive results under different conditions of the data sets [8,18]. Let $\mathbf{X}_1$ and $\mathbf{X}_2$ be random vectors that generate the samples of the positive and negative classes respectively, with means and covariance matrices given by (μ_i, Σ_i) for $i = 1, 2$, where $\Sigma_i \in \Re^{n\times n}$ are symmetric positive semidefinite matrices.

In order to construct a maximum margin linear classifier, such that the probability of false-negative and false-positive errors does not exceed $\eta_1 \in (0, 1]$ and $\eta_2 \in (0, 1]$ respectively, Nath and Bhattacharyya [15] suggested considering the following quadratic chance-constrained programming problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \Pr\{\mathbf{w}^\top \cdot \mathbf{X}_1 - b \geq 0\} \geqslant \eta_1, \\ & \Pr\{\mathbf{w}^\top \cdot \mathbf{X}_2 - b \leq 0\} \geqslant \eta_2. \end{aligned} \tag{6}$$

In other words, the model requires that the random variable $\mathbf{X}_i$ lies on the correct side of the hyperplane, with a probability greater than η_i for $i = 1, 2$. In this case, we want to be able to classify each training pattern $i = 1, 2$ correctly, up to the rate η_i, even for the *worst data distribution*, considering mean and covariance $\mathbf{X}_i \sim (\mu_i, \Sigma_i)$. For this purpose, the probability constraints in (6) are replaced with their *robust* counterparts:

$$\inf_{\mathbf{X}_1\sim(\mu_1,\Sigma_1)} \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_1 - b \geq 0\} \geqslant \eta_1, \ \inf_{\mathbf{X}_2\sim(\mu_2,\Sigma_2)} \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_2 - b \leq 0\} \geqslant \eta_2.$$

Thanks to an appropriate application of the multivariate Chebyshev inequality [10, Lemma 1], this worst distribution approach leads to the following deterministic problem

$$\begin{aligned} & \min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \\ & \text{s.t. } \mathbf{w}^\top \cdot \boldsymbol{\mu}_1 - b \geq 1 + \kappa_1\sqrt{\mathbf{w}^\top \cdot \Sigma_1\mathbf{w}}, \\ & b - \mathbf{w}^\top \cdot \boldsymbol{\mu}_2 \geq 1 + \kappa_2\sqrt{\mathbf{w}^\top \cdot \Sigma_2\mathbf{w}}, \end{aligned} \tag{7}$$

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

By introducing a new variable t and a constraint $\|\mathbf{w}\| \leqslant t$, Formulation (7) can be cast as the following problem

$$\begin{aligned} \min_{\mathbf{w},b,t} \quad & t \\ \text{s.t.} \quad & \|\mathbf{w}\| \leqslant t \\ & \mathbf{w}^\top \cdot \boldsymbol{\mu}_1 - b \geq 1 + \kappa_1\|S_1^\top\mathbf{w}\|, \\ & b - \mathbf{w}^\top \cdot \boldsymbol{\mu}_2 \geq 1 + \kappa_2\|S_2^\top\mathbf{w}\|, \end{aligned}$$

where $\Sigma_i = S_i S_i^\top$, for $i = 1, 2$. This new problem is a convex formulation with a linear objective function and three second-order cone (SOC) constraints [3]. An SOC constraint on the variable $\mathbf{x} \in \Re^n$ is of the form

$$\|A\mathbf{x} + b\| \leqslant c^\top \cdot \mathbf{x} + d,$$

where $d \in \Re, c \in \Re^n, b \in \Re^m, A \in \Re^{m \times n}$ are given.

3. Alternative SOCP-based approaches for support vector classification

In this section, we present two novel approaches for binary classification using second-order cones and describing their relationship with standard SVM and SOCP-SVM. A comparison between these approaches is presented in the next section.

3.1. ξ-SOCP Support Vector Machine

This formulation extends the ideas of the soft-margin SVM approach [9] for training data that are not linearly separable (cf. (2)) to SOCP-SVM. The main idea is to provide a relaxation of the conic constraints by including a slack variable, penalizing it in the objective function. The structural risk is then controlled by the trade-off between the Euclidean norm minimization and a margin variable ξ, as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 + C\xi \\ \text{s.t.} \quad & \inf_{\mathbf{x}_1 \sim (\mu_1, \Sigma_1)} \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_1 \geqslant b - \xi\} \geqslant \eta_1, \\ & \inf_{\mathbf{x}_2 \sim (\mu_2, \Sigma_2)} \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_2 \leqslant b + \xi\} \geqslant \eta_2, \\ & \xi \geq 0, \end{aligned}$$

where $C > 0$ is a sufficiently large penalty parameter, which can be calibrated in the same form as soft-margin SVM (using cross-validation for instance). The parameters η_1, η_2 are also similar to the SOCP-SVM formulation presented in previous sections, taking values in $(0, 1)$.

Again, thanks to an appropriate application of the multivariate Chebyshev inequality, the above problem can now be stated as the following quadratic SOCP problem:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \\ \text{s.t. } & \mathbf{w}^\top \cdot \mu_1 - b \geq 1 - \xi + \kappa_1 \|S_1^\top \mathbf{w}\|, \\ & b - \mathbf{w}^\top \cdot \mu_2 \geq 1 - \xi + \kappa_2 \|S_2^\top \mathbf{w}\|, \\ & \xi \geqslant 0, \end{aligned} \tag{8}$$

where $\Sigma_i = S_i S_i^\top$ and $\kappa_i = \sqrt{\frac{\eta_i}{1-\eta_i}}$ for $i = 1, 2$. This problem will be called the ξ-SOCP-SVM formulation.

By introducing a new variable t and a constraint $\frac{1}{2}\|\mathbf{w}\|^2 \leqslant t$, Formulation (8) can be cast as the following linear SOCP problem

$$\begin{aligned} \min_{\mathbf{w},b,\xi,t} \ & t + C\xi \\ \text{s.t.} \ & \left\| \begin{pmatrix} \sqrt{2}\mathbf{w} \\ 1 - t \end{pmatrix} \right\| \leqslant 1 + t, \\ & \mathbf{w}^\top \cdot \mu_1 - b \geq 1 - \xi + \kappa_1 \|S_1^\top \mathbf{w}\|, \\ & b - \mathbf{w}^\top \cdot \mu_2 \geq 1 - \xi + \kappa_2 \|S_2^\top \mathbf{w}\|, \\ & \xi \geqslant 0. \end{aligned} \tag{9}$$

Applying the KKT conditions to the Lagrangian of Problem (8) (see Appendix B for details), we obtain the following dual formulation results for ξ-SOCP-SVM:

$$\begin{aligned} \min_{\mathbf{z}_1,\mathbf{z}_2} \ & \frac{1}{2}\|\mathbf{z}_1 - \mathbf{z}_2\|^2 \\ \text{s.t.} \ & \mathbf{z}_i \in \mathbf{B}_i(\mu_i, S_i, \kappa_i), \ i = 1, 2, \\ & \|\mathbf{z}_1 - \mathbf{z}_2\| \geqslant \frac{2}{\sqrt{C}}, \end{aligned} \tag{10}$$

where

$$\mathbf{B}_i(\mu_i, S_i, \kappa_i) = \{\mathbf{z}_i : \mathbf{z}_i = \mu_i - (-1)^i \kappa_i S_i \mathbf{u}_i, \|\mathbf{u}_i\| \leq 1\}, \ i = 1, 2.$$

The sets $\mathbf{B}_i(\mu_i, S_i, \kappa_i)$ are ellipsoids centered at μ_i whose shape is determined by S_i and size by κ_i. Thus, the dual problem (10) can be seen as finding the minimum distance between two ellipsoids under the constraint $\|\mathbf{z}_1 - \mathbf{z}_2\| \geqslant \frac{2}{\sqrt{C}}$. Note that, if C is too small, the dual problem will not be feasible. If the ellipsoids are intersecting, and if C is too large, then we obtain $\mathbf{w} = 0$. This follows since $\mathbf{w} = t(\mathbf{z}_1 - \mathbf{z}_2)$ (cf. (19) in Appendix B).

In Fig. 1, we illustrate the points in the ellipsoids (in 2D) obtained by using the formulation (10) for $C = 2^{-4}$(blue hyperplane), $C = 2^{-1}$(red hyperplane), and $C = 2^{2}$(violet hyperplane).

3.2. r-SOCP Support Vector Machine

The reasoning behind the r-SOCP-SVM is that we can control the complexity of the approach (and therefore reduce the Structural risk) by maximizing a margin variable r directly, eliminating the Euclidean norm of the coefficients from the formulation. This margin variable is essentially the same as ξ in the ξ-SOCP formulation, resulting in an extension of the LP-SVM formulation presented in Section 2.3 to second-order cone programming.

The new formulation transforms a quadratic problem with conic restrictions (QSOCP) to a linear one with conic restrictions (SOCP), reducing the computational complexity of the approach. For this, we consider the following linear chance-constrained programming problem:

$$\begin{aligned} \min_{\mathbf{w},b,r} \ & -r \\ \text{s.t.} \ & \inf_{\mathbf{x}_1 \sim (\mu_1,\Sigma_1)} \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_1 \geqslant b + r\} \geqslant \eta_1, \\ & \inf_{\mathbf{x}_2 \sim (\mu_2,\Sigma_2)} \text{Prob}\{\mathbf{w}^\top \cdot \mathbf{X}_2 \leqslant b - r\} \geqslant \eta_2, \\ & -1 \leqslant w_j \leqslant 1, \qquad j = 1, \ldots, n, \\ & r \geqslant 0. \end{aligned} \tag{11}$$

The previous formulation can be cast into the following linear SOCP problem:

$$\begin{aligned} \min_{\mathbf{w},b,r} \ & -r \\ \text{s.t.} \ & \mathbf{w}^\top \cdot \mu_1 - b \geqslant r + \kappa_1 \|S_1^\top \mathbf{w}\|, \\ & b - \mathbf{w}^\top \cdot \mu_2 \geqslant r + \kappa_2 \|S_2^\top \mathbf{w}\|, \\ & -1 \leqslant w_j \leqslant 1, \qquad j = 1, \ldots, n, \\ & r \geqslant 0, \end{aligned} \tag{12}$$

where $\Sigma_i = S_i S_i^\top$ and $\kappa_i = \sqrt{\frac{\eta_i}{1-\eta_i}}$ for $i = 1, 2$.

We referred to the above formulation as r-SOCP-SVM. This approach has the advantage of needing only two conic constraints for classification instead of three, as in the case of standard SOCP-SVM or ξ-SOCP-SVM. Additionally, no tradeoff parameter C is needed, reducing calibration times. The dual formulation for r-SOCP-SVM can be obtained (see Appendix C for derivation), leading to the following problem:

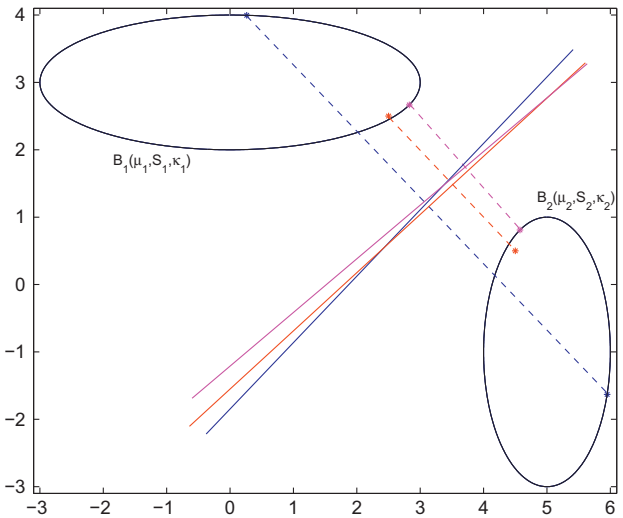

Fig. 1. Geometric interpretation of the formulation (10) for different values of C and its optimal hyperplane.

$$\min_{\mathbf{z}_1,\mathbf{z}_2} \frac{1}{2}\|\mathbf{z}_1 - \mathbf{z}_2\|_1 \quad \text{s.t.} \quad \mathbf{z}_i \in \mathbf{B}_i(\mu_i, S_i, \kappa_i),\ i = 1,2. \tag{13}$$

Thus, the dual formulation minimizes the distance between two ellipsoids using the 1-norm. Fig. 2 shows a comparison between SOCP-SVM, based on the 2-norm (blue hyperplane), and *r*-SOCP-SVM, based on the 1-norm (red hyperplane).

4. Experimental results

We applied the proposed and alternative approaches to six well-known benchmark data sets from the UCI Repository [6]. These sets have already been used for benchmarks in Support Vector Machines (see, for example, Ali and Smith-Miles [1] and Song et al. [20]).

We provide a description of all the data sets in Section 4.1, while Section 4.2 presents a summary of the performance obtained for all the proposed and alternative approaches. Finally, an empirical study regarding the influence of the different parameters and an extended discussion of the results is presented in Section 4.3.

4.1. Description of data sets and validation procedure

A brief description of the data sets is presented here. More information on these datasets can be found in the UCI Repository [6].

- **Australian Credit (AUS)**: This data set contains 690 granted loans from an Australian credit company, 383 good and 307 bad payers in terms of repayment, described by 14 variables (6 numerical and 8 categorical attributes). All attribute information has been modified to protect confidentiality of the data.
- **Wisconsin Breast Cancer (WBC)**: This data set contains 569 observations of tissue samples (212 diagnosed as malignant and 357 diagnosed as benign tumors) described by 30 continuous features, 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, such as the perimeter, the area, the symmetry, and the number of concave portions of the contour.
- **Pima Indians Diabetes (DIA)**: The Pima Indians Diabetes data set presents 8 features and 768 instances (500 tested negative for diabetes and 268 tested positive). All patients are females at least 21 years old of Pima Indian heritage. The features include age, number of times pregnant, diastolic blood pressure and body mass index, among others.
- **German Credit (GC)**: This data set presents 1000 granted loans, 700 good and 300 bad payers in terms of repayment, described by 24 attributes. The variables include loan information (amount, duration, and purpose), credit history, personal information (sex, marital status, number of years in present employment) and other variables to assess financial capacity and willingness to pay (properties, telephone, among others).
- **Ionosphere (ION)**: This radar data presents 351 instances, 225 labeled as *good* radar returns (evidence of some type of structure in the ionosphere) and 126 labeled as *bad* radar returns (no evidence of structure, their signals pass through the ionosphere), described by 34 continuous attributes. The targets were free electrons in the ionosphere.

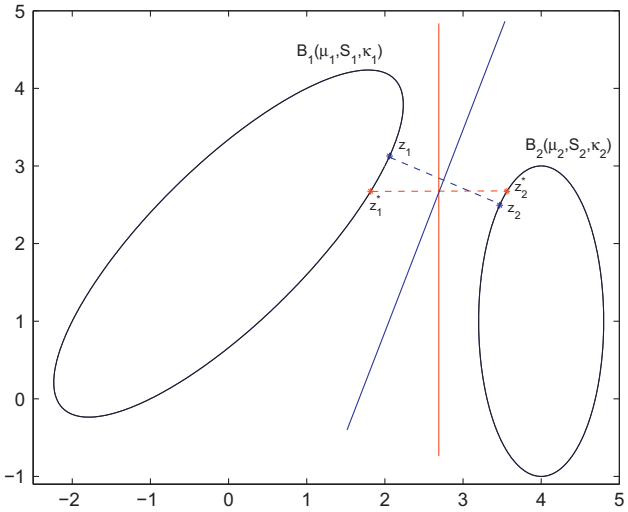

Fig. 2. Geometric interpretation of the distance between two ellipsoids by using the 1-norm (red) and the 2-norm (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- **Splice (SPL)**: This data set contains 1000 randomly selected instances (from the complete set of 3190 splice junctions), in which 517 are labeled as IE (intron–exon) and 483 as EI (exon–intron) borders, described by 60 categorical variables (the gene sequence). Given a DNA sequence, the problem posed in this dataset is recognizing the boundaries between exons and introns (the parts of the sequence retained after splicing and the parts that are spliced out, respectively).
- **Colorectal Microarray (CMA)**: This data set contains the expression of the 2000 genes with the highest minimal intensity across 62 tissues (40 tumor and 22 normal). The genes are placed in order of descending minimal intensity. More information about this data set can be found in Alon et al. [4].
- **Lymphoma Microarray (LMA)**: The lymphoma problem contains the gene expression of 96 samples (61 malignant and 35 normal) described by 4026 features. The problem refers to the diffuse large B-cell lymphoma (DLBCL), the most common subtype of non-Hodgkin's lymphoma. More information about this data set can be found in Alizadeh et al. [2].

Table 1 summarizes the relevant information for each benchmark data set.

Results for the classification approaches SVM in its standard version, LP-SVM, SOCP-SVM (Formulation (7)), and the proposed approaches ξ-SOCP-SVM (Formulation (8)) and r-SOCP-SVM (Formulation (12)) are presented next. The following model selection procedure was performed: training and test subsets were constructed using 10-fold cross-validation for the first six datasets and leave-one-out validation for the microarray data. For this work we studied the metric area under the curve defined by one run, which is widely known as balanced accuracy [19], as the main performance measure. This metric is simply the average between the sensitivity and the specificity.

A grid search is performed to study the influence of the parameters C for soft-margin models and η for SOCP approaches. In this case, we consider $\eta = \eta_1 = \eta_2$ and study the following values of $\eta = \{0.2, 0.4, 0.6, 0.8\}$. We use the following set of values for hyperparameter C:

$$C = \{2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}, 2^{7}\}.$$

For the best C value in terms of AUC we performed a finer grid search around this value, dividing the interval given by the previous and the subsequent point into 10 parts.

For the above procedure, we used the Spider Toolbox for Matlab [26] for standard SVM approaches, and the SeDuMi Matlab Toolbox for SOCP-based classifiers [22].

4.2. Summary of classification results

In Table 2, we present the results for all eight data sets. Table 2 summarizes the best performance (using the AUC measure) of all the approaches along the different values of C and η. (The best method is in bold type).

In Table 2 we observe that the best predictive results were achieved using ξ-SOCP-SVM in seven out of eight datasets, while standard SVM had better AUC in one dataset (Wisconsin Breast Cancer). ξ-SOCP-SVM had the second best performance in this dataset, demonstrating its effectiveness and robustness. The r-SOCP-SVM method is not far from the best value in most cases. Its advantages are due to its simplicity, resulting in faster iterations during the optimization process and avoiding the calibration of an extra parameter C.

4.3. Influence of the parameters and discussion

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

Table 3 summarizes the predictive performance in terms of the AUC for SOCP-SVM, ξ-SOCP-SVM (best performance along the different values of C) and r-SOCP-SVM, respectively. The average, the minimum, and the maximum performance along different values of η are computed and presented in this table. Best results for each dataset are presented in bold.

Table 1
Number of features, number of examples, percentage of each class, and Imbalance Ratio (IR) for all data sets.

Dataset	#Features	#Examples	%Class (min.,maj.)	IR
AUS	14	690	(55.5,44.5)	1.2
WBC	30	569	(62.7,37.3)	1.7
DIA	8	768	(65.1,34.9)	1.9
GEC	24	1000	(70.0,30.0)	2.3
ION	34	351	(64.1,35.9)	1.8
SPL	60	1000	(51.7,48.3)	1.1
CMA	2000	62	(64.5,35.5)	1.8
LMA	4026	96	(63.5,36.5)	1.7

Table 2
Mean AUC, in percentage, for all data sets. Best performance along all parameters.

	AUS	WBC	DIA	GEC	ION	SPL	CMA	LMA
SVM	85.9	**97.4**	70.6	66.6	83.1	80.7	83.4	93.9
LP-SVM	86.3	96.5	73.3	69.4	84.1	80.7	86.9	94.1
SOCP	86.9	96.8	74.2	72.3	83.8	81.3	85.9	94.1
ξ-SOCP	**87.7**	97.3	**75.4**	**72.8**	**84.9**	**81.5**	**90.0**	**94.6**
r-SOCP	86.3	96.5	72.7	72.3	83.5	80.9	86.3	86.3

Table 3
Mean AUC, in percentage, for all data sets. Sensitivity analysis for parameter η.

	AUS	WBC	DIA	GEC	ION	SPL	CMA	LMA
SOCP$_{\eta=0.2}$	86.2	94.3	74.2	72.3	82.7	81.3	85.9	94.1
SOCP$_{\eta=0.4}$	86.2	95.6	71.1	72.0	82.7	81.1	85.9	94.1
SOCP$_{\eta=0.6}$	86.3	96.7	45.9	62.1	83.8	78.3	85.9	94.1
SOCP$_{\eta=0.8}$	86.9	96.8	73.4	65.4	78.3	32.4	85.9	94.1
ξ-SOCP$_{\eta=0.2}$	87.3	94.3	74.3	**72.8**	82.7	81.3	86.3	91.7
ξ-SOCP$_{\eta=0.4}$	**87.7**	95.6	74.3	72.6	82.9	81.1	86.3	94.6
ξ-SOCP$_{\eta=0.6}$	87.2	96.7	74.4	72.6	84.3	**81.5**	88.8	94.6
ξ-SOCP$_{\eta=0.8}$	87.1	**97.3**	**75.4**	71.6	**84.9**	80.8	**90.0**	**94.6**
r-SOCP$_{\eta=0.2}$	83.9	93.6	72.7	72.3	80.6	74.6	85.0	83.9
r-SOCP$_{\eta=0.4}$	85.1	94.4	63.8	70.3	82.6	80.9	86.3	85.1
r-SOCP$_{\eta=0.6}$	86.3	96.5	70.1	69.5	83.5	78.0	86.3	86.3
r-SOCP$_{\eta=0.8}$	78.8	96.1	62.4	55.9	81.9	76.3	79.6	78.9

An important influence of parameter η can be observed in Table 3. The optimal η value varies along the different methods and datasets, and predictive performance is strongly affected by this parameter. Since no clear rule can be defined in order to obtain this value, it is important to set it using cross-validation considering the values presented in this work (or a broader range of them). Nevertheless, the results are relatively stable along all different η values.

In order to study the influence of hyperparameter C in the solution, Figs. 3 and 4 present the predictive performance in terms of AUC for standard SVM, LP-SVM, and ξ-SOCP-SVM, by varying this parameter along the set of different values described earlier. Fig. 3 presents the first four datasets, while Fig. 4 shows the latter four.

Figs. 3 and 4 show similar results for all eight datasets while varying the parameter C. While standard SVM and LP-SVM present very stable results, ξ-SOCP-SVM has very high variance along the different values of C. A finer grid search is performed around the optimal value for this method, and the results are stable in this search.

Although the results obtained with our approach are better in terms of balanced performance in five out of six trials (in the remaining case AUC is slightly low, and the difference is not significant), the parameter C shows a very strong influence in the final outcome of the proposed method, which makes it even more influential than the parameter η. Performing an adequate grid search is highly recommended, varying the parameters C and η along the suggested values in order to obtain the desired results.

The proposed approaches are based on SOCP formulations, which are known to be more time-consuming than linear and quadratic programming, and are therefore, in general, less suitable for machine learning where huge datasets are to be analyzed. Table 4 provides a comparison for one run of the proposed method (one fold using 10-fold cross-validation or leave-one-out in the case of microarray datasets). The mean running time (in seconds) is obtained by averaging all running times for different folds. Additionally, the average number of iterations required for all SOCP approaches is presented in parenthesis.

It is important to notice that all running times are tractable and reasonable. All presented times are relatively similar for all methods, with the approach r-SOCP-SVM being the only exception, since it is significantly slower than the others, especially for the lymphoma microarray data. The reason for this is the number of iterations required for convergence for this approach (presented in parenthesis), which is higher than that for the other SOCP methods. ξ-SOCP-SVM is consistently faster than standard SOCP-SVM, and even faster than traditional SVM in seven out of eight datasets, which seems contradictory given the additional decision variable. The method's ability to find a better initial solution and to reach convergence in fewer iterations makes it not only the most accurate one, but also the fastest among the SOCP methods, although the grid search requires more experiments for an adequate parameter setting. Considering 25 and 4 experiments for the grid search of C and η, respectively, a total of 250 experiments were performed for standard SVM and LP-SVM, 40 experiments for standard SOCP and r-SOCP-SVM, and 1000 experiments for ξ-SOCP-SVM, considering 10-fold cross-validation. The most time-consuming family of experiments is ξ-SOCP-SVM for the LMA dataset, which requires 100 experiments using LOO-cross-validation (96 instances), resulting in approximately $100 \cdot 96 \cdot 1.35 = 3.6$ h. The remaining experiments were always below one hour's duration.

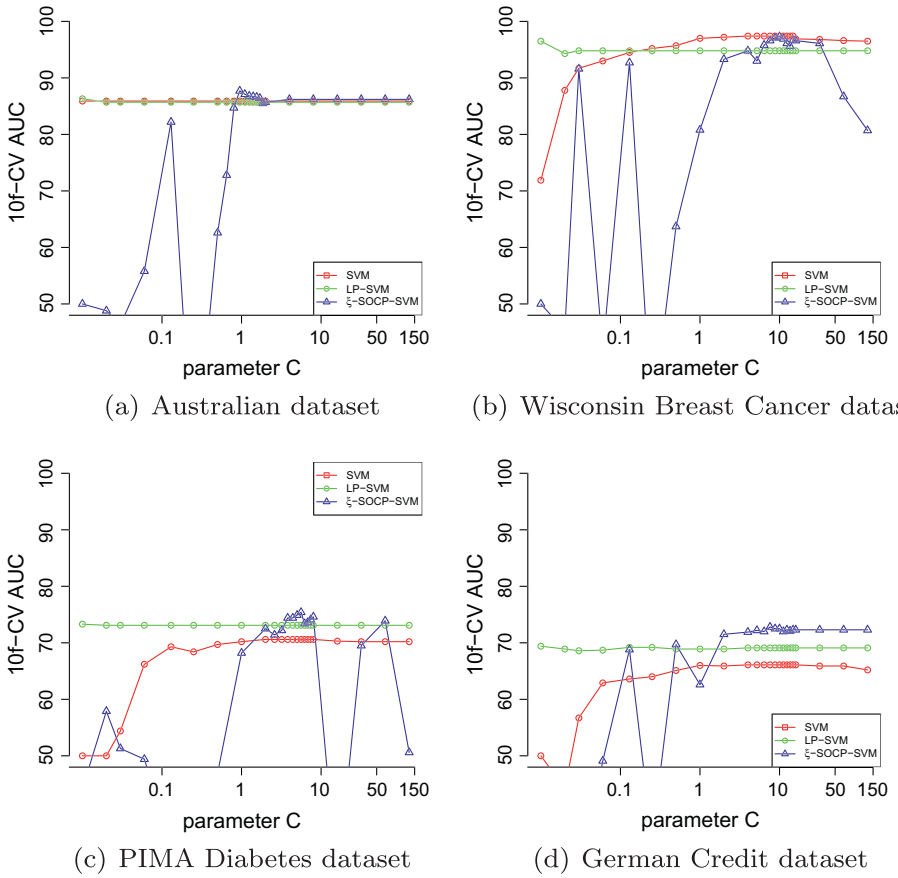

Fig. 3. AUC by varying parameter C for standard SVM, LP-SVM and ξ-SOCP-SVM.

5. Conclusions

In this work, we presented two binary classification approaches based on second-order cone programming and Support Vector Machines. The main idea is to provide alternative SOCP formulations adapting two SVM-based methods, namely soft-margin l_2-SVM and LP-SVM, to improve the well-known SOCP-SVM in terms of predictive performance and computational complexity. While the first proposed approach, ξ-SOCP-SVM, focuses on the first challenge, adjusting better to data by controlling the complexity of the model via a slack variable, the second one, r-SOCP-SVM, focuses on the second objective, reducing the complexity of SOCP-SVM by loosening the bound of the VC dimension. A comparison with other SVM-based classification approaches shows the advantages of the proposed methods:

- The method ξ-SOCP-SVM outperforms other SVM-based classification techniques in terms of predictive performance, based on its ability to generalize better by assuming the worst distribution of the data, and directly controlling the error rate via the parameter η.
- They provide more efficient implementations, leading to a reduction in terms of running times.
- They can be extended to variations of SVM, such as Multi-class SOCP-SVM or kernel-based SOCP-SVM.

Several conclusions can be drawn from the experimental section of this work. Predictive performance (in terms of AUC) is significantly improved with SOCP-SVM classifiers, compared to standard SVM. This result demonstrates the advantage of treating the training patterns as two different (conic) constraints. This is particularly important in imbalanced data sets with uneven error costs, where accuracy becomes an ineffective measure of predictive performance. Additionally, the ξ-SOCP-SVM method achieves better overall performance than all benchmark approaches used in these experiments, achieving best or second best performance for all datasets. By contrast, the r-SOCP-SVM formulation leads to comparable results with reduced computational times and without the need for the calibration of an extra hyperparameter C, which has proved to be necessary in order to achieve the desired results, according to our experiments.

There are several opportunities for future work. These methods can be extended to other machine learning tasks which are beyond the scope of this work but could be handled well by the proposed formulations. For instance, the extension of

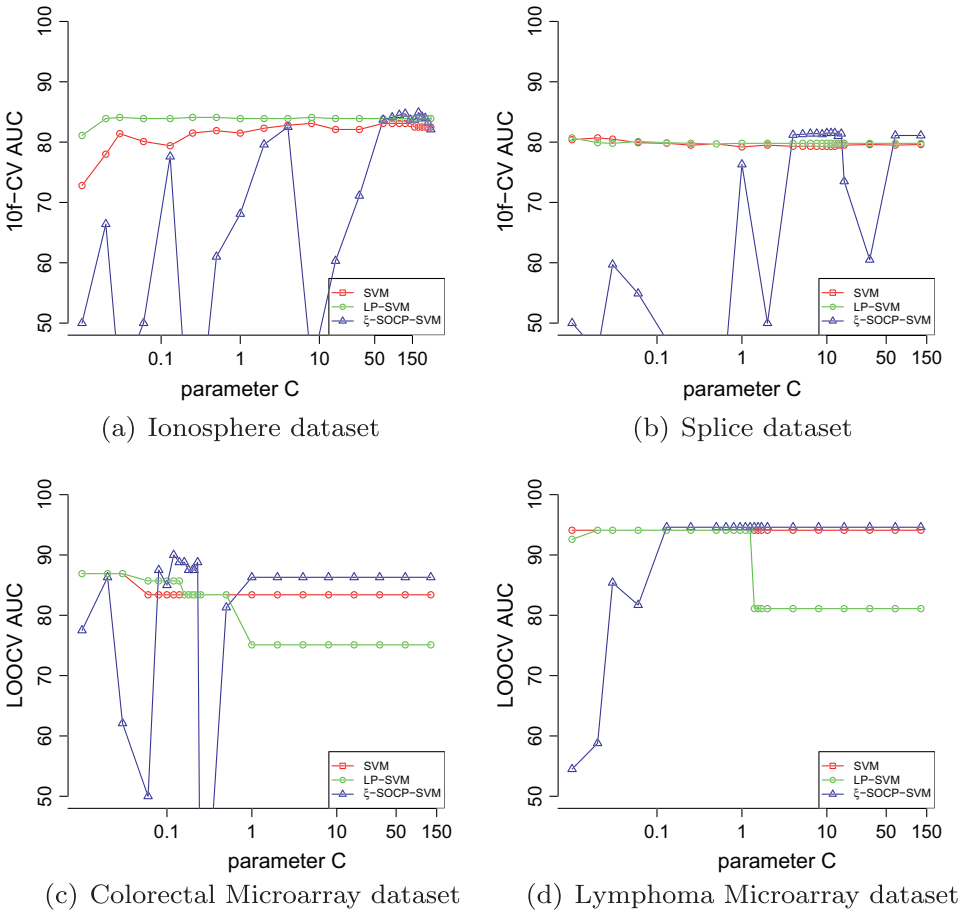

Fig. 4. AUC by varying parameter C for standard SVM, LP-SVM and ξ-SOCP-SVM.

Table 4
Average running times, in seconds, and number of iterations for all datasets.

	AUS	WBC	DIA	GEC	ION	SPL	CMA	LMA
SVM	0.5	0.3	0.3	0.5	0.3	0.7	0.8	1.2
LP-SVM	0.2	0.1	0.1	0.3	0.1	0.4	0.4	0.7
SOCP	0.24(8)	0.25(8)	0.29(10)	0.46(10)	0.27(9)	0.61(9)	0.69(10)	1.91(11)
ξ-SOCP	0.19(6)	0.21(6)	0.2(6)	0.24(6)	0.21(6)	0.47(6)	0.55(6)	1.35(7)
r-SOCP	0.83(13)	0.82(10)	0.84(11)	1.05(16)	0.91(13)	1.32(15)	1.85(19)	7.39(23)

these methods to kernel approaches might lead to better performance, thanks to the ability of constructing non-linear classifiers. SOCP-SVM also presents interesting properties for classification on highly imbalanced data sets, and a redefinition of the margin variable r can be considered to adjust the hyperplane, and to favor the target data in the construction of the classifier.

There is a pressing need for more efficient implementations of second-order cone programming formulations. Our formulations were solved by using SeDuMi solver. An ad hoc algorithm for SOCP-SVM, however, would potentially improve their running times. For instance, an iterative scheme (e.g. [11]) could be implemented to find the distance between two ellipsoids, thus solving the formulations (10) and (13). Faster implementations are necessary for the method to become a viable alternative to traditional SVM for large scale datasets.

Acknowledgements

The first author was supported by FONDECYT project 11121196, while the second was funded by FONDECYT project 11110188 and by CONICYT Anillo ACT1106. The authors are grateful to the anonymous reviewers who contributed to improving the quality of the original paper.

Appendix A. Dual formulation for LP-SVM

Let us denote the number of elements of the positive and negative class by m_1 and m_2, respectively; by $A \in \Re^{n\times m_1}$ a data matrix for the positive class; by $B \in \Re^{n\times m_2}$ a data matrix for the negative class; and by $\mathbf{e} = (1,\ldots,1)$ a vector of ones of appropriate dimension. We denote the vectors in $\Re^{m_1}$ by a subscript 1; those in $\Re^{m_2}$ by a subscript 2; and those in $\Re^n$ without a subscript. With this notation, the problem (3) can be rewritten as

$$\begin{aligned} \min_{\mathbf{w},r,b} \quad & -r \\ \text{s.t.} \quad & A^\top \mathbf{w} + (b-r)\mathbf{e} \geqslant 0, \\ & -B^\top \mathbf{w} - (b+r)\mathbf{e} \geqslant 0, \\ & -1 \leqslant w_j \leqslant 1, \quad j = 1,\ldots,n, \\ & r \geqslant 0. \end{aligned} \tag{14}$$

The Lagrangian function associated with problem (14) is given by

$$L(\mathbf{w},b,r,\mathbf{z}_1,\mathbf{z}_2,t,\mathbf{u},\mathbf{v}) = -r - \langle A^\top \mathbf{w} + (b-r)\mathbf{e}, \mathbf{z}_1\rangle - \langle -B^\top \mathbf{w} - (b+r)\mathbf{e}, \mathbf{z}_2\rangle - rt - \langle \mathbf{e}-\mathbf{w}, \mathbf{u}\rangle - \langle \mathbf{e}+\mathbf{w}, \mathbf{v}\rangle.$$

Therefore, the dual problem of the linear one (14) is the following:

$$\begin{aligned} &\max \ L(\mathbf{w},b,r,\mathbf{z}_1,\mathbf{z}_2,t,\mathbf{u},\mathbf{v}) \\ &\text{s.t. } \nabla_{\mathbf{w}} L = -A\mathbf{z}_1 + B\mathbf{z}_2 + \mathbf{u} - \mathbf{v} = 0, \\ &\frac{\partial L}{\partial b} = -\mathbf{e}^\top \mathbf{z}_1 + \mathbf{e}^\top \mathbf{z}_2 = 0, \\ &\frac{\partial L}{\partial r} = -1 + \mathbf{e}^\top \mathbf{z}_1 + \mathbf{e}^\top \mathbf{z}_2 - t = 0, \\ &\mathbf{z}_1 \geq 0, \mathbf{z}_2 \geq 0, t \geq 0, \mathbf{u} \geq 0, \mathbf{v} \geq 0, \end{aligned}$$

where $\nabla_{\mathbf{w}} L$ denotes the gradient of L with respect to the vector $\mathbf{w}$. Simplifying, one has

$$\begin{aligned} \max_{\mathbf{u},\mathbf{v},\mathbf{z}_1,\mathbf{z}_2} \quad & -\mathbf{e}^\top(\mathbf{u}+\mathbf{v}) \\ \text{s.t. } & \mathbf{u} - \mathbf{v} = A\mathbf{z}_1 - B\mathbf{z}_2, \\ & \mathbf{e}^\top \mathbf{z}_1 = 1, \ \mathbf{e}^\top \mathbf{z}_2 = 1, \\ & \mathbf{z}_1 \geq 0, \mathbf{z}_2 \geq 0, \mathbf{u} \geq 0, \mathbf{v} \geq 0. \end{aligned} \tag{15}$$

Hence, the dual can be stated as follows:

$$\begin{aligned} \max_{\mathbf{z}_1,\mathbf{z}_2} \quad & -\|A\mathbf{z}_1 - B\mathbf{z}_2\|_1 \\ \text{s.t. } & \mathbf{e}^\top \mathbf{z}_1 = 1, \ \mathbf{e}^\top \mathbf{z}_2 = 1, \\ & \mathbf{z}_1 \geq 0, \mathbf{z}_2 \geq 0. \end{aligned}$$

The above problem finding the closest points of the convex hulls of the sets A and B, using the 1-norm.

Appendix B. Dual formulation for ξ-SOCP-SVM

We denote by

$$\mathbf{g}^i(\mathbf{w},b,\xi) = \begin{pmatrix} (-1)^{i-1}\mu_i^\top & (-1)^i & 1 \\ \kappa_i S_i^\top & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ b \\ \xi \end{pmatrix} + \begin{pmatrix} -1 \\ \mathbf{0} \end{pmatrix}, \tag{17}$$

for $i = 1,2$. Thus, the two conic constraints in (8) can be written as $\mathbf{g}^i(\mathbf{w},b,\xi) \in \mathcal{K}^{n+1}$ for $i = 1,2$, where $\mathcal{K}^{n+1} := \{y = (y_1,\bar{y}) \in \Re \times \Re^n : \|\bar{y}\| \leqslant y_1\}$. This set is called *second-order cone* or *Lorentz cone*.

The Lagrangian functional associated with problem (8) is given by

$$L(\mathbf{w},b,\xi,\lambda_1,\lambda_2,\tau) = \frac{1}{2}\|\mathbf{w}\|^2 + C\xi - \sum_{i=1}^{2} \langle \mathbf{g}^i(\mathbf{w},b,\xi), \lambda_i\rangle - \xi\tau. \tag{18}$$

Thus, the dual of the SOCP problem (8) is the following

$$\begin{aligned}
&\max\ L(\mathbf{w}, b, \xi, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \tau)\\
&\text{s.t. } \nabla_{\mathbf{w}} L = \mathbf{w} - (\mu_1\lambda_{11} + \kappa_1 S_1\bar{\boldsymbol{\lambda}}_1) - (-\mu_2\lambda_{21} + \kappa_2 S_2\bar{\boldsymbol{\lambda}}_2) = \mathbf{0},\\
&\quad \frac{\partial L}{\partial b} = \lambda_{11} - \lambda_{21} = 0,\\
&\quad \frac{\partial L}{\partial \xi} = C - \lambda_{11} - \lambda_{21} - \tau = 0,\\
&\quad \boldsymbol{\lambda}_1 = (\lambda_{11}, \bar{\boldsymbol{\lambda}}_1),\ \boldsymbol{\lambda}_2 = (\lambda_{21}, \bar{\boldsymbol{\lambda}}_2) \in \mathcal{K}^{n+1},\ \tau \geq 0.
\end{aligned}$$

From the second and third equality it follows that $\tau = C - 2t$ with $t = \lambda_{11} = \lambda_{21}$. Then,

$$\mathbf{w} = (\mu_1 t + \kappa_1 S_1\bar{\boldsymbol{\lambda}}_1) - t(\mu_2 t - \kappa_2 S_2\bar{\boldsymbol{\lambda}}_2), \tag{19}$$

and, $t \leqslant \frac{C}{2}$ since $\tau \geq 0$. Consequently, the dual problem can be written as

$$\begin{aligned}
&\max_{t,\mathbf{u}_1,\mathbf{u}_2,\mathbf{z}_1,\mathbf{z}_2}\ -\frac{t^2}{2}\|\mathbf{z}_1 - \mathbf{z}_2\|^2 + 2t\\
&\text{s.t. } \mathbf{z}_1 = \mu_1 + \kappa_1 S_1\mathbf{u}_1,\ \mathbf{z}_2 = \mu_2 - \kappa_2 S_2\mathbf{u}_2\\
&\quad \|\mathbf{u}_1\| \leq 1,\ \|\mathbf{u}_2\| \leq 1,\ 0 \leqslant t \leqslant \frac{C}{2}.
\end{aligned} \tag{20}$$

It is clear that the objective function, in the variable t, is maximized at the point

$$t = \frac{2}{\|\mathbf{z}_1 - \mathbf{z}_2\|^2}, \quad \text{whenever } \|\mathbf{z}_1 - \mathbf{z}_2\| \geqslant \frac{2}{\sqrt{C}}, \tag{21}$$

and with maximum value $\frac{2}{\|\mathbf{z}_1 - \mathbf{z}_2\|^2}$. Then, by using (21) the dual problem of (8) can be stated as follows

$$\begin{aligned}
&\min_{\mathbf{z}_1,\mathbf{z}_2}\ \frac{1}{2}\|\mathbf{z}_1 - \mathbf{z}_2\|^2\\
&\text{s.t. } \mathbf{z}_i \in \mathbf{B}_i(\mu_i, S_i, \kappa_i),\ i = 1, 2,\\
&\quad \|\mathbf{z}_1 - \mathbf{z}_2\| \geqslant \frac{2}{\sqrt{C}},
\end{aligned} \tag{22}$$

where

$$\mathbf{B}_i(\mu_i, S_i, \kappa_i) = \{\mathbf{z}_i : \mathbf{z}_i = \mu_i - (-1)^i\kappa_i S_i\mathbf{u}_i, \|\mathbf{u}_i\| \leq 1\},\ i = 1, 2. \tag{23}$$

Lemma 1. *The Lagrange multipliers associated with the conic constraints of the SOC problem (8) are always different from zero.*

Proof. The Karush–Kuhn–Tucker (KKT) conditions for the problem (8) are the following:

$$\nabla_{\mathbf{w}} L = \mathbf{0}, \quad \frac{\partial L}{\partial b} = 0, \quad \frac{\partial L}{\partial \xi} = 0, \tag{24}$$

$$\boldsymbol{\lambda}_i^{\top} \cdot \mathbf{g}^i(\mathbf{w}, b, \xi) = 0,\ i = 1, 2, \tag{25}$$

$$\xi\tau = 0, \tag{26}$$

$$\boldsymbol{\lambda}_i, \mathbf{g}^i(\mathbf{w}, b, \xi) \in \mathcal{K}^{n+1}, i = 1, 2, \tag{27}$$

$$\xi, \tau \geq 0. \tag{28}$$

From the expression $\nabla_{\mathbf{w}} L = \mathbf{0}$, we obtain

$$\|\mathbf{w}\|^2 - (\lambda_{11}\mu_1^{\top} \cdot \mathbf{w} + \kappa_1\bar{\boldsymbol{\lambda}}_1^{\top} \cdot S_1^{\top}\mathbf{w}) - (-\lambda_{21}\mu_2^{\top} \cdot \mathbf{w} + \kappa_2\bar{\boldsymbol{\lambda}}_2^{\top} \cdot S_2^{\top}\mathbf{w}) = 0. \tag{29}$$

And, from (25) and (17) one has

$$\begin{aligned}
&\lambda_{11}\mathbf{w}^{\top} \cdot \mu_1 + \kappa_1\bar{\boldsymbol{\lambda}}_1^{\top} \cdot S_1^{\top}\mathbf{w} = \lambda_{11}(-\xi + b + 1),\\
&-\lambda_{21}\mathbf{w}^{\top} \cdot \mu_2 + \kappa_2\bar{\boldsymbol{\lambda}}_2^{\top} \cdot S_2^{\top}\mathbf{w} = \lambda_{21}(-\xi - b + 1).
\end{aligned} \tag{30}$$

Substituting (30) in (29) we get

$$\|\mathbf{w}\|^2 - \lambda_{11}(-\xi + b + 1) - \lambda_{21}(-\xi - b + 1) = 0.$$

But $\lambda_{11} = \lambda_{21}$ (cf. (24)), so

$$\|\mathbf{w}\|^2 = 2\lambda_{11}(1 - \xi), \tag{31}$$

which together with (28) implies that $\xi \in [0,1]$. Indeed, if $\lambda_{11} = 0$, then $\mathbf{w} = \mathbf{0}$ and from the second equality in (24) and (26) it follows that $\xi = 0$. Moreover, note that in this last case we obtain that $-b \geq 1$ and $b \geq 1$ from the two first constraints of the problem (8), which is a contradiction. Thus, we have also proven that the Lagrange multipliers $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$ associated with the conic constraints are always different from zero. □

Appendix C. Dual formulation for *r*-SOCP-SVM

The Lagrangian associated with formulation (12) is given by:

$$L(\mathbf{w}, b, r, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \mathbf{z}_1, \mathbf{z}_2, \tau) = -r - \sum_{i=1}^{2} \langle \mathbf{g}^i(\mathbf{w}, b, \xi), \boldsymbol{\lambda}_i \rangle - r\tau - \langle \mathbf{e} - \mathbf{w}), \mathbf{z}_1 \rangle - \langle \mathbf{e} + \mathbf{w}), \mathbf{z}_2 \rangle,$$

where

$$\mathbf{g}^i(\mathbf{w}, b, r) = \begin{pmatrix} (-1)^{i-1}\mu_i^\top & (-1)^i & -1 \\ \kappa_i S_i^\top & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ b \\ r \end{pmatrix}, \quad \text{for } i = 1, 2. \tag{32}$$

Thus, the dual formulation of the linear SOCP problem (12) is

$$\begin{aligned} \max\ & L(\mathbf{w}, b, r, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \mathbf{z}_1, \mathbf{z}_2, \tau) \\ \text{s.t. } \nabla_{\mathbf{w}} L = -(\mu_1 \lambda_{11} + \kappa_1 S_1 \bar{\boldsymbol{\lambda}}_1) - (-\mu_2 \lambda_{21} + \kappa_2 S_2 \bar{\boldsymbol{\lambda}}_2) + \mathbf{z}_1 - \mathbf{z}_2 &= \mathbf{0}, \\ \frac{\partial L}{\partial b} = \lambda_{11} - \lambda_{21} &= 0, \\ \frac{\partial L}{\partial r} = -1 + \lambda_{11} + \lambda_{21} - \tau &= 0, \\ \boldsymbol{\lambda}_1 = (\lambda_{11}, \bar{\boldsymbol{\lambda}}_1),\ \boldsymbol{\lambda}_2 = (\lambda_{21}, \bar{\boldsymbol{\lambda}}_2) \in \mathcal{K}^{n+1},\ \tau \geq 0,\ \mathbf{z}_1, \mathbf{z}_2 &\geq 0. \end{aligned}$$

Simplifying, one has

$$\begin{aligned} \max_{\lambda, \mathbf{z}_1, \mathbf{z}_2, \mathbf{u}_1, \mathbf{u}_2}\ & -\boldsymbol{e}^\top(\mathbf{z}_1 + \mathbf{z}_2) \\ \text{s.t. } \mathbf{z}_1 - \mathbf{z}_2 &= \lambda(\tilde{\mathbf{z}}_1 - \tilde{\mathbf{z}}_2), \\ \tilde{\mathbf{z}}_i = \mu_i - (-1)^i \kappa_i S_i \mathbf{u}_i,\ \|\mathbf{u}_i\| &\leq 1,\ i = 1, 2, \\ \lambda \geqslant \frac{1}{2},\ \mathbf{z}_1 &\geq 0,\ \mathbf{z}_2 \geq 0. \end{aligned}$$

Then, the dual problem of (12) can be stated as follows:

$$\begin{aligned} -\min_{\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2}\ & \frac{1}{2}\|\tilde{\mathbf{z}}_1 - \tilde{\mathbf{z}}_2\|_1 \\ \text{s.t. } \tilde{\mathbf{z}}_i \in \mathbf{B}_i(\mu_i, S_i, \kappa_i),\ & i = 1, 2, \end{aligned}$$

where $\mathbf{B}_i$ is defined in (23).

References

[1] S. Ali, K.A. Smith-Miles, On learning algorithm selection for classification, Appl. Soft Comput. 6 (2006) 119–138.
[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, D. Goldfarb, Second-order cone programming, Math. Program. 95 (2003) 3–51.
[4] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, 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, pp. 6745–6750.
[5] F. Alvarez, J. López, H. RamírezC, Interior proximal algorithm with variable metric for second-order cone programming: applications to structural optimization and support vector machines, Optim. Method. Softw. 25 (6) (2010) 859–881.
[6] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, 2007.
[7] T. Bäck, Adaptive business intelligence based on evolution strategies: some application examples of self-adaptive software, Inform. Sci. 148 (1–4) (2002) 113–121.
[8] C. Bhattacharyya, Second order cone programming formulations for feature selection, J. Mach. Learn. Res. 5 (2004) 1417–1433.
[9] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[10] G. Lanckriet, L.E. Ghaoui, C. Bhattacharyya, M. Jordan, A robust minimax approach to classification, J. Mach. Learn. Res. 3 (2003) 555–582.
[11] A. Lian, S.P. Han, On the distance between two ellipsoids, SIAM J. Optim. 13 (2002) 298–308.
[12] S. Maldonado, R. Weber, A wrapper method for feature selection using support vector machines, Inform. Sci. 179 (2009) 2208–2217.
[13] O.L. Mangasarian, A finite newton method for classification, Optim. Method. Softw. 17 (5) (2002) 913–929.
[14] D. Meyer, F. Leisch, K. Hornik, The support vector machine under test, Neurocomputing 55 (2003) 169–186.
[15] S. Nath, 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.
[16] X. Peng, Building sparse twin support vector machine classifiers in primal space, Inform. Sci. 181 (18) (2001) 3967–3980.

[17] J. Platt, Advances in Kernel Methods-Support Vector Learning, MIT Press, Cambridge, MA, 1999. Chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208.
[18] P.K. Shivaswamy, C. Bhattacharyya, A.J. Smola, Second order cone programming approaches for handling missing and uncertain data, J. Mach. Learn. Res. 7 (2006) 1283–1314.
[19] M. Sokolova, N. Japkowicz, 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, pp. 1015–1021.
[20] L. Song, A. Smola, A. Gretton, J. Bedo, K. Borgwardt, Feature selection via dependence maximization, J. Mach. Learn. Res. 13 (2012) 1393–1434.
[21] E. Straszecka, Combining uncertainty and imprecision in models of medical diagnosis, Inform. Sci. 176 (20) (2006) 3026–3059.
[22] J.F. Sturm, Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, Optim. Method. Softw. 11 (12) (1999) 625–653.
[23] L.C. Thomas, J.N. Crook, D.B. Edelman, Credit Scoring and its Applications, SIAM, 2002.
[24] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[25] V. Vapnik, A. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, Theory Probab. Appl. 16 (2) (1971) 264–280.
[26] J. Weston, A. Elisseeff, G. Baklr, F. Sinz, The Spider Machine Learning Toolbox, 2005.
[27] S.J. Yen, Y.C. Wu, J.C. Yang, Y.S. Lee, C.J. Lee, J.J. Liu, A support vector machine-based context-ranking model for question answering, Inform. Sci. 224 (2013) 77–87.
[28] Y. Zhao, Y. Lu, Y. Tian, L. Li, Q. Ren, X. Chai, Image processing based recognition of images with a limited number of pixels using simulated prosthetic vision, Inform. Sci. 180 (16) (2010) 2915–2924.
[29] W. Zhou, L. Zhang, L. Jiao, Linear programming support vector machines, Patt. Recogn. 35 (2002) 2927–2936.