Appl Intell (2016) 45:265–276
DOI 10.1007/s10489-016-0764-4

A second-order cone programming formulation for twin support vector machines

Sebastián Maldonado[1] · **Julio López**[2] · **Miguel Carrasco**[1]

Published online: 23 February 2016

Abstract Second-order cone programming (SOCP) formulations have received increasing attention as robust optimization schemes for Support Vector Machine (SVM) classification. These formulations study the worst-case setting for class-conditional densities, leading to potentially more effective classifiers in terms of performance compared to the standard SVM formulation. In this work we propose an SOCP extension for Twin SVM, a recently developed classification approach that constructs two nonparallel classifiers. The linear and kernel-based SOCP formulations for Twin SVM are derived, while the duality analysis provides interesting geometrical properties of the proposed method. Experiments on benchmark datasets demonstrate the virtues of our approach in terms of classification performance compared to alternative SVM methods.

Keywords Support vector classification · Twin support vector machines · Second-order cone programming

✉ Sebastián Maldonado
smaldonado@uandes.cl

Julio López
julio.lopez@udp.cl

Miguel Carrasco
micarrasco@uandes.cl

[1] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Monseñor Álvaro del Portillo, 12455, Las Condes, Santiago, Chile

[2] Facultad de Ingeniería, Universidad Diego Portales, Ejército 441, Santiago, Chile

1 Introduction

Twin SVM [14] has gained popularity in the pattern analysis community due to its superior classification performance and its interesting geometrical properties [25, 29]. This method aims at constructing two classifiers in such a way that each one is close to one of the two training patterns, and as far as possible from the other. The method is potentially faster than SVM since it divides the original problem into two smaller subproblems, and may achieve better empirical results [14].

Second-order cone programming has received increased interest in recent years within the SVM community [10, 18, 32]. In this paper we use the SOCP-SVM formulation proposed by Nath and Bhattacharyya [24], which provides a robust setting in which a maximum margin classifier is constructed in such a way that the true positive and true negative rates should be above a predefined value.

It is important to note this method differs from the SOCP formulations for SVM proposed by Goldfarb and Iyengar [12] and Zhong and Fukushima [34], which study the problem of classification with noisy data (i.e. instances with measurement errors). The SOCP methods by Goldfarb and Iyengar [12] and Zhong and Fukushima [34] have as many constraints as training samples, resulting in models that are computationally very expensive in terms of running times. In contrast, the SOCP-SVM method by Nath and Bhattacharyya [24] has two chance constraints, one for each training pattern, resulting in a much more efficient training.

Several improvements have been made to the original Twin SVM version by Jayadeva et al. [14] (TWSVM). One extension of TWSVM, namely the Twin bounded SVM (TBSVM) formulation [28], represents the base formulation

of our analysis. The aim of this work is to develop a new classification approach combining the ideas of Twin SVM and SOCP-SVM. On one hand, our proposal constructs two nonparallel classifiers so that each hyperplane is closer to one of the training patterns and as far as possible from the other (Twin SVM principle) while, on the other hand, each training pattern is represented by an ellipsoid characterized by the mean and covariance of each class. The proposal is transposed into two SOCP models, which can be solved efficiently by interior point algorithms [1, 2]. This proposal is also extended to nonlinear classification via kernel methods.

To the best of our knowledge, the only reference that combines SOCP-SVM and Twin SVM is the work proposed by Qi et al. [26]. In this work, the authors extend the SOCP-based SVM formulation by Goldfarb and Iyengar [12] and Zhong and Fukushima [34] to Twin SVM.

This paper is structured as follows: Section 2 introduces SVM for binary classification and the relevant extensions for this work: Twin SVM and SOCP-SVM. The proposed SOCP-SVM approach is presented in Section 3. Section 4 provides experimental results using benchmark datasets. A summary of this work can be found in Section 5, where we provide its main conclusions and address future developments.

2 Prior work in support vector machines

In this section, we describe the SVM formulations for binary classification developed by Cortes and Vapnik [8]. Subsequently, the Twin SVM formulation [14, 28] is presented. Finally, the SVM based on second-order cone programming [6, 24] is described.

2.1 Soft margin SVM

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 soft-margin SVM is aimed at finding a hyperplane of the form $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ by solving the following quadratic programming problem (QPP):

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

where ξ_i is the soft margin error of the i-th training point and $C > 0$ is a regularization parameter.

A non-linear classifier can be obtained via the Kernel Trick. The dual of Formulation (1) allows the use of kernel functions, which define an inner product in a higher dimensional Hilbert space, where a hyperplane with maximal margin is constructed. The kernel-based SVM formulation follows [27]:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,s=1}^{m}\alpha_i\alpha_s y_i y_s \mathcal{K}(\mathbf{x}_i, \mathbf{x}_s)$$
$$\text{s.t.} \sum_{i=1}^{m}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m, \qquad (2)$$

where $\boldsymbol{\alpha}$ are the dual variables corresponding to the constraints in (1), and $\mathcal{K}\colon \Re^n \times \Re^n \to \Re$ is a kernel function satisfying the Mercer's condition (see [23]). A common choice is the *Gaussian kernel*, which has the following form:

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

where σ is a positive parameter that controls the width of the kernel [27].

2.2 Twin support vector machine

The Twin SVM is a binary classification method that performs classification using two nonparallel hyperplanes instead of the single hyperplane used in the classical SVM [14]. These two hyperplanes are obtained by solving two smaller-sized QPPs.

Let us denote the number of elements of the positive and negative class by m_1 and m_2 respectively, by $A \in \Re^{m_1 \times n}$ a data matrix for the positive class (i.e. for $y_i = +1$), and by $B \in \Re^{m_2 \times n}$ a data matrix for the negative class (i.e. for $y_i = -1$).

The linear Twin SVM formulation finds two non-parallel hyperplanes in $\Re^n$ of the form

$$\mathbf{w}_1^\top \mathbf{x} + b_1 = 0, \quad \mathbf{w}_2^\top \mathbf{x} + b_2 = 0, \qquad (4)$$

in such a way that each hyperplane is closer to data points of one of the two classes, and is as far as possible from those of the other class. A new point is assigned to class $+1$ ($k = 1$) or -1 ($k = 2$) depending on its proximity to the two non-parallel hyperplanes.

Formally, the linear Twin SVM method solves the following two QPPs:

$$\min_{\mathbf{w}_1,b_1,\boldsymbol{\xi}_2} \frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{c_3}{2}\left(\|\mathbf{w}_1\|^2 + b_1^2\right) + c_1\mathbf{e}_2^\top\boldsymbol{\xi}_2$$
$$\text{s.t.} -(B\mathbf{w}_1 + \mathbf{e}_2 b_1) \geq \mathbf{e}_2 - \boldsymbol{\xi}_2, \quad \boldsymbol{\xi}_2 \geq 0, \qquad (5)$$

and

$$\min_{\mathbf{w}_2,b_2,\boldsymbol{\xi}_1} \frac{1}{2}\|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 + \frac{c_4}{2}\left(\|\mathbf{w}_2\|^2 + b_2^2\right) + c_2\mathbf{e}_1^\top\boldsymbol{\xi}_1$$
$$\text{s.t.}(A\mathbf{w}_2 + \mathbf{e}_1 b_2) \geq \mathbf{e}_1 - \boldsymbol{\xi}_1, \quad \boldsymbol{\xi}_1 \geq 0, \qquad (6)$$

where c_1, c_2, c_3, and c_4 are positive parameters, and $\mathbf{e}_1$ and $\mathbf{e}_2$ are vectors of ones of appropriate dimensions. Here, c_1 and c_2 determine the tradeoff between the respective

model fit (the first term of the objective function of problems (5) and (6) is the sum of the squared distances from the hyperplane to instances of this studied class), and the sum of the slack variables (thus attempting to minimize misclassification).

Parameters c_3 and c_4 relate the second term of the objective function of problems (5) and (6) to the first and third ones described above, and act as a regularization term in the dual of these formulations to avoid possible ill-conditioning at inverting matrices. Let $H = [A\ \mathbf{e}_1] \in \Re^{m_1\times(n+1)}$ and $G = [B\ \mathbf{e}_2] \in \Re^{m_2\times(n+1)}$. Since the symmetric matrices $H^\top H + c_3 I$ and $G^\top G + c_4 I$ are positive definite for any $c_3, c_4 > 0$, the Wolfe dual [22] of the problems (5) and (6) are given by:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^\top \mathbf{e}_2 - \frac{1}{2}\boldsymbol{\alpha}^\top G(H^\top H + c_3 I)^{-1} G^\top \boldsymbol{\alpha} \\ &\text{s.t.} 0 \le \boldsymbol{\alpha} \le c_1 \mathbf{e}_2, \end{aligned} \tag{7}$$

and

$$\begin{aligned} &\max_{\boldsymbol{\gamma}} \boldsymbol{\gamma}^\top \mathbf{e}_1 - \frac{1}{2}\boldsymbol{\gamma}^\top H(G^\top G + c_4 I)^{-1} H^\top \boldsymbol{\gamma} \\ &\text{s.t.} 0 \le \boldsymbol{\gamma} \le c_2 \mathbf{e}_1, \end{aligned} \tag{8}$$

respectively. The nonparallel hyperplanes (4) are obtained from the solution $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ of (7) and (8) by

$$\mathbf{v}_1 = -(H^\top H + c_3 I)^{-1} G^\top \boldsymbol{\alpha}, \quad \mathbf{v}_2 = (G^\top G + c_4 I)^{-1} H^\top \boldsymbol{\gamma},$$

where $\mathbf{v}_k = [\mathbf{w}_k^\top,\ b_k]^\top \in \Re^{n+1}$, for $k = 1, 2$.

Notice that Formulation (5)–(6) corresponds to the Twin-Bounded SVM formulation (TBSVM) given by Shao et al. [28], which extends the *original Twin SVM (TWSVM) proposed by Jayadeva et al.* [14]. Both problems are equivalent when setting $c_3 = c_4 = \epsilon$, with $\epsilon > 0$ a fixed small parameter.

A new sample $\mathbf{x}$ belongs to the class k^* iff

$$k^* = \underset{k=1,2}{\operatorname{argmin}} \left\{ d_k(\mathbf{x}) := \frac{|\mathbf{w}_k^\top \mathbf{x} + b_k|}{\|\mathbf{w}_k\|} \right\}, \tag{9}$$

where d_k is the perpendicular distance of the point $\mathbf{x}$ from the hyperplane $\mathbf{w}_k^\top \mathbf{x} + b_k = 0$, $k = 1, 2$.

Figure 1 presents the geometrical interpretation of Formulation (5)–(6) in a two-dimensional toy dataset:

In Fig. 1 we observe the two nonparallel hyperplanes (represented by dashed lines) over the respective training patterns, represented in the form of convex hulls [5, 8]. The solid line represents the decision function (9), which provides the boundaries for the classification of new observations to one of each class.

A kernel-based classifier can be derived by considering the following non-linear surfaces:

$$\mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{u}_1 + b_1 = 0, \text{ and } \mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{u}_2 + b_2 = 0, \tag{10}$$

where $\mathbb{X} = [A^\top\ B^\top] \in \Re^{n\times m}$ represents the matrix of both training patterns (sorted by class), $\mathcal{K}(\mathbf{x}, \mathbb{X})$ denotes a row vector, which is defined by

$$\mathcal{K}(\mathbf{x}, \mathbb{X}) = [\mathcal{K}(\mathbf{x}, \mathbb{X}_{\bullet 1}), \mathcal{K}(\mathbf{x}, \mathbb{X}_{\bullet 2}), \ldots, \mathcal{K}(\mathbf{x}, \mathbb{X}_{\bullet m})], \tag{11}$$

with $\mathbb{X}_{\bullet j} \in \Re^n$ denoting the j-th column of the matrix $\mathbb{X}$, and $\mathcal{K}\colon \Re^n \times \Re^n \to \Re$ is a kernel function that satisfies the Mercer's condition. For these surfaces, the following quadratic problems can be constructed (kernel-based Twin SVM):

$$\begin{aligned} &\min_{\mathbf{u}_1, b_1, \boldsymbol{\xi}_2} \frac{1}{2}\left\|\mathcal{K}(A^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_1 b_1\right\|^2 + \frac{c_3}{2}\left(\|\mathbf{u}_1\|^2 + b_1^2\right) + c_1 \mathbf{e}_2^\top \boldsymbol{\xi}_2 \\ &\text{s.t.} -(\mathcal{K}(B^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_2 b_1) \ge \mathbf{e}_2 - \boldsymbol{\xi}_2, \quad \boldsymbol{\xi}_2 \ge 0, \end{aligned} \tag{12}$$

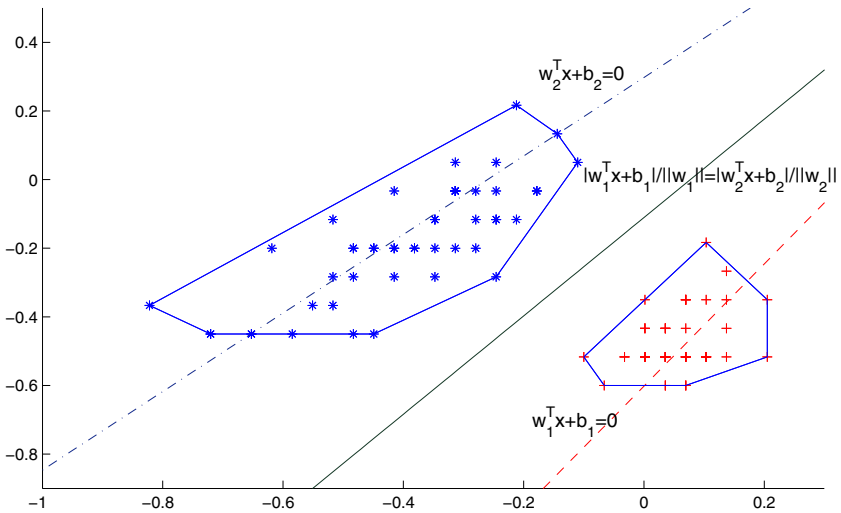

Fig. 1 Geometric interpretation for TBSVM

and

$$\min_{\mathbf{u}_2,b_2,\boldsymbol{\xi}_1} \frac{1}{2}\left\|\mathcal{K}(B^\top,\mathbb{X})\mathbf{u}_2+\mathbf{e}_2b_2\right\|^2+\frac{c_4}{2}\left(\|\mathbf{u}_2\|^2+b_2^2\right)+c_2\mathbf{e}_1^\top\boldsymbol{\xi}_1$$
$$\text{s.t.}(\mathcal{K}(A^\top,\mathbb{X})\mathbf{u}_2+\mathbf{e}_1b_2)\geq\mathbf{e}_1-\boldsymbol{\xi}_1,\quad \boldsymbol{\xi}_1\geq 0, \tag{13}$$

where c_1, c_2, c_3, and c_4 are positive parameters.

2.3 Second-order cone programming SVM

Here we introduce the robust SVM version based on second-order cones presented by Nath and Bhattacharyya [24]. Let $\mathbf{X}_1$ and $\mathbf{X}_2$ be random vectors that generate the observations of the positive and negative classes respectively, with means and covariance matrices given by $(\boldsymbol{\mu}_k,\Sigma_k)$ for $k=1,2$, where $\Sigma_k\in\Re^{n\times n}$ are symmetric positive semidefinite matrices.

The aim of this method is to construct a maximum margin classifier in such a way that the probability of false-negative (resp. false-positive) errors does not exceed $1-\eta_1$ (resp. $1-\eta_2$) with $\eta_1,\eta_2\in(0,1)$, which becomes the following quadratic chance-constrained programming problem:

$$\begin{array}{ll}\min_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2\\ \text{s.t.} & \Pr\{\mathbf{w}^\top\mathbf{X}_1+b\geq 1\}\geq\eta_1,\\ & \Pr\{\mathbf{w}^\top\mathbf{X}_2+b\leq -1\}\geq\eta_2.\end{array}\tag{14}$$

Formulation (14) suggests classifying each training pattern $k=1,2$ correctly, up to the rate η_k, even for the *worst data distribution*. For this goal, the probability constraints in (14) are replaced with their *robust* counterparts:

$$\inf_{\mathbf{X}_1\sim(\boldsymbol{\mu}_1,\Sigma_1)}\Pr\{\mathbf{w}^\top\mathbf{X}_1+b\geq 1\}\geq\eta_1,$$
$$\inf_{\mathbf{X}_2\sim(\boldsymbol{\mu}_2,\Sigma_2)}\Pr\{\mathbf{w}^\top\mathbf{X}_2+b\leq -1\}\geq\eta_2.$$

The application of the multivariate Chebyshev inequality [15, Lemma 1] leads to the following (conic) constraints:

$$\mathbf{w}^\top\boldsymbol{\mu}_1+b\geq 1+\kappa_1\sqrt{\mathbf{w}^\top\Sigma_1\mathbf{w}},\quad -(\mathbf{w}^\top\boldsymbol{\mu}_2+b)\geq 1+\kappa_2\sqrt{\mathbf{w}^\top\Sigma_2\mathbf{w}},$$

where $\kappa_k=\sqrt{\frac{\eta_k}{1-\eta_k}}$, for $k=1,2$. Hence, the following deterministic problem can be derived:

$$\begin{array}{ll}\min\limits_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2\\ \text{s.t.} & \mathbf{w}^\top\boldsymbol{\mu}_1+b\geq 1+\kappa_1\|S_1^\top\mathbf{w}\|,\\ & -(\mathbf{w}^\top\boldsymbol{\mu}_2+b)\geq 1+\kappa_2\|S_2^\top\mathbf{w}\|,\end{array}\tag{15}$$

where $\Sigma_k=S_kS_k^\top$, for $k=1,2$. This problem is an instance of quadratic SOCP with two second-order cones (SOCs) constraints [1]. An SOC constraint on the variable $\mathbf{x}\in\Re^n$ is of the form

$$\|D\mathbf{x}+\mathbf{b}\|\leq\mathbf{c}^\top\mathbf{x}+d,$$

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

Note that Problem (15) can be cast into a linear SOCP with three SOCs constraints by introducing a new variable t and an additional constraint $\|\mathbf{w}\|\leq t$. The solutions for both problems are essentially the same but linear SOCP formulations are required by some SOCP solvers, such as SeDuMi Toolbox for Matlab [31].

A Kernel version can be derived for non-linear classification [6, 24]. The weight vector $\mathbf{w}\in\Re^n$ can be rewritten as $\mathbf{w}=\mathbb{X}\mathbf{s}+M\mathbf{r}$, where M is a matrix whose columns (as vectors) are orthogonal to the training data points, $\mathbf{s}$, $\mathbf{r}$ are vectors of combining coefficients with the appropriate dimension, and $\mathbb{X}=[A^\top\ B^\top]\in\Re^{n\times m}$ is the data matrix containing both training patterns.

On the other hand, the empirical estimates of the mean $\boldsymbol{\mu}_k$ and covariance Σ_k are given by:

$$\hat{\boldsymbol{\mu}}_1=\frac{1}{m_1}A_1^\top\mathbf{e}_1,\quad \hat{\boldsymbol{\mu}}_2=\frac{1}{m_2}B_2^\top\mathbf{e}_2,\quad \hat{\Sigma}_k=S_kS_k^\top,\ k=1,2, \tag{16}$$

where

$$S_1=\frac{1}{\sqrt{m_1}}(A^\top-\hat{\boldsymbol{\mu}}_1\mathbf{e}_1^\top),\quad S_2=\frac{1}{\sqrt{m_2}}(B^\top-\hat{\boldsymbol{\mu}}_2\mathbf{e}_2^\top). \tag{17}$$

Thus, for each class k, we have

$$\mathbf{w}^\top\boldsymbol{\mu}_k=\mathbf{s}^\top\mathbf{g}_k,\quad \mathbf{w}^\top\Sigma_k\mathbf{w}=\mathbf{s}^\top\Xi_k\mathbf{s},\quad k=1,2,$$

where

$$\mathbf{g}_k=\frac{1}{m_k}\begin{bmatrix}\mathbf{K}_{1k}\mathbf{e}_k\\ \mathbf{K}_{2k}\mathbf{e}_k\end{bmatrix},\quad \Xi_k=\frac{1}{m_k}\begin{bmatrix}\mathbf{K}_{1k}\\ \mathbf{K}_{2k}\end{bmatrix}\left(I_{m_k}-\frac{1}{m_k}\mathbf{e}_k\mathbf{e}_k^\top\right)\begin{bmatrix}\mathbf{K}_{1k}^\top\ \mathbf{K}_{2k}^\top\end{bmatrix},$$

with $\mathbf{K}_{11}=AA^\top$, $\mathbf{K}_{12}=\mathbf{K}_{21}^\top=BA^\top$, $\mathbf{K}_{22}=BB^\top$ matrices whose elements are inner products of data points. For instance, the entry (l,s) for the matrix $\mathbf{K}_{kk'}$ is the following $(\mathbf{K}_{kk'})_{ls}=(\mathbf{x}_l^k)^\top\mathbf{x}_s^{k'}$. Using a kernel function, this quantity becomes: $(\mathbf{K}_{kk'})_{ls}=\mathcal{K}(\mathbf{x}_l^k,\mathbf{x}_s^{k'})$.

Finally, the non-linear formulation is given by:

$$\begin{array}{ll}\min\limits_{\mathbf{s},b} & \frac{1}{2}\mathbf{s}^\top\mathbf{K}\mathbf{s}\\ \text{s.t.} & \mathbf{s}^\top\mathbf{g}_1-b\geq 1+\kappa_1\sqrt{\mathbf{s}^\top\Xi_1\mathbf{s}}\\ & b-\mathbf{s}^\top\mathbf{g}_2\geq 1+\kappa_2\sqrt{\mathbf{s}^\top\Xi_2\mathbf{s}},\end{array}\tag{18}$$

where $\mathbf{K}=[\mathbf{K}_{11},\mathbf{K}_{12};\mathbf{K}_{21},\mathbf{K}_{22}]\in\Re^{m\times m}$.

3 Twin SOCP-SVM, a robust Twin SVM classifier

In this section, we present a novel approach for binary classification using second-order cones and non-parallel hyperplanes. This formulation extends the ideas of the TBSVM approach [28] to SOCP-SVM.

The reasoning behind this approach is developing two nonparallel classifiers in such a way that each hyperplane is closest to one of the two classes and as far as possible from the other class. However, ellipsoids are used to characterize each training pattern instead of the convex hulls, following the ideas of SOCP-SVM.

The linear formulation of Twin SOCP-SVM is presented in Section 3.1. The dual form of Twin SOCP-SVM is derived in Section 3.2, providing the geometrical interpretation of the method. The kernel-based version of Twin SOCP-SVM is described in Section 3.3.

3.1 Linear Twin SOCP-SVM formulation

Let us consider the following quadratic chance-constrained programming problems:

$$\min_{\mathbf{w}_1,b_1} \quad \tfrac{1}{2}\|A\mathbf{w}_1+\mathbf{e}_1b_1\|^2+\tfrac{\theta_1}{2}\left(\|\mathbf{w}_1\|^2+b_1^2\right)$$
$$\text{s.t.} \quad \inf_{\mathbf{X}_2\sim(\boldsymbol{\mu}_2,\Sigma_2)} \Pr\{\mathbf{w}_1{}^\top\mathbf{X}_2+b_1\le -1\}\ge\eta_2,$$

and

$$\min_{\mathbf{w}_2,b_2} \quad \tfrac{1}{2}\|B\mathbf{w}_2+\mathbf{e}_2b_2\|^2+\tfrac{\theta_2}{2}\left(\|\mathbf{w}_2\|^2+b_2^2\right)$$
$$\text{s.t.} \quad \inf_{\mathbf{X}_1\sim(\boldsymbol{\mu}_1,\Sigma_1)} \Pr\{\mathbf{w}_2{}^\top\mathbf{X}_1+b_2\ge 1\}\ge\eta_1,$$

where $\theta_1,\theta_2>0$. The parameters η_1 and η_2 have a similar interpretation compared with the SOCP-SVM formulation, with values in $(0,1)$.

Thanks to an appropriate application of the multivariate Chebyshev inequality, the above problems can now be stated as the following quadratic SOCP problems (Twin SOCP-SVM formulation):

$$\min_{\mathbf{w}_1,b_1} \ \frac{1}{2}\|A\mathbf{w}_1+\mathbf{e}_1b_1\|^2+\frac{\theta_1}{2}\left(\|\mathbf{w}_1\|^2+b_1^2\right)$$
$$\text{s.t.} \ -\mathbf{w}_1^\top\boldsymbol{\mu}_2-b_1\ge 1+\kappa_2\|S_2^\top\mathbf{w}_1\|, \tag{19}$$

and

$$\min_{\mathbf{w}_2,b_2} \ \frac{1}{2}\|B\mathbf{w}_2+\mathbf{e}_2b_2\|^2+\frac{\theta_2}{2}\left(\|\mathbf{w}_2\|^2+b_2^2\right)$$
$$\text{s.t.} \ \mathbf{w}_2^\top\boldsymbol{\mu}_1+b_2\ge 1+\kappa_1\|S_1^\top\mathbf{w}_2\|, \tag{20}$$

where $\Sigma_k=S_kS_k^\top$ and $\kappa_k=\sqrt{\frac{\eta_k}{1-\eta_k}}$ for $k=1,2$.

Remark 1 Note that the objective functions of problems (19)–(20) can be written compactly as

$$\frac{1}{2}\|A\mathbf{w}_1+\mathbf{e}_1b_1\|^2+\frac{\theta_1}{2}\left(\|\mathbf{w}_1\|^2+b_1^2\right)=\frac{1}{2}\mathbf{v}_1^\top(H^\top H+\theta_1 I)\mathbf{v}_1, \tag{21}$$

and

$$\frac{1}{2}\|B\mathbf{w}_2+\mathbf{e}_2b_2\|^2+\frac{\theta_2}{2}\left(\|\mathbf{w}_2\|^2+b_2^2\right)=\frac{1}{2}\mathbf{v}_2^\top(G^\top G+\theta_2 I)\mathbf{v}_2, \tag{22}$$

respectively, where $\mathbf{v}_k=[\mathbf{w}_k^\top,b_k]^\top\in\Re^{n+1}$,

$$H=[A\ \mathbf{e}_1]\in\Re^{m_1\times(n+1)},\quad G=[B\ \mathbf{e}_2]\in\Re^{m_2\times(n+1)}. \tag{23}$$

Then, by introducing the new variables t_1,t_2, and the constraints

$$\|(H^\top H+\theta_1 I)^{1/2}\mathbf{v}_1\|\le t_1,\quad \|(G^\top G+\theta_2 I)^{1/2}\mathbf{v}_2\|\le t_2,$$

the problems (19) and (20) can be cast into linear SOCP problems with two SOC constraints each.

The decision function is similar to the one used for the TBSVM method; that is, a new sample $\mathbf{x}$ belongs to the class k^* iff $k^*=\text{argmin}_{k=1,2}\left\{\frac{|\mathbf{w}_k^\top\mathbf{x}+b_k|}{\|\mathbf{w}_k\|}\right\}$.

It is important to note that the proposed formulation (19)-(20) differs from the one proposed by Qi et al. [26] in several aspects, such as the model structure and their goals. Specifically, the Robust Twin SVM (R-TWSVM) formulation [26] consists of solving the following quadratic SOCP problems:

$$\min_{\mathbf{w}_1,b_1,\boldsymbol{\xi}_2,t_1} \ \frac{1}{2}\|A\mathbf{w}_1+\mathbf{e}_1b_1\|^2+\frac{1}{2}t_1^2+c_1\mathbf{e}_2^\top\boldsymbol{\xi}_2$$
$$\text{s.t.} \ -(B\mathbf{w}_1+\mathbf{e}_2b_1)-t_1\mathbf{r_1}\ge\mathbf{e}_2-\boldsymbol{\xi}_2,\quad \boldsymbol{\xi}_2\ge 0,$$
$$\|\ \mathbf{w}_1\ \|\le t_1, \tag{24}$$

and

$$\min_{\mathbf{w}_2,b_2,\boldsymbol{\xi}_1,t_2} \ \frac{1}{2}\|B\mathbf{w}_2+\mathbf{e}_2b_2\|^2+\frac{1}{2}t_2^2+c_2\mathbf{e}_1^\top\boldsymbol{\xi}_1$$
$$\text{s.t.} \quad (A\mathbf{w}_2+\mathbf{e}_1b_2)-t_2\mathbf{r_2}\ge\mathbf{e}_1-\boldsymbol{\xi}_1,\quad \boldsymbol{\xi}_1\ge 0,$$
$$\mathbf{w}_2\le t_2, \tag{25}$$

where $\mathbf{r}_i$ is a vector whose components correspond to the radius of the ball around each training data, and represents a bound for the noise in each example. In our model we assume that the training samples are observed values generated by a random variable, which have a mean vector and a covariance matrix. This model leads to a formulation based on two quadratic optimization problems with one conic constraint each (Formulation (19)-(20)). Geometrically, this can be interpreted as separating two ellipsoids by a hyperplane which is obtained by constructing two non-parallel hyperplanes (see Section 3.2 and Fig. 2 for details). Conversely, the R-TWSVM method assumes that training data has measurement errors, which also leads to a SOC formulation but with additional linear constraints. Geometrically, this can be interpreted as separating two sets of spheres whose cardinalities are congruent with the sample sizes of each class (see [26, §1] for details).

3.2 Dual formulation of Twin SOCP-SVM and geometric interpretation

In this section we present the dual formulation of Twin SOCP-SVM (Formulations (19) and (20)), and provide geometrical insights for the method.

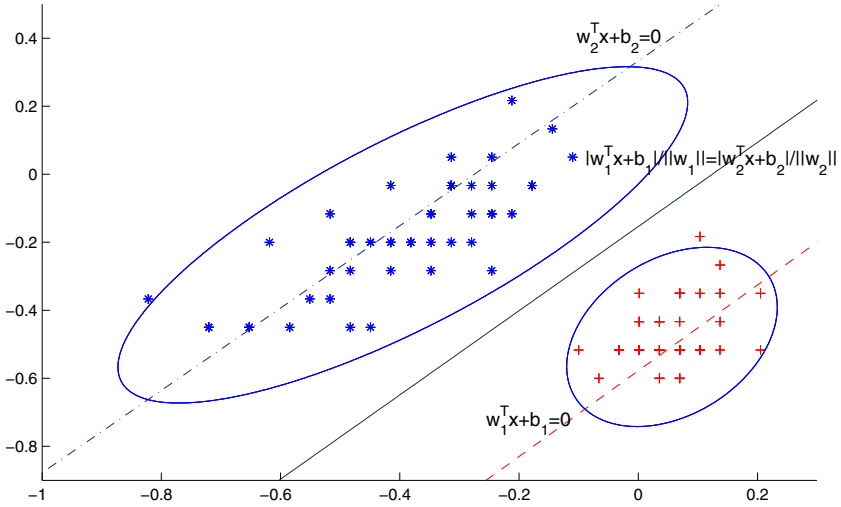

Fig. 2 Geometric interpretation for Twin SOCP-SVM

The following theorem gives the dual formulation of problems (19)–(20).

Theorem 1 *The duals of the problems* (19)–(20) *are given by*

$$\min_{\mathbf{z},\mathbf{u}} \frac{1}{2}\begin{pmatrix} z^\top & 1 \end{pmatrix}(H^\top H+\theta_1 I)^{-1}\begin{pmatrix} z \\ 1 \end{pmatrix}$$
$$s.t.\ \mathbf{z} \in \mathbf{B}(\boldsymbol{\mu}_2, S_2, \kappa_2), \tag{26}$$

and

$$\min_{\mathbf{p},\mathbf{u}} \frac{1}{2}\begin{pmatrix} \mathbf{p}^\top & 1 \end{pmatrix}(G^\top G+\theta_2 I)^{-1}\begin{pmatrix} \boldsymbol{p} \\ 1 \end{pmatrix}$$
$$s.t.\ \boldsymbol{p} \in \boldsymbol{B}(\boldsymbol{\mu}_1, S_1, \kappa_1), \tag{27}$$

where

$$\boldsymbol{B}(\boldsymbol{\mu}, S, \kappa) = \{z \in \Re^n : \mathbf{z} = \boldsymbol{\mu} + \kappa S\boldsymbol{u}, \|\boldsymbol{u}\| \leq 1\}, \tag{28}$$

which denotes an ellipsoid centered at $\boldsymbol{\mu}$ *whose shape is determined by* S*, and size by* κ*.*

The proof of Theorem 1 is presented in the Appendix A.1.

The previous result is important since we can link the proposed formulation to the geometrical interpretation: the ellipsoids $\mathbf{B}(\boldsymbol{\mu}, S, \kappa)$ define the two hyperplanes, and subsequently the classification rule. Figure 2 illustrates the geometrical interpretation of the proposed approach.

The following remark relates the primal and dual variables of the Twin SOCP-SVM formulation, which is relevant since we can solve the dual formulations and then obtain both non-parallel hyperplanes. The weights $\mathbf{w}_k$ provide interesting insight into the solution found, since we can assess the importance of each attribute in the final solution [4, 21].

Remark 2 Note that if $\mathbf{z}^* \in \Re^n$ is an optimal solution of Problem (26), then by using (A.7) and (A.5), the solution $\mathbf{v}_1^* = [\mathbf{w}_1^{*\top}, b_1^*]^\top$ of Problem (19) can be computed by:

$$\mathbf{v}_1^* = \frac{-1}{\hat{\mathbf{z}}^\top(H^\top H+\theta_1 I)^{-1}\hat{\mathbf{z}}}(H^\top H+\theta_1 I)^{-1}\hat{\mathbf{z}}, \quad \hat{\mathbf{z}} = [\mathbf{z}^{*\top}, 1]^\top. \tag{29}$$

Moreover, given an optimal solution $\mathbf{p}^* \in \Re^n$ of the problem (27), one can compute the solution $\mathbf{v}_2^* = [\mathbf{w}_2^{*\top}, b_2^*]^\top$ of Problem (20) by:

$$\mathbf{v}_2^* = \frac{1}{\hat{\mathbf{p}}^\top(G^\top G+\theta_2 I)^{-1}\hat{\mathbf{p}}}(G^\top G+\theta_2 I)^{-1}\hat{\mathbf{p}}, \quad \hat{\mathbf{p}} = [\mathbf{p}^{*\top}, 1]^\top. \tag{30}$$

The use of properties of Schur complement [13] of the matrices $(H^\top H+\theta_1 I)$ and $(G^\top G+\theta_2 I)$ allows us to rewrite the formulations (26)–(27) as follows:

Proposition 1 *The formulations* (26)–(27) *can be written equivalently as*

$$\min_{z,u} \frac{1}{2}\left\| C_s(\theta_1)^{-1/2}\left(z - \frac{m_1}{m_1+\theta_1}\hat{\boldsymbol{\mu}}_1\right)\right\|^2$$
$$s.t.\ z \in \boldsymbol{B}(\boldsymbol{\mu}_2, S_2, \kappa_2), \tag{31}$$

and

$$\min_{z,u} \frac{1}{2}\left\| C_s(\theta_2)^{-1/2}\left(z - \frac{m_2}{m_2+\theta_2}\hat{\boldsymbol{\mu}}_2\right)\right\|^2$$
$$s.t.\ z \in \boldsymbol{B}(\boldsymbol{\mu}_1, S_1, \kappa_1). \tag{32}$$

Table 1 The metadata 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
LIVER	6	345	(58.0,42.0)	1.4
GER	24	1000	(70.0,30.0)	2.3
DIA	8	768	(65.1,34.9)	1.9
HEART	13	270	(55.6,44.4)	1.25
IONO	34	351	(64.1,35.9)	1.8

When $\theta_1 = \theta_2 = 0$ *and the symmetric matrices* $H^\top H, G^\top G$ *are positive definite, the above problems are reduced to*

$$\min_{z,u} \frac{1}{2} \left\| \hat{\Sigma}_1^{-1/2}(z - \hat{\mu}_1) \right\|^2$$
$$s.t.\ z \in \boldsymbol{B}(\mu_2, S_2, \kappa_2), \tag{33}$$

and

$$\min_{z,u} \frac{1}{2} \left\| \hat{\Sigma}_2^{-1/2}(z - \hat{\mu}_2) \right\|^2$$
$$s.t.\ z \in \boldsymbol{B}(\mu_1, S_1, \kappa_1). \tag{34}$$

The proof of Proposition 1 can be found in the Appendix A.2.

Formulation (33) (resp. (34)) can be interpreted as the problem of minimizing the Mahalanobis distance [9] on $\mathbf{B}(\mu_2, S_2, \kappa_2)$ (resp. $\mathbf{B}(\mu_1, S_1, \kappa_1)$).

3.3 Kernel-based Twin SOCP-SVM formulation

In this section we extend Twin SOCP-SVM to Kernel functions to obtain non-linear classifiers. Following the notation introduced in Section 2.3, the weight vectors for each one of the twin hyperplanes can be written as $\mathbf{w}_k = \mathbb{X}\mathbf{s}_k + M\mathbf{r}_k$, where $\mathbb{X}$ and M are equivalent to the matrices described in Section 2.3, and $\mathbf{s}_k$, $\mathbf{r}_k$ are vectors with appropriate dimension. For each problem we have:

$$\mathbf{w}_k^\top \boldsymbol{\mu}_k = \mathbf{s}_k^\top \mathbf{g}_k, \quad \mathbf{w}_k^\top \Sigma_k \mathbf{w}_k = \mathbf{s}_k^\top \Xi_k \mathbf{s}_k, \quad k = 1, 2,$$

and

$$A\mathbf{w}_1 = [\mathbf{K}_{11}\ \mathbf{K}_{12}]\mathbf{s}_1 = \mathbf{K}_{1\bullet}\mathbf{s}_1, \ B\mathbf{w}_2 = [\mathbf{K}_{21}\ \mathbf{K}_{22}]\mathbf{s}_2 = \mathbf{K}_{2\bullet}\mathbf{s}_2,$$

where $\mathbf{g}_k$, Ξ_k, and $\mathbf{K}_{kk'}$ have a similar form compared to the notation presented in Section 2.3. Hence, in order to obtain a Kernel formulation for the problems (19) and (20), we replace the inner product that appears in the expressions $\mathbf{K}_{kk'}$ with any function $\mathcal{K}\colon \Re^n \times \Re^n \to \Re$ satisfying Mercer's condition (see [23]), obtaining the following problems (kernel-based Twin SOCP-SVM):

$$\min_{\mathbf{s}_1, b_1} \frac{1}{2} \|\mathbf{K}_{1\bullet}\mathbf{s}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta_1}{2} \left(\|\mathbf{s}_1\|^2 + b_1^2 \right)$$
$$\text{s.t.} \ -\mathbf{s}_1^\top \mathbf{g}_2 - b_1 \geq 1 + \kappa_2 \|\Lambda_2^\top \mathbf{s}_1\|, \tag{35}$$

and

$$\min_{\mathbf{s}_2, b_2} \frac{1}{2} \|\mathbf{K}_{2\bullet}\mathbf{s}_2 + \mathbf{e}_2 b_2\|^2 + \frac{\theta_2}{2} \left(\|\mathbf{s}_2\|^2 + b_2^2 \right)$$
$$\text{s.t.} \ \mathbf{s}_2^\top \mathbf{g}_1 + b_2 \geq 1 + \kappa_1 \|\Lambda_1^\top \mathbf{s}_2\|, \tag{36}$$

where $\Xi_k = \Lambda_k \Lambda_k^\top$, for $k = 1, 2$. Then, the solutions of problems (35) and (36) generate the following kernel-based surfaces:

$$\mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{s}_1 + b_1 = 0, \quad \mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{s}_2 + b_2 = 0, \tag{37}$$

where the row vector $\mathcal{K}(\mathbf{x}, \mathbb{X})$ is defined in (11).

According to this, a new point $\mathbf{x} \in \Re^n$ belongs to the class k^* iff

$$k^* = \arg\min_{k=1,2} \frac{|\mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{s}_k + b_k|}{\sqrt{\mathbf{s}_k^\top \mathbf{K}\mathbf{s}_k}}, \tag{38}$$

where $\mathbf{K} = [\mathbf{K}_{11}, \mathbf{K}_{12}; \mathbf{K}_{21}, \mathbf{K}_{22}] \in \Re^{m \times m}$.

4 Experimental results

We applied the proposed and alternative approaches to the following seven well-known benchmark data sets from the UCI Repository [3]: Australian Credit (AUS), Wisconsin Breast Cancer (WBC), BUPA Liver (LIVER), German

Table 2 Predictive performance summary for all linear approaches and for all datasets

	AUS	WBC	LIVER	GER	DIA	HEART	IONO
SVM$_l$	86.2	97.3	51.5	69.4	72.1	50.8	93.2
TBSVM$_l$	86.7	96.8	65.9	72.2	73.4	85.0	85.2
SOCP-SVM$_l$	86.8	96.5	63.9	72.2	74.9	84.7	86.1
Twin SOCP-SVM$_l$	87.0	98.0	68.1	72.9	76.1	85.5	81.4

Credit (GER), Pima Indians Diabetes (DIA), Heart/Statlog (HEART), and Ionosphere (IONO). Table 1 summarizes the relevant information for each benchmark data set, including the number of variables, the sample size, the percentage of observations in each class, and the imbalance ratio (IR). More information on these datasets can be found in the UCI Repository [3].

The following classification approaches are studied and reported:

- Standard SVM, linear (SVM$_l$, Formulation (1)) and kernel-based version (SVM$_K$, Formulation (2)).
- Twin-Bounded SVM, linear (TBSVM$_l$, Formulation (5)–(6)) and kernel-based version (TBSVM$_K$, Formulation (12)–(13)).
- SOCP-SVM, linear (SOCP-SVM$_l$, Formulation (15)) and kernel-based version (SOCP-SVM$_K$, Formulation (18)).
- The proposed Twin SOCP-SVM method, linear (Twin SOCP-SVM$_l$, Formulation (19)–(20)) and kernel-based version (Twin SOCP-SVM$_K$, Formulation (35)–(36)).

The following model selection procedure was performed: training and test subsets were constructed using 10-fold cross-validation for all the datasets. We used the metric area under the curve (AUC) as the main performance measure,which is arguably the most commonly used metric for model comparison in the machine learning community [30].

A grid search was performed for SVM parameters C and σ; Twin SVM parameter $c_i,\ i = \{1, 2, 3, 4\}$; SOCP parameters η_k; and parameter θ_k used in the proposed approach. We studied the following values of $\eta_k \in \{0.2, 0.4, 0.6, 0.8\}$. We used the following set of values for parameters C, c_i, θ_k and σ:

$$C,\ c_i,\ \theta_i,\ \sigma \in \{2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, \ldots, 2^3, 2^4, 2^5, 2^6, 2^7\}.$$

For this procedure, we used LIBSVM for Matlab [7] for standard SVM approaches, the SeDuMi Matlab Toolbox for SOCP-based classifiers [31], and the codes provided by Yuan-Hai Shao, author of Twin-Bounded SVM [28], which are publicly available in http://www.optimal-group.org/.

Tables 2 and 3 summarize the best performance (in terms of AUC) of all the techniques. In Table 2, we present the results of the linear approaches for all seven data sets, while the results of the kernel-based methods are presented in Table 3. For each table, the best method is emphasized in bold type.

In Table 2 we observe that the best predictive results were achieved using the proposed Twin SOCP-SVM in six out of seven datasets, while standard SVM had better AUC in one dataset (Ionosphere). The methods TBSVM, SOCP-SVM, and Twin SOCP-SVM have relatively similar performances in all datasets, outperforming standard SVM in LIVER and HEART. In such data, standard SVM fails at finding an adequate classifier, leading to poor predictive performance.

It can be seen in Table 3 that no method outperformed the others in all the experiments, although the proposed method performs better on four out of seven datasets. The differences in terms of AUC are not conclusive in most cases.

We used the robustness analysis procedure proposed in [11] to assess the overall performance of our approach. The relative performance of a given method on a dataset is represented by the ratio between its AUC and the highest among all the compared strategies. Formally, the AUC ratio for method M and dataset i is:

$$AUCRatio_i(M) = \frac{AUC(M)}{max_j\ AUC(j)}, \tag{39}$$

where $AUC(j)$ is the AUC for method j when trained on dataset i. The larger the value of $AUCRatio_i(M)$, the better the performance of M in dataset i. The best method M^* will have $AUCRatio_i(M^*) = 1$ for dataset i. The value of $\sum_i AUCRatio_i(M)$ represents a measure of robustness and overall performance for an algorithm M, and the larger its value, the better the overall performance and robustness [11]. Figure 3 presents the distribution of $AUCRatio_i(a)$ for the four methods and all datasets. For each method we selected the best performance in terms of AUC between its linear and kernel-based versions.

In Figure 3 we observe that Twin SOCP-SVM has the best overall performance, being very close to the optimal performance measure of 7 (6.98). Comparing the different classification strategies (SVM, TBSVM, SOCP-SVM, and Twin SOCP-SVM), standard SVM has the lowest overall performance. We conclude that the proposed method represents an excellent alternative for the classification task, since it achieves best overall performance while reducing the size of a SOCP problem by splitting it into two

Table 3 Predictive performance summary for all kernel-based approaches and for all datasets

	AUS	WBC	LIVER	GER	DIA	HEART	IONO
SVM$_K$	86.2	97.1	73.3	68.8	72.1	79.4	94.1
TBSVM$_K$	87.6	97.0	65.0	72.4	75.6	62.3	95.4
SOCP-SVM$_K$	86.9	97.4	72.9	72.2	76.3	79.5	95.2
Twin SOCP-SVM$_K$	87.3	97.7	72.4	73.1	76.5	76.3	95.4

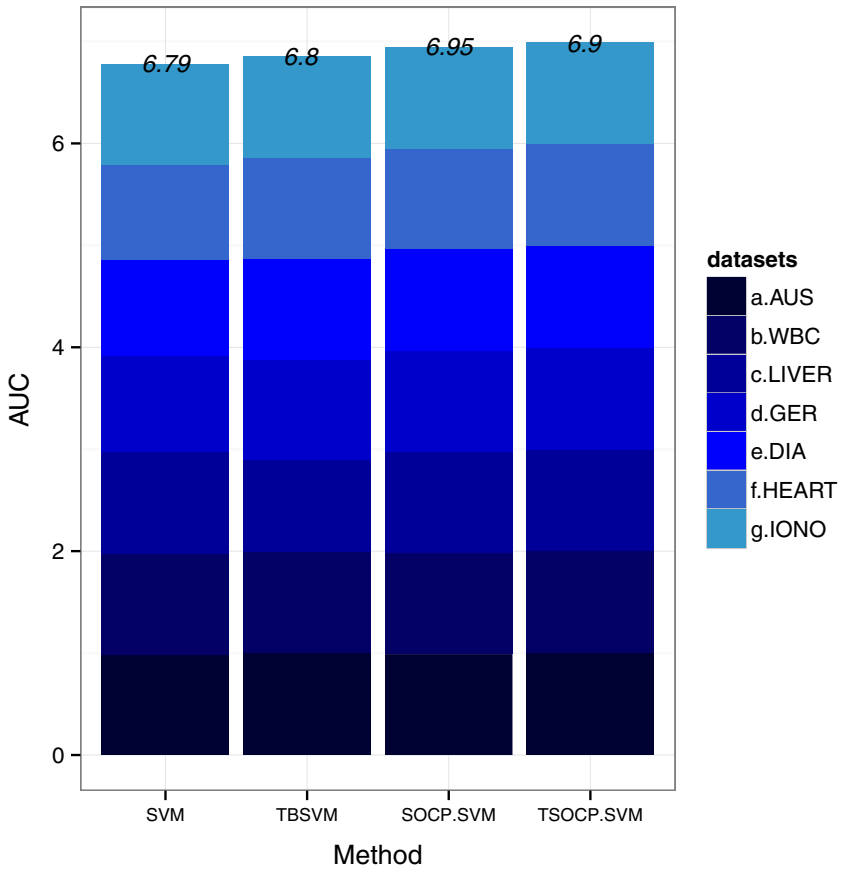

Fig. 3 Sum of AUC ratios for all methods

smaller SOCP problems, instead of one, and creating two non-parallel hyperplanes, one for each training pattern.

5 Conclusions

In this paper, we present a novel classification approach, which extends the ideas of Twin SVM [14, 28] to second-order cones. The method is presented as a linear classifier, and subsequently extended to a kernel-based method. The dual form of the method is also computed, and some interesting geometrical properties are discussed.

SOCP formulations for SVM have the ability of generalizing training patterns effectively by considering a robust setting for data distribution [19, 24]. The design benefits the correct prediction of both classes. This fact is demonstrated empirically on seven benchmark datasets, where best overall results in terms of AUC are achieved by the proposed method. Additionally, the proposal represents an improvement on the state-of-the-art of SOCP formulations for SVM, since it reduces the size of the original model by splitting it into two smaller problems, leading to a reduction in terms of computational times, and providing the opportunity to solve larger SOCP problems. This is an important point since the complexity of SOCP formulations is higher than QP formulations [17], and there are no efficient solvers designed for SOCP-SVM, in contrast to standard SVM (LIBSVM [7], for example).

We identified the following opportunities for future research:

- There is a need for more efficient implementations for SOCP-based SVM formulations. Although several techniques have been suggested for solving SVM efficiently [16], none of these methods has been adapted for SOCP-based SVM. Such implementations would allow the construction of classifiers in large scale datasets. SOCP-based SVM formulations will become a real alternative to traditional SVM when this can be achieved. This work presents an interesting step in that direction, dividing the original formulation into two smaller problems. The previously developed r-SOCP-SVM method [19] also follows the same path, on which one of the three SOC constraints is removed, reducing the running times compared to the original SOCP-SVM formulation.
- The proposal can be extended to multi-class classification, exploiting some of the properties of multi-category Twin SVM [33]. Although some efforts have already been made to extend SOCP-SVM to multi-class (see e.g. [6]), there are interesting research opportunities for

multi-class formulations due to their vast application domains.

– The SOCP-SVM method has a balanced design since each constraint corresponds to a particular training pattern that should be correctly classified up to a rate η, making it especially suitable for class-imbalanced classification [20]. A previous work extends SOCP-SVM method for this task, and the Twin SOCP-SVM can also be extended to deal with skewed labels.

Appendix: A Dual formulation for twin SOCP-SVM

A.1 Proof of Theorem 1

Proof The Lagrangian function associated with Problem (19) is given by

$$L(\mathbf{w}_1, b_1, \lambda) = \frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta_1}{2}\left(\|\mathbf{w}_1\|^2 + b_1^2\right) + \lambda(\mathbf{w}_1^\top \boldsymbol{\mu}_2 + b_1 + 1 + \kappa_2\|S_2^\top \mathbf{w}_1\|),$$

where $\lambda \geq 0$. Since $\|\mathbf{v}\| = \max_{\|\mathbf{u}\|\leq 1} \mathbf{u}^\top \mathbf{v}$ holds for any $\mathbf{v} \in \Re^n$, we can rewrite the Lagrangian as follows:

$$L(\mathbf{w}_1, b_1, \lambda) = \max_{\mathbf{u}}\{L_1(\mathbf{w}_1, b_1, \lambda, \mathbf{u}) : \|\mathbf{u}\| \leq 1\},$$

with L_1 given by

$$L_1(\mathbf{w}_1, b_1, \lambda, \mathbf{u}) = \frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta_1}{2}\left(\|\mathbf{w}_1\|^2 + b_1^2\right) + \lambda(\mathbf{w}_1^\top \boldsymbol{\mu}_2 + b_1 + 1 + \kappa_2 \mathbf{w}_1^\top S_2 \mathbf{u}). \quad (A.1)$$

Thus, problem (19) can be written equivalently as:

$$\min_{\mathbf{w}_1, b_1} \max_{\mathbf{u}, \lambda}\{L_1(\mathbf{w}_1, b_1, \lambda, \mathbf{u}) : \|\mathbf{u}\| \leq 1, \lambda \geq 0\}.$$

Hence, the dual problem of (19) is given by:

$$\max_{\mathbf{u}, \lambda} \min_{\mathbf{w}_1, b_1} \{L_1(\mathbf{w}_1, b_1, \lambda, \mathbf{u}) : \|\mathbf{u}\| \leq 1, \lambda \geq 0\}. \quad (A.2)$$

The above expression allows the construction of the dual formulation. The detailed description of this procedure can be found in [22]. The computation of the first order condition for the inner optimization task (the minimization problem) yields to

$$\nabla_{\mathbf{w}_1} L_1 = A^\top(A\mathbf{w}_1 + \mathbf{e}_1 b_1) + \theta_1 \mathbf{w}_1 + \lambda(\boldsymbol{\mu}_2 + \kappa_2 S_2 \mathbf{u}) = 0, \quad (A.3)$$

$$\nabla_{b_1} L_1 = \mathbf{e}_1^\top(A\mathbf{w}_1 + \mathbf{e}_1 b_1) + \theta_1 b_1 + \lambda = 0. \quad (A.4)$$

Let us denote by $\hat{\mathbf{z}} = [\mathbf{z}^\top, 1]^\top \in \Re^{n+1}$, with $\mathbf{z} = \boldsymbol{\mu}_2 + \kappa_2 S_2 \mathbf{u} \in \Re^n$. Then the relations (A.3)–(A.4) can be written compactly as

$$(H^\top H + \theta_1 I)\mathbf{v}_1 + \lambda \hat{\mathbf{z}} = 0.$$

Since the symmetric matrix $(H^\top H + \theta_1 I)$ is positive definite, for any $\theta_1 > 0$, one has

$$\mathbf{v}_1 = -\lambda(H^\top H + \theta_1 I)^{-1}\hat{\mathbf{z}}. \quad (A.5)$$

On the other hand, by replacing (A.3)–(A.4) in (A.1) and using the relations (21) and (A.5), the dual problem can be stated as follows:

$$\begin{array}{l} \max_{\mathbf{z},\mathbf{u},\lambda} -\frac{1}{2}\lambda^2 \hat{\mathbf{z}}^\top (H^\top H + \theta_1 I)^{-1}\hat{\mathbf{z}} + \lambda \\ \text{s.t. } \mathbf{z} = \boldsymbol{\mu}_2 + \kappa_2 S_2 \mathbf{u}, \ \|\mathbf{u}\| \leq 1, \\ \lambda \geq 0. \end{array} \quad (A.6)$$

Notice that the objective function of the dual problem (A.6) is concave with respect to λ, and it attains its maximum value at

$$\lambda^* = \frac{1}{\hat{\mathbf{z}}^\top (H^\top H + \theta_1 I)^{-1}\hat{\mathbf{z}}}, \quad (A.7)$$

with optimal value

$$\frac{1}{2}\frac{1}{\hat{\mathbf{z}}^\top (H^\top H + \theta_1 I)^{-1}\hat{\mathbf{z}}}.$$

Then, by using (A.7) the dual problem of (19) can be stated as follows:

$$\begin{array}{ll} \min_{\mathbf{z},\mathbf{u}} & \frac{1}{2}\begin{pmatrix}\mathbf{z}^\top & 1\end{pmatrix}(H^\top H + \theta_1 I)^{-1}\begin{pmatrix}\mathbf{z} \\ 1\end{pmatrix} \\ \text{s.t.} & \mathbf{z} \in \mathbf{B}(\boldsymbol{\mu}_2, S_2, \kappa_2), \end{array} \quad (A.8)$$

where

$$\mathbf{B}(\boldsymbol{\mu}, S, \kappa) = \{\mathbf{z} \in \Re^n : \mathbf{z} = \boldsymbol{\mu} + \kappa S\mathbf{u}, \|\mathbf{u}\| \leq 1\}.$$

Similarly, since the symmetric matrix $(G^\top G + \theta_2 I)$ is positive definite, for any $\theta_2 > 0$, one can prove that the dual of the problem (20) is given by:

$$\begin{array}{ll} \min_{\mathbf{p},\mathbf{u}} & \frac{1}{2}\begin{pmatrix}\mathbf{p}^\top & 1\end{pmatrix}(G^\top G + \theta_2 I)^{-1}\begin{pmatrix}\mathbf{p} \\ 1\end{pmatrix} \\ \text{s.t.} & \mathbf{p} \in \mathbf{B}(\boldsymbol{\mu}_1, S_1, \kappa_1). \end{array} \quad (A.9)$$

□

A.2 Proof of Proposition 1

Proof Let us denote the objective function of Problem (26) by

$$f(\mathbf{z}) = \frac{1}{2}\begin{pmatrix}\mathbf{z}^\top & 1\end{pmatrix}(H^\top H + \theta_1 I)^{-1}\begin{pmatrix}\mathbf{z} \\ 1\end{pmatrix}.$$

Since the symmetric matrix $H^\top H + \theta_1 I = \begin{pmatrix} A^\top A + \theta_1 I & A^\top \mathbf{e}_1 \\ \mathbf{e}_1^\top A & \mathbf{e}_1^\top \mathbf{e}_1 + \theta_1 \end{pmatrix}$ is positive definite for each $\theta_1 > 0$, where $H = [A \ \mathbf{e}_1] \in \mathbb{R}^{m_1 \times (n+1)}$ (cf. (23)), Theorem 7.7.6 of [13] implies that the matrix

$C_s(\theta_1) = A^\top A + \theta_1 I - \frac{1}{m_1+\theta_1} A^\top \mathbf{e}_1 \mathbf{e}_1^\top A$ is invertible, and that

$$(H^\top H + \theta_1 I)^{-1} = \begin{pmatrix} I & 0 \\ -\frac{1}{m_1+\theta_1}\mathbf{e}_1^\top A & 1 \end{pmatrix} \begin{pmatrix} C_s(\theta_1)^{-1} & 0 \\ 0 & \frac{1}{m_1+\theta_1} \end{pmatrix} \times \begin{pmatrix} I & -\frac{1}{m_1+\theta_1} A^\top \mathbf{e}_1 \\ 0 & 1 \end{pmatrix}. \quad \text{(A.10)}$$

By using the first equality of (16), and making the product of the matrices we have that

$$f(\mathbf{z}) = \frac{1}{2}\left(\left(\mathbf{z}^\top - \frac{m_1}{m_1+\theta_1}\hat{\boldsymbol{\mu}}_1^\top\right) C_s(\theta_1)^{-1} \times \left(\mathbf{z} - \frac{m_1}{m_1+\theta_1}\hat{\boldsymbol{\mu}}_1\right) + \frac{1}{m_1+\theta_1}\right).$$

Thus, (31) follows. Formulation (32) can be derived in a similar way.

Now, we suppose that $\theta_1 = 0$ and that $H^\top H$ is positive definite. Since

$$C_s(0) = A^\top \left(I - \frac{1}{m_1}\mathbf{e}_1\mathbf{e}_1^\top\right) A = A^\top \left(I - \frac{1}{m_1}\mathbf{e}_1\mathbf{e}_1^\top\right) \times \left(I - \frac{1}{m_1}\mathbf{e}_1\mathbf{e}_1^\top\right) A = m_1 S_1 S_1^\top = m_1 \hat{\Sigma}_1,$$

where the last two equalities follow from (16) and (17), the expression (A.10) can be written as

$$(H^\top H)^{-1} = \frac{1}{m_1}\begin{pmatrix} I & 0 \\ -\hat{\boldsymbol{\mu}}_1^\top & 1 \end{pmatrix}\begin{pmatrix} \hat{\Sigma}_1^{-1} & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} I & -\hat{\boldsymbol{\mu}}_1 \\ 0 & 1 \end{pmatrix}.$$

Then,

$$f(\mathbf{z}) = \frac{1}{2m_1}\left((\mathbf{z}^\top - \hat{\boldsymbol{\mu}}_1^\top)\hat{\Sigma}_1^{-1}(\mathbf{z} - \hat{\boldsymbol{\mu}}_1) + 1\right) = \frac{1}{2m_1}\left(\|\hat{\Sigma}_1^{-1/2}(\mathbf{z} - \hat{\boldsymbol{\mu}}_1)\|^2 + 1\right).$$

Hence, (33) holds. Formulation (34) can be derived in a similar way. □

Acknowledgments The first author was supported by FONDECYT project 1140831, the second was funded by CONICYT Anillo ACT1106, and third author was supported by FONDECYT project 1130905. Support from the Chilean "Instituto Sistemas Complejos de Ingeniería" (ICM: P-05-004-F, CONICYT: FB016, www.sistemasdeingenieria.cl) is greatly acknowledged.

References

1. Alizadeh F, Goldfarb D (2003) Second-order cone programming. Math Program 95:3–51
2. Alvarez F, López J, Ramírez CH (2010) Interior proximal algorithm with variable metric for second-order cone programming: applications to structural optimization and support vector machines. Optimization Methods Software 25(6):859–881
3. Bache K, Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml
4. Bai L, Wang Z, Shao YH, Deng NY (2014) A novel feature selection method for twin support vector machine. Knowl-Based Syst 59(0):1–8
5. Bennett K, Bredensteiner E (2000) Duality and geometry in svm classifiers. In: In Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, pp 57–64
6. Bosch P, López J, Ramírez H, Robotham H (2013) Support vector machine under uncertainty: An application for hydroacoustic classification of fish-schools in chile. Expert Syst Appl 40(10):4029–4034
7. Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2:1–27. software available at, http://www.csie.ntu.edu.tw/cjlin/libsvm
8. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
9. De Maesschalck R, Jouan-Rimbaud D, Massart D (2000) The mahalanobis distance. Chemom Intell Lab Syst 50:1–18
10. Debnath R, Muramatsu M, Takahashi H (2005) An efficient support vector machine learning method with second-order cone programming for large-scale problems. Appl Intell 23:219–239
11. Geng X, Zhan DC, Zhou ZH (2005) Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Trans Syst Man Cybern B Cybern 35(6):1098–1107
12. Goldfarb D, Iyengar G (2003) Robust convex quadratically constrained programs. Math Program 97(3):495–515
13. Horn RA, Johnson CR (1990) Matrix Analysis, 1st edn. Cambridge University Press, New York
14. Jayadeva KhemchandaniR, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
15. Lanckriet G, Ghaoui L, Bhattacharyya C, Jordan M (2003) A robust minimax approach to classification. J Mach Learn Res 3:555–582
16. Li C, Liu K, Wang H (2011) The incremental learning algorithm with support vector machine based on hyperplane-distance. Appl Intell 34:19–27
17. Lobo M, Vandenberghe L, Boyd S, Lebret H (1998) Applications of second-order cone programming. Linear Algebra Appl 284:193–228
18. López J, Maldonado S, Carrasco M (2015) A novel multi-class svm model using second-order cone constraints. Applied Intelligence In Press
19. Maldonado S, López J (2014a) Alternative second-order cone programming formulations for support vector classification. Inform Sci 268:328–341
20. Maldonado S, López J (2014b) Imbalanced data classification using second-order cone programming support vector machines. Pattern Recogn 47:2070–2079
21. Maldonado S, Famili F, Weber R (2014) Feature selection for high-dimensional class-imbalanced data sets using support vector machines. Inform Sci 286:228–246
22. Mangasarian OL (1994) Nonlinear Programming. Classics in Applied Mathematics, Society for Industrial and Applied Mathematics
23. Mercer J (1909) Functions of positive and negative type, and their connection with the theory of integral equations. Philos Trans R Soc Lond A 209:415–446
24. Nath S, Bhattacharyya C (2007) Maximum margin classifiers with specified false positive and false negative error rates. In: Proceedings of the SIAM International Conference on Data mining
25. Peng X (2011) Building sparse twin support vector machine classifiers in primal space. Inform Sci 181(18):3967–3980
26. Qi Z, Tian Y, Shi Y (2013) Robust twin support vector machine for pattern classification. Pattern Recogn 46(1):305–316

27. Schölkopf B, Smola AJ (2002) Learning with Kernels. MIT Press
28. Shao Y, Zhang C, Wang X, Deng N (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–68
29. Shao YH, Chen WJ, Deng NY (2014) Nonparallel hyperplane support vector machine for binary classification problems. Inform Sci 263(1):22–35
30. Sokolova M, Japkowicz N (2006) Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. In: Advances in Artificial Intelligence. Springer, Berlin Heidelberg, pp 1015–1021
31. Sturm J (1999) Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optimization Methods and Software 11(12):625–653. special issue on Interior Point Methods (CD supplement with software)
32. Trafalis TB, Alwazzi SA (2010) Support vector machine classification with noisy data: a second order cone programming approach. Int J Gen Syst 39(7):757–781
33. Xie J, Hone K, Xie W, Gao X, Shi Y, Liu X (2013) Extending twin support vector machine classifier for multi-category classification problems. Intelligent Data Analysis 17(4):649–664
34. Zhong P, Fukushima M (2007) Second-order cone programming formulations for robust multiclass classification. Neural Comput 19:258–282

Sebastián Maldonado received his B.S. and M.S. degree from the University of Chile, in 2007, and his Ph.D. degree from the University of Chile, in 2011. He is currently Assistant Professor at the School of Engineering and Applied Sciences, Universidad de los Andes, Santiago, Chile. His research interests include statistical learning, data mining and business analytics.

Julio López received his B.S. degree in Mathematics in 2000 from the University of Trujillo, Perú. He also received the M.S. degree in Sciences in 2003 from the University of Trujillo, Perú and the Ph.D. degree in Engineering Sciences, minor Mathematical Modelling in 2009 from the University of Chile. Currently, he is an assistant Professor of Institute of Basic Sciences at the University Diego Portales, Santiago, Chile. His research interests include conic programming, convex analysis, algorithms and machine learning.

Miguel Carrasco received his B.S. degree in Mathematics in 2002 and the B.S. degree in Computing Sciences in 2005 from the University of Chile. He also received the Ph.D. degree in Engineering Sciences, minor Mathematical Modelling in 2007 from the University of Chile in collaboration with University of Montpellier II, France. Currently, he is full time professor at the School of Engineering and Applied Sciences, Universidad de los Andes, Santiago, Chile. His research interests include convex analysis, proximal type algorithms, conic programming and topology optimization.