Appl Intell (2017) 46:983–992
DOI 10.1007/s10489-016-0881-0

Robust kernel-based multiclass support vector machines via second-order cone programming

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

Published online: 6 January 2017

Abstract Kernel methods are very important in pattern analysis due to their ability to capture nonlinear relationships in datasets. The best known kernel-based technique is Support Vector Machine (SVM), which can be used for several pattern recognition tasks, including multiclass classification. In this paper, we focus on maximum margin classifiers for nonlinear multiclass learning, based on second-order cone programming (SOCP), proposing three novel formulations that extend the most common strategies for this task: One-vs.-The-Rest, One-vs.-One, and All-Together optimization. The proposed SOCP formulations achieved superior performance compared to their traditional SVM counterparts on benchmark datasets, demonstrating the virtues of robust optimization.

Keywords Multiclass classification · Second-order cone programming · Kernel methods · Support vector machines.

1 Introduction

Multiclass classification solves the problem of predicting more than two classes, where each data point can be assigned to only one of them. This is an important task in artificial intelligence, with broad applications such as text classification [19], biotechnology (DNA microarray analysis of multiple tumor types [20]), and business analytics (credit assignment based on two types of defaulters in addition to the good payers: those who cannot pay due to the lack of cash, and those who do not have the willingness to pay [7]).

Support Vector Machine (SVM) [36] is one of the standard tools for multiclass classification. A series of binary classifiers can be constructed for this task [6], although it can also be tackled directly by solving a single multiclass SVM [8, 13, 38]. SVM has proved to be very effective for multiclass learning thanks to the use of kernel functions [38]. These functions project the data points onto a high-dimensional feature space, resulting in nonlinear classifiers.

Second-order cone programming (SOCP) formulations have been proposed as robust settings for maximum margin classifiers [2, 4, 29]. These strategies assume the worst data distribution for a given mean and covariance matrix, and aim at classifying each training pattern correctly for specified false positive and false negative error rates.

In our work, we extend the SOCP formulation for binary classification proposed by Nath and Bhattacharyya [29] to kernel-based multiclass classification. It is important to note that Nath and Bhattacharyya's work differs from the SOCP-SVM formulations proposed to deal with noisy data (i.e. instances with measurement errors [33, 41]), and SOCP formulations that solve the standard SVM model based on reduced convex hulls [9]. We previously proposed a multiclass SOCP formulation based on the concept of the center of the configuration [23], which corresponds to a point equidistant to all training patterns. The method proposed in

✉ Sebastián Maldonado
smaldonado@uandes.cl

Julio López
julio.lopez@udp.cl

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

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

this work follows a different strategy: all hyperplanes are constructed according to the ideas Weston et al. [38] and Bredensteiner and Bennett [8], i.e., in such a way that the training points from class k should be closer to the k-th hyperplane rather than to a classifier conformed by the $k-1$ other classes.

The paper is structured as follows: Section 2 discusses previous work on SVM for kernel-based multiclass classification. Section 3 describes the work of Nath and Bhattacharyya [29] for (linear) binary classification and its extension to multiclass classification that we proposed in López and Maldonado [21]. The proposed methods for kernel-based multiclass SVM via SOCP are introduced in Section 4. Section 5 provides experimental results using benchmark datasets. We then present the main conclusions of this study in Section 6 and address possible future developments.

2 Kernel-based multi-class support vector machine

In this section, we provide a brief description of the three best known SVM approaches for kernel-based multiclass classification, namely One-vs.-The-Rest, One-vs.-One, and All-Together multiclass SVM. Additionally, we include recent developments in kernel-based multiclass SVM for comparison purposes, such as One-vs.-The-Rest twin SVM [40], Adaptive Multi-Hyperplane Machine (AMM), and Budgeted Stochastic Gradient Descent (BSGD) [10]. The latter two methods are highly-optimized SVM implementations designed to achieve reduced training times.

2.1 One-vs.-the-rest SVM

Considering training examples $\mathbf{x}_i \in \Re^n$, $i = 1, \dots, m$, and their respective labels $y_i \in \{1, \dots, K\}$, One-vs.-The-Rest SVM (OvR-SVM) constructs K hyperplanes of the form $f_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + b_k$ such that each training sample has to be classified correctly into class k or a second group of instances made up of all the remaining classes except k, $k = 1, \dots, K$ [6, 36]. This hyperplane maximizes the *margin*, which is computed as the sum of the distances to the closest points of each of the two new classes. The maximization of this margin is equivalent to minimizing the Euclidean norm of $\mathbf{w}_k$. The label vector can be redefined as $y_i^k \in \{-1, 1\}$, where 1 is used for the samples from class k and -1 for the elements of other classes of the $K-1$, for each $k = 1, \dots, K$.

Nonlinear classifiers can be obtained by mapping the data samples onto a higher dimensional space via a kernel function. The kernel-based OvR-SVM formulation for the k-th class can be stated as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha}^k} \quad & \sum_{i=1}^m \alpha_i^k - \frac{1}{2} \sum_{i,s=1}^m \alpha_i^k \alpha_s^k y_i^k y_s^k K(\mathbf{x}_i, \mathbf{x}_s) \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i^k y_i^k = 0, \\ & 0 \le \alpha_i^k \le C, \qquad i = 1, \dots, m. \end{aligned} \tag{1}$$

We based our analysis on the *Gaussian kernel*, which usually achieves the best empirical performance [25, 32], and has the following form:

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

where $\sigma > 0$ is a parameter that controls the width of the kernel [32]. Once all K hyperplanes are constructed, the decision function is given by $f_k(\mathbf{x}) = \sum_{i=1}^m \alpha_i^k y_i^k K(\mathbf{x}, \mathbf{x}_i) + b_k$. Then, a new sample $\mathbf{x}$ is classified into the class with the greatest value of $f_k(\mathbf{x})$.

2.2 One-vs.-one SVM

Another well-known classification approach is One-versus-One (OvO) SVM [17]. This method constructs $K(K-1)/2$ hyperplanes, one for each pair of classes. The following problem is solved for the k-th and the l-th classes ($k < l$):

$$\begin{aligned} \max_{\boldsymbol{\alpha}^{kl}} \quad & \sum_{i=1}^m \alpha_i^{kl} - \frac{1}{2} \sum_{i,s=1}^m \alpha_i^{kl} \alpha_s^{kl} y_i^{kl} y_s^{kl} K(\mathbf{x}_i, \mathbf{x}_s) \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i^{kl} y_i^{kl} = 0, \\ & 0 \le \alpha_i^{kl} \le C, \qquad i = 1, \dots, m, \end{aligned} \tag{3}$$

where $y_i^{kl} = 1$ means the sample belongs to the class k, while $y_i^{kl} = -1$ represents the opposite case (class l). Once all classifiers are constructed, the decision rule for a new sample $\mathbf{x}$ is given by $f_{kl}(\mathbf{x}) = \sum_{i=1}^m \alpha_i^{kl} y_i^{kl} K(\mathbf{x}, \mathbf{x}_i) + b_{kl}$. The Max-Wins voting strategy is usually used [11], in which each hyperplane assigns the samples to one of the two corresponding classes, increasing the vote by one for the assigned class. A majority vote scheme finally determines the label of each new instance.

2.3 All-together multiclass SVM

Multiclass SVM can also be performed by solving a single optimization problem, as proposed in Weston et al. [38]

or Bredensteiner and Bennett [8]. The first approach, called MC-SVM, constructs K classifiers simultaneously, solving the following formulation:

$$\min_{\boldsymbol{\alpha}^k} \quad \sum_{i,s=1}^{m} \left(\frac{1}{2} c_s^{y_i} a_i a_s - \sum_{k=1}^{K} \alpha_i^k \alpha_s^{y_i} + \frac{1}{2} \sum_{k=1}^{K} \alpha_i^k \alpha_s^k \right)$$
$$K(\mathbf{x}_i, \mathbf{x}_s) - 2 \sum_{i=1}^{m} \sum_{k=1}^{K} \alpha_i^k$$
$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i^k = \sum_{i=1}^{m} c_i^k a_i, \quad k = 1, \ldots, K,$$
$$0 \leq \alpha_i^k \leq C, \quad i = 1, \ldots, m, \; k = 1, \ldots, K,$$
$$\alpha_i^{y_i} = 0, \quad i = 1, \ldots, m, \tag{4}$$

where

$$a_i = \sum_{k=1}^{K} \alpha_i^k, \quad c_i^k = \begin{cases} 1, & \text{if } y_i = k \\ 0, & \text{if } y_i \neq k \end{cases}.$$

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

$$k^* = \underset{k=1,\ldots,K}{\operatorname{argmax}} \left\{ \sum_{i=1}^{m} (c_i^k a_i - \alpha_i^k) K(\mathbf{x}_i, \mathbf{x}) + b_k \right\}.$$

2.4 One-vs.-the-rest twin support vector machine

The OvR twin SVM extends the ideas of the traditional twin SVM (TWSVM) for binary classification proposed by Jayadeva [16] by solving K quadratic programming problems (QPPs), one for each class [40]. Each QPP constructs two nonparallel hyperplanes in such a way that each function is as close as possible to one of the two classes, and as far as possible from the other class, under the One-vs.-The-Rest framework. Each of the K problems solved by OvR twin SVM has the following form:

$$\begin{aligned} \min_{\mathbf{s}_k, b_k, \boldsymbol{\xi}} \quad & \tfrac{1}{2} \| K(A^k, \mathbb{X}) \mathbf{s}_k + b_k \mathbf{e}_k \|^2 + c \, \tilde{\mathbf{e}}_k^\top \boldsymbol{\xi} \\ \text{s.t.} \quad & -(K(\tilde{A}^k, \mathbb{X}) \mathbf{s}_k + \tilde{\mathbf{e}}_k b_k) + \boldsymbol{\xi} \geq \tilde{\mathbf{e}}_k \\ & \boldsymbol{\xi} \geq 0, \end{aligned} \tag{5}$$

where $A^k \in \Re^{n \times m_k}$ and $\tilde{A}^k \in \Re^{n \times m - m_k}$ represent the data matrices for class k and for the remaining classes, respectively; $\mathbb{X} = [A^1 \, A^2 \, \ldots \, A^K] \in \Re^{n \times m}$, c is a positive parameter; and $\mathbf{e}_k$ and $\tilde{\mathbf{e}}_k$ are vectors of ones of appropriate dimensions. A new sample $\mathbf{x} \in \Re^n$ belongs to a given class k^* iff $k^* = \operatorname{argmin}_{k=1,\ldots,K} \{K(\mathbf{x}, \mathbb{X}) \mathbf{s}_k + b_k\}$, where

$$K(\mathbf{x}, \mathbb{X}) = [K(\mathbf{x}, \mathbb{X}_{\bullet 1}), K(\mathbf{x}, \mathbb{X}_{\bullet 2}), \ldots, K(\mathbf{x}, \mathbb{X}_{\bullet m})], \tag{6}$$

with $\mathbb{X}_{\bullet j}$ denoting the j-th column of the matrix $\mathbb{X}$.

2.5 Optimized approximations for efficient SVM classification

Several strategies have been proposed to speed up the training process for SVM classification. In particular, we used the Adaptive Multi-Hyperplane Machine (AMM) and the Budgeted Stochastic Gradient Descent (BSGD) for benchmarking purposes. These two methods are designed to construct nonlinear decision boundaries, being suitable for benchmarking kernel-based approaches.

The AMM method constructs multiple linear classifiers in order to approximate a nonlinear function, while the BSGD method incrementally updates the support vectors via stochastic gradient descent, while fixing the cardinality of support vectors in the model. This latter method constructs a nonlinear classifier using a Gaussian kernel [10].

3 Maximum margin classifiers based on second-order cone programming

In this section, we describe the SOCP formulation for maximum margin (binary) classification proposed by Nath and Bhattacharyya [29], and formalize the One-vs.-The-Rest, One-vs.-One, and All-Together extensions for linear multiclass classification. The first two formulations (OvR and OvO) were used previously in López and Maldonado [21] in the context of feature selection for microarray classification, but only as linear classifiers. The All-Together method for linear SOCP classification was proposed in López and Maldonado [22].

3.1 SOCP formulation for binary classification

Let us consider $\mathbf{X}_k$ as a random variable that generates the samples of the class k, with mean and covariance given by $(\boldsymbol{\mu}_k, \Sigma_k)$ for $k = 1, 2$. Assuming specified false-negative and false-positive errors $1 - \eta_k$, with $\eta_k \in (0, 1)$, a linear classifier can be constructed by requiring that the accuracy for class k should be at least η_k. Nath and Bhattacharyya [29] suggested the following quadratic chance-constrained programming model:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \tfrac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \Pr\{\mathbf{w}^\top \mathbf{X}_1 + b \geq 1\} \geq \eta_1, \\ & \Pr\{\mathbf{w}^\top \mathbf{X}_2 + b \leq -1\} \geq \eta_2. \end{aligned} \tag{7}$$

According to the authors, the probabilistic constraints can be replaced in (7) with their robust counterparts:

$$\inf_{\mathbf{X}_1\sim(\boldsymbol{\mu}_1,\Sigma_1)} \Pr\{\mathbf{w}^\top\mathbf{X}_1 + b \geq 1\} \geq \eta_1, \inf_{\mathbf{X}_2\sim(\boldsymbol{\mu}_2,\Sigma_2)} \Pr\{\mathbf{w}^\top\mathbf{X}_2 + b \leq -1\} \geq \eta_2. \quad (8)$$

The intuition behind this step is classifying each class correctly even for the worst data distribution [37]. Applying the multivariate Chebyshev inequality [18, Lemma 1], the constraints in (8) are equivalent to:

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

where $\kappa_i = \sqrt{\frac{\eta_k}{1-\eta_k}}$, for $k = 1, 2$. The Chebyshev inequality provides a bound that holds for a family of distributions having the similar mean and covariance, and the worst case corresponds to the case of equality for this bound [33]. Replacing these constraints in Formulation (7) leads to the following quadratic SOCP problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq 1 + \kappa_1\|S_1^\top\mathbf{w}\|, \\ & -\mathbf{w}^\top\boldsymbol{\mu}_2 - b \geq 1 + \kappa_2\|S_2^\top\mathbf{w}\|, \end{aligned} \quad (9)$$

where $\Sigma_k = S_k S_k^\top$, for $k = 1, 2$. This decomposition can be performed, for example, via Cholesky factorization. Problem (9) is a convex formulation with a quadratic objective function and two second-order cone (SOC) constraints [1].

3.2 One-vs.-the-rest SOCP, linear version

The previous formulation can be extended easily to OvR classification. The following quadratic chance-constrained programming problem is proposed in López and Maldonado [21, 22] for each class $k = 1, \ldots, K$:

$$\begin{aligned} \min_{\mathbf{w}_k,b_k} \quad & \tfrac{1}{2}\|\mathbf{w}_k\|^2 \\ \text{s.t.} \quad & \inf_{\mathbf{X}_k\sim(\boldsymbol{\mu}_k,\Sigma_k)} \Pr\{\mathbf{w}_k{}^\top\mathbf{X}_k + b_k \geq 1\} \geq \eta_k, \\ & \inf_{\mathbf{X}_k^c\sim(\boldsymbol{\mu}_k^c,\Sigma_k^c)} \Pr\{\mathbf{w}_k{}^\top\mathbf{X}_k^c + b_k \leq -1\} \geq \eta_k^c, \end{aligned} \quad (10)$$

where $\mathbf{X}_k^c$ is a random variable that generates instances of all classes except k. This random variable has a mean and covariance $(\boldsymbol{\mu}_k^c, \Sigma_k^c)$, where Σ_k and $\Sigma_k^c \in \Re^{n\times n}$ are symmetric positive semidefinite matrices. Again, the application of the Chebyshev-Cantelli inequality leads to the following quadratic SOCP formulation, for each $k = 1, \ldots, K$:

$$\begin{aligned} \min_{\mathbf{w}_k,b_k,t_k} \quad & \tfrac{1}{2}\|\mathbf{w}_k\|^2 \\ \text{s.t.} \quad & \mathbf{w}_k^\top\boldsymbol{\mu}_k + b_k \geq 1 + \kappa_k\sqrt{\mathbf{w}_k^\top\Sigma_k\mathbf{w}_k}, \\ & -(\mathbf{w}_k^\top\boldsymbol{\mu}_k^c + b_k) \geq 1 + \kappa_k^c\sqrt{\mathbf{w}_k^\top\Sigma_k^c\mathbf{w}_k}, \end{aligned} \quad (11)$$

with $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$ (resp. $\kappa_k^c = \sqrt{\frac{\eta_k^c}{1-\eta_k^c}}$). The decision rule for a new data point $\mathbf{x} \in \Re^n$ follows: $\mathbf{x}$ belongs to the class k^* iff $k^* = \arg\max_{k=1,\ldots,K}\{\mathbf{w}_k^\top\mathbf{x} + b_k\}$.

3.3 One-vs.-one SOCP, linear version

Similar to the One-vs.-The-Rest SOCP formulation, the OvO-SVM method can be extended for maximum margin SOCP classification. As described in López and Maldonado [21, 22], samples from the k-th and the l-th classes $(k < l)$ can be classified by solving the following quadratic chance-constrained programming problem:

$$\begin{aligned} \min_{\mathbf{w}_{kl},b_{kl}} \quad & \tfrac{1}{2}\|\mathbf{w}_{kl}\|^2 \\ \text{s.t.} \quad & \inf_{\mathbf{X}_k\sim(\boldsymbol{\mu}_k,\Sigma_k)} \Pr\{\mathbf{w}_{kl}{}^\top\mathbf{X}_k + b_{kl} \geq 1\} \geq \eta_{kl}, \\ & \inf_{\mathbf{X}_l\sim(\boldsymbol{\mu}_l,\Sigma_l)} \Pr\{\mathbf{w}_{kl}{}^\top\mathbf{X}_l + b_{kl} \leq -1\} \geq \eta_{lk}, \end{aligned} \quad (12)$$

where $\eta_{kl}, \eta_{lk} \in (0, 1)$. Formulation (12) can be rewritten as the following quadratic SOCP problem:

$$\begin{aligned} \min_{\mathbf{w}_{kl},b_{kl}} \quad & \tfrac{1}{2}\|\mathbf{w}_{kl}\|^2 \\ \text{s.t.} \quad & \mathbf{w}_{kl}^\top\boldsymbol{\mu}_k + b_{kl} \geq 1 + \kappa_{kl}\sqrt{\mathbf{w}_{kl}^\top\Sigma_k\mathbf{w}_{kl}}, \\ & -\mathbf{w}_{kl}^\top\boldsymbol{\mu}_l - b_{kl} \geq 1 + \kappa_{lk}\sqrt{\mathbf{w}_{kl}^\top\Sigma_l\mathbf{w}_{kl}}, \end{aligned} \quad (13)$$

with $\kappa_{kl} = \sqrt{\frac{\eta_{kl}}{1-\eta_{kl}}}$ (resp. $\kappa_{lk} = \sqrt{\frac{\eta_{lk}}{1-\eta_{lk}}}$). This model requires $K(K-1)/2$ binary classifiers, one for each pair of classes. The decision function is given by $f_{kl}(\mathbf{x}) = \mathbf{w}_{kl}^\top\mathbf{x} + b_{kl}$, and the label for a new point $\mathbf{x} \in \Re^n$ is assigned by the Max-Wins voting strategy.

3.4 All-together multiclass SVM, linear version

Following the ideas of All-Together Multiclass SVM described in Section 2.3, a multiclass SOCP formulation in which all classifiers are constructed in a single optimization problem was presented in López and Maldonado [22]. For each class k, one classifier $(\mathbf{w}_k \in \Re^n, b_k \in \Re)$ is constructed in separate classes k and l such that the probability that the random variable $\mathbf{X}_k$ lies on the correct side of the hyperplane is greater than $\eta_{kl} \in (0, 1)$, $k, l = 1, \ldots, K,\ k \neq l$. The following chance-constrained quadratic programming formulation is proposed:

$$\begin{aligned} \min_{\mathbf{w}_k,b_k} \quad & \tfrac{1}{2}\sum_{k=1}^{K}\sum_{l=1}^{k-1}\|\mathbf{w}_k - \mathbf{w}_l\|^2 + \tfrac{1}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|^2 \\ \text{s.t.} \quad & \Pr\{(\mathbf{w}_k - \mathbf{w}_l)^\top\mathbf{X}_k - (b_k - b_l) - 1 \geq 0\} \geq \eta_{kl}, \\ & k, l = 1, \ldots, K,\ k \neq l. \end{aligned} \quad (14)$$

Equivalent to the previous formulations, the worst distribution approach based on the Chebyshev-Cantelli inequality leads to the following deterministic problem:

$$\begin{aligned} \min_{\mathbf{w}_k, b_k} \quad & \frac{1}{2}\sum_{k=1}^{K}\sum_{l=1}^{k-1}\|\mathbf{w}_l-\mathbf{w}_k\|^2+\frac{1}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|^2 \\ \text{s.t.} \quad & (\mathbf{w}_k-\mathbf{w}_l)^\top\boldsymbol{\mu}_k-(b_k-b_l)\geq 1+\kappa_{kl}\|S_k^\top(\mathbf{w}_k-\mathbf{w}_l)\|, \\ & k,l=1,\ldots,K,\ k\neq l, \end{aligned} \tag{15}$$

where $\kappa_{kl}=\sqrt{\frac{\eta_{kl}}{1-\eta_{kl}}}$, for $k,l=1,\ldots,K,\ k\neq l$.

4 Proposed kernel-based multiclass SOCP formulations

In this section, we propose three novel multiclass formulations using SOCs for kernel-based classification. We first formalize the One-vs.-The-Rest extension. Secondly, the One-vs.-One SOCP formulation is presented. Finally, an "all-together" multiclass approach for maximum margin SOCP classification is formalized.

4.1 One-vs.-the-rest SOCP, kernel-based version

The One-vs.-The-Rest SOCP formulation (Problem (11)) can be extended to nonlinear classification. Let us denote by m_k the number of elements of the class k, by m_k^c the number of elements of all classes except k, by $A^k\in\Re^{n\times m_k}$ a matrix whose columns are points of the class k, by $(A^k)^c\in\Re^{n\times m_k^c}$ a matrix whose columns are the points of all classes except k, and by $\mathbf{e}$ a vector of ones of appropriate dimension. Then, the empirical estimates of the mean and covariance are given by:

$$\boldsymbol{\mu}_k=\frac{1}{m_k}A^k\mathbf{e},\ \ \boldsymbol{\mu}_k^c=\frac{1}{m_k^c}(A^k)^c\mathbf{e},$$
$$\Sigma_k=S_kS_k^\top,\ \ \Sigma_k^c=S_k^c(S_k^c)^\top,$$

with

$$S_k=\frac{1}{\sqrt{m_k}}(A^k-\boldsymbol{\mu}_k\mathbf{e}^\top),\qquad S_k^c=\frac{1}{\sqrt{m_k^c}}((A^k)^c-\boldsymbol{\mu}_k^c\mathbf{e}^\top).$$

Since $\mathbf{w}_k\in\Re^n$, it can be written as $\mathbf{w}_k=[A^k,(A^k)^c]\mathbf{s}_k+M\mathbf{r}_k$, where M is a matrix with its columns as vectors orthogonal to the training data points, and $\mathbf{s}_k,\mathbf{r}_k$ are vectors of combining coefficients. Then,

$$\mathbf{w}_k^\top\boldsymbol{\mu}_k=\mathbf{s}_k^\top\mathbf{g}_k,\ \ \mathbf{w}_k^\top\boldsymbol{\mu}_k^c=\mathbf{s}_k^\top\mathbf{g}_k^c,$$
$$\mathbf{w}_k^\top\Sigma_k\mathbf{w}_k=\mathbf{s}_k^\top\mathbf{G}_k\mathbf{s}_k,\ \ \mathbf{w}_k^\top\Sigma_k^c\mathbf{w}_k=\mathbf{s}_k^\top\mathbf{G}_k^c\mathbf{s}_k,$$

where

$$\mathbf{g}_k=\frac{1}{m_k}\begin{bmatrix}\mathbf{K}_{11}^k\mathbf{e}\\ \mathbf{K}_{21}^k\mathbf{e}\end{bmatrix},\ \ \mathbf{g}_k^c=\frac{1}{m_k^c}\begin{bmatrix}\mathbf{K}_{12}^k\mathbf{e}\\ \mathbf{K}_{22}^k\mathbf{e}\end{bmatrix},$$

$$\mathbf{G}_k=\frac{1}{m_k}\begin{bmatrix}\mathbf{K}_{11}^k\\ \mathbf{K}_{21}^k\end{bmatrix}\left(I_{m_k}-\frac{1}{m_k}\mathbf{e}\mathbf{e}^\top\right)\left[\mathbf{K}_{11}^{k\ \top}\ \mathbf{K}_{21}^{k\ \top}\right],$$

$$\mathbf{G}_k^c=\frac{1}{m_k^c}\begin{bmatrix}\mathbf{K}_{12}^k\\ \mathbf{K}_{22}^k\end{bmatrix}\left(I_{m_k^c}-\frac{1}{m_k^c}\mathbf{e}\mathbf{e}^\top\right)\left[\mathbf{K}_{12}^{k\ \top}\ \mathbf{K}_{22}^{k\ \top}\right],$$

with $\mathbf{K}_{11}^k=(A^k)^\top A^k$, $\mathbf{K}_{12}^k=(\mathbf{K}_{21}^k)^\top=(A^k)^\top(A^k)^c$, $\mathbf{K}_{22}^k=(A^k)^c)^\top(A^k)^c$, matrices whose elements are inner products of data points. Hence, in order to design nonlinear classifiers, we replace each inner product by any function $K:\Re^n\times\Re^n\to\Re$ satisfying the Mercer condition (see [27]). Thus, the above equalities are replaced by $\mathbf{K}_{11}^k=K(A^k,A^k)$, $\mathbf{K}_{12}^k=(\mathbf{K}_{21}^k)^\top=K(A^k,(A^k)^c)$, $\mathbf{K}_{22}^k=K((A^k)^c,(A^k)^c)$. Hence, the k-th kernel-based OvR-SOCP problem solves the following formulation:

$$\begin{aligned} \min_{\boldsymbol{\alpha}_k,b_k} \quad & \frac{1}{2}\boldsymbol{\alpha}_k^\top\mathbf{K}^k\boldsymbol{\alpha}_k \\ \text{s.t.} \quad & \boldsymbol{\alpha}_k{}^\top\mathbf{g}_k+b_k\geq 1+\kappa_k\sqrt{\boldsymbol{\alpha}_k{}^\top G_k\boldsymbol{\alpha}_k}, \\ & -\boldsymbol{\alpha}_k{}^\top\mathbf{g}_k^c-b_k\geq 1+\kappa_k^c\sqrt{\boldsymbol{\alpha}_k^\top G_k^c\boldsymbol{\alpha}_k}, \end{aligned} \tag{16}$$

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

4.2 One-vs.-One SOCP, kernel-based version

Formulation (13) (OvO-SOCP) can also be extended to kernel-based classification by introducing kernel functions. Taking only training points from the k-th and the l-th classes ($k<l$) into account, kernel-based OvO-SOCP solves the following problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}_{kl},b_{kl}} \quad & \frac{1}{2}\boldsymbol{\alpha}_{kl}^\top\mathbf{K}^{kl}\boldsymbol{\alpha}_{kl} \\ \text{s.t.} \quad & \boldsymbol{\alpha}_{kl}{}^\top\mathbf{g}_k+b_{kl}\geq 1+\kappa_{kl}\sqrt{\boldsymbol{\alpha}_{kl}{}^\top G_k\boldsymbol{\alpha}_{kl}}, \\ & -\boldsymbol{\alpha}_{kl}{}^\top\mathbf{g}_l-b_{kl}\geq 1+\kappa_{lk}\sqrt{\boldsymbol{\alpha}_{kl}^\top G_l\boldsymbol{\alpha}_{kl}}, \end{aligned} \tag{17}$$

where $\mathbf{K}^{kl}=[\mathbf{K}_{kk},\mathbf{K}_{kl};\mathbf{K}_{lk},\mathbf{K}_{ll}]\in\Re^{m_k+m_l\times m_k+m_l}$,

$$\mathbf{g}_k=\frac{1}{m_k}\begin{bmatrix}\mathbf{K}_{kk}\mathbf{e}\\ \mathbf{K}_{lk}\mathbf{e}\end{bmatrix},\ \ \mathbf{g}_l=\frac{1}{m_l}\begin{bmatrix}\mathbf{K}_{kl}\mathbf{e}\\ \mathbf{K}_{ll}\mathbf{e}\end{bmatrix},$$

$$\mathbf{G}_k=\frac{1}{m_k}\begin{bmatrix}\mathbf{K}_{kk}\\ \mathbf{K}_{lk}\end{bmatrix}\left(I_{m_k}-\frac{1}{m_k}\mathbf{e}\mathbf{e}^\top\right)\left[\mathbf{K}_{kk}^\top\ \mathbf{K}_{lk}^\top\right],$$

$$\mathbf{G}_l=\frac{1}{m_l}\begin{bmatrix}\mathbf{K}_{kl}\\ \mathbf{K}_{ll}\end{bmatrix}\left(I_{m_l}-\frac{1}{m_l}\mathbf{e}\mathbf{e}^\top\right)\left[\mathbf{K}_{kl}^\top\ \mathbf{K}_{ll}^\top\right],$$

with $\mathbf{K}_{kk}=K(A^k,A^k)$, $\mathbf{K}_{kl}=(\mathbf{K}_{lk})^\top=K(A^k,A^l)$, $\mathbf{K}_{ll}=K(A^l,A^l)$.

4.3 All-together multiclass SOCP-SVM, kernel-based version

In order to obtain the kernel-based version of the multiclass SOCP formulation, we first rewrite the objective function of the linear version (Formulation (15)). Note that, for any $\mathbf{w}_k \in \Re^n$, $k = 1, \ldots, K$, the following relation holds:

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{k-1}\|\mathbf{w}_k - \mathbf{w}_l\|^2 = \sum_{k=1}^{K}\|\mathbf{w}_k - \frac{1}{K}\Sigma_{l=1}^{K}\mathbf{w}_l\|^2. \tag{18}$$

Additionally, the following relationship between the primal and dual variables can be derived from Formulation (15) (see Remark 4 in [22]):

$$\mathbf{w}_k = \frac{1}{K+1}\sum_{\substack{l=1\\ l\neq k}}^{K}(\alpha_{kl}\mathbf{z}_{kl} - \alpha_{kl}\mathbf{z}_{lk}), \quad k = 1, \ldots, K, \tag{19}$$

where $\mathbf{z}_{kl}, \alpha_{kl}$ are solutions of the following problem (see [22] for details):

$$\begin{array}{ll} \max\limits_{\alpha_{kl},\mathbf{z}_{kl}} & \sum\limits_{\substack{k,l=1\\ l\neq k}}^{K}\alpha_{kl} - \frac{1}{2(K+1)}\left\|\sum\limits_{\substack{k,l=1\\ l\neq k}}^{K}\alpha_{kl}{H^{kl}}^{\top}\mathbf{z}_{kl}\right\|^2 \\ \text{s.t.} & \mathbf{z}_{kl} = \boldsymbol{\mu}_k - \kappa_{kl}S_k\mathbf{u}^{kl},\ \|\mathbf{u}^{kl}\| \leq 1,\ k, \\ & l = 1, \ldots, K,\ k \neq l, \\ & \sum\limits_{\substack{l=1\\ l\neq k}}^{K}(\alpha_{kl} - \alpha_{lk}) = 0, \quad k = 1, \ldots, K, \\ & \alpha_{kl} \geq 0, \end{array}$$

with H^{kl} denoting an $n \times nK$ matrix with all blocks being $n \times n$ zero matrices, except for the k-th block being I_n (the identity matrix in $\Re^{n\times n}$), and the l-th block being $-I_n$, i.e.,

$$H^{kl} = [0, \ldots, 0, I_n, 0, \ldots, 0, -I_n, 0, \ldots, 0], \quad k, l = 1, \ldots, K,\ k \neq l.$$

Then, from (19) we deduce that

$$\sum_{k=1}^{K}\mathbf{w}_k = \frac{1}{K+1}\sum_{k=1}^{K}\sum_{\substack{l=1\\ l\neq k}}^{K}(\alpha_{kl}\mathbf{z}_{kl} - \alpha_{lk}\mathbf{z}_{lk}) = 0.$$

Taking into account the previous relation and (18), Problem (15) can be rewritten as:

$$\begin{array}{ll} \min\limits_{\mathbf{w}_k,b_k} & \frac{1}{2}\sum\limits_{k=1}^{K}\|\mathbf{w}_k\|^2 \\ \text{s.t.} & (\mathbf{w}_k - \mathbf{w}_l)^{\top}\boldsymbol{\mu}_k - (b_k - b_l) \geq 1 \\ & +\kappa_{kl}\|S_k^{\top}(\mathbf{w}_k - \mathbf{w}_l)\|, \\ & \quad k, l = 1, \ldots, K,\ k \neq l, \\ & \sum\limits_{k=1}^{K}\mathbf{w}_k = 0. \end{array} \tag{20}$$

The previous formulation can be extended to a kernel model. We first note that the empirical estimates of the mean and covariance of the training dataset of the class k are given by

$$\boldsymbol{\mu}_k = \frac{1}{m_k}A^k\mathbf{e}, \quad \Sigma_k = S_kS_k^{\top} \text{ with } S_k = \frac{1}{\sqrt{m_k}}(A^k - \boldsymbol{\mu}_k\mathbf{e}^{\top})$$

for $k = 1, \ldots, K$. Then,

$$\mathbf{w}_k^{\top}\boldsymbol{\mu}_k = \mathbf{s}_k^{\top}\mathbf{g}_k, \quad \mathbf{w}_k^{\top}\Sigma_k\mathbf{w}_k = \mathbf{s}_k^{\top}\mathbf{G}_k\mathbf{s}_k,$$

where

$$\mathbf{g}_k = \tfrac{1}{m_k}\begin{bmatrix}\mathbf{K}_{1k}\mathbf{e}\\ \vdots \\ \mathbf{K}_{Kk}\mathbf{e}\end{bmatrix},$$

$$\mathbf{G}_k = \tfrac{1}{m_k}\begin{bmatrix}\mathbf{K}_{1k}\\ \vdots \\ \mathbf{K}_{Kk}\end{bmatrix}\left(I_{m_k} - \tfrac{1}{m_k}\mathbf{e}\mathbf{e}^{\top}\right)[\mathbf{K}_{1k}^{\top} \cdots \mathbf{K}_{Kk}^{\top}],$$

with $\mathbf{K}_{kl} = (\mathbf{K}_{lk})^{\top} = {A^k}^{\top}(A^l) \in \Re^{m_k\times m_l}$ matrices whose elements are inner products of data points. Hence, similar to Section 4.1, the kernel formulation of Problem (15) results from replacing the inner products that appears in the matrices $\mathbf{K}_{kl}$ by any function $K : \Re^n \times \Re^n \to \Re$ satisfying the Mercer condition. Therefore, from (20) follows that the nonlinear formulation is given by

$$\begin{array}{ll} \min\limits_{\mathbf{s}_k,b_k} & \frac{1}{2}\sum\limits_{k=1}^{K}\mathbf{s}_k^{\top}\mathbf{K}\mathbf{s}_k \\ \text{s.t.} & (\mathbf{s}_k - \mathbf{s}_l)^{\top}\mathbf{g}_k - (b_k - b_l) \geq 1 \\ & +\kappa_{kl}\sqrt{(\mathbf{s}_k - \mathbf{s}_l)^{\top}\mathbf{G}_k(\mathbf{s}_k - \mathbf{s}_l)}, \\ & \quad k, l = 1, \ldots, K,\ k \neq l, \\ & \sum\limits_{k=1}^{K}\mathbf{s}_k = 0, \end{array} \tag{21}$$

where $\mathbf{K} \in \Re^{m\times m}$ is a symmetric matrix formed with the blocks $\mathbf{K}_{kl}$. Thanks to the Mercer condition, the symmetric matrix $\mathbf{K}$ is positive semidefinite.

Since the matrices $\mathbf{G}_k$ are positive semi-definite matrices, they can be factorized as $\mathbf{G}_k = \mathbf{D}_k\mathbf{D}_k^{\top}$ for each $k = 1, \ldots, K$. Thus, Problem (21) is a quadratic second-order cone programming one.

Finally, for a new sample $\mathbf{x} \in \Re^n$, we set the classification functions as

$$f_k(\mathbf{x}) = K(\mathbf{x}, \mathbb{X})\mathbf{s}_k + b_k, \quad k = 1, \ldots, K,$$

where the row vector $K(\mathbf{x}, \mathbb{X})$ is defined in (6).

5 Experimental results

We applied the proposed approaches, namely the OvR-SOCP, OvO-SOCP, and All-Together MC-SOCP methods, to seven benchmark data sets: the first six from the UCI Machine Learning Repository [3], and the last used in the classification of fish schools (see [5] for more details).

Table 1 Number of examples, number of variables and number of classes for all data sets

Dataset	#examples	#variables	#classes
IRIS	150	4	3
HAYES-ROTH	160	4	3
WINE	178	13	3
GLASS	214	13	6
LED7DIGIT	500	7	10
VOWEL	528	12	11
FISH	762	12	3

We used the standard SVM counterparts (OvR-SVM, OvO-SVM, and All-Together MC-SVM) together with recently developed multiclass SVM formulations (OvR-TWSVM, AMM, and BSGD) as alternative approaches for comparison. The relevant meta-data for each benchmark data set is presented in Table 1.

The following model selection procedure was performed: Training and test subsets were constructed using 10-fold cross-validation for all datasets. Each data point was assigned to one of the 10 subsets using stratified sampling in order to guarantee that these subsets were of almost equal size and balance ratio. The average of the 10 outcomes of the model evaluations was used as a predictor of the performance metric. More information about this procedure can be found in [14]. We used linear and Gaussian kernels.

A grid search was performed to study the influence of the kernel parameter σ, parameter C for standard SVM models, parameter c for OvR twin SVM, and η for SOCP approaches. For C, c, and σ parameters we studied the following values:

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

We explored $\eta_{kl} \in \{0.2, 0.4, 0.6, 0.8\}$ (All-Together MC-SOCP and One-vs-One SOCP-SVM), and η_k, $\eta_k^c \in \{0.2, 0.4, 0.6, 0.8\}$ (One-vs.-The-Rest classification). Balanced accuracy was used as the performance metric, which corresponds to the average recall for all classes. The recall of class k can be computed as the number of correct class k matches divided by the total number of actual class k cases. Regarding the implementation of the approaches, we used the Spider Toolbox for Matlab [39] for the standard SVM approaches, the Budgeted SVM toolbox [10] for AMM and BSGD, the successive overrelaxation (SOR) technique for OvR twin SVM [26], and the SeDuMi Matlab Toolbox for the SOCP-based classifiers [34].

Tables 2 and 3 present a summary of the results for all seven data sets and for linear and Gaussian kernels, respectively. The AMM and BSGD methods were developed as nonlinear approaches, and therefore are presented only in Table 3. The best performance among all methods in terms of balanced accuracy is highlighted in bold type.

In Tables 2 and 3 we first observe that results are better for the kernel-based versions of the seven strategies. In particular, there is a major difference in terms of performance for datasets Hayes-Roth, Vowel, and Fish. This fact demonstrates the virtues of the Gaussian kernel for multiclass classification.

A comparison between SVM and SOCP classifiers (Table 3) leads to important conclusions. First, the SOCP approaches usually achieve better results than their SVM counterparts. Although in some cases all methods reach similar performance, especially for those datasets with accuracy of almost 100 % (Iris, Wine, and Vowel), in other cases the gain is significant (Glass and Fish). Secondly, the One-vs.-The-Rest strategy performs slightly worse compared to the One-vs.-One and All-Together approaches. This fact confirms what some literature reviews suggest for multiclass classification [15], although in other cases the results are not conclusive [31]. Finally, kernel-based MC-SOCP and OvO-SOCP have the best overall performance, achieving the best balanced accuracy in four out of seven cases, although no method outperformed others in all the kernel-based experiments. Regarding the recently developed multiclass SVM formulations, the optimized approaches AMM and BSGD are always below standard and SOCP methods in terms of performance, while the OvR-TWSVM method

Table 2 Performance summary for different classification approaches. Linear kernel

	Iris	Hayes-Roth	Wine	Glass	Led7digit	Vowel	Fish
OVR-SVM$_l$	94.7	61.5	98.6	60.7	74.1	56.4	74.4
OVO-SVM$_l$	**98.0**	64.9	98.6	66.1	74.3	**90.0**	**80.0**
MC-SVM$_l$	96.0	57.9	**99.0**	57.3	75.2	72.1	69.7
OVR-TWSVM$_l$	93.3	65.4	**99.0**	58.7	74.0	58.3	75.1
OVR-SOCP$_l$	96.7	66.5	**99.0**	64.1	75.8	54.5	67.3
OVO-SOCP$_l$	97.3	63.1	**99.0**	74.8	**75.9**	81.3	77.2
MC-SOCP$_l$	97.3	**71.6**	**99.0**	**76.3**	75.7	73.6	76.1

Table 3 Performance summary for different classification approaches. Gaussian kernel

	Iris	Hayes-Roth	Wine	Glass	Led7digit	Vowel	Fish
OVR-SVM$_G$	97.3	87.2	**99.5**	71.8	74.2	**99.6**	81.6
OVO-SVM$_G$	98.0	87.7	99.0	72.2	74.7	**99.6**	82.6
MC-SVM$_G$	97.3	87.8	99.0	71.4	75.9	99.0	83.2
OVR-TWSVM$_G$	98.0	87.1	98.4	71.7	71.4	98.5	**87.7**
AMM	96.7	49.7	98.3	57.0	66.4	61.7	73.2
BSGD	96.0	53.8	96.7	73.3	60.0	98.3	62.9
OVR-SOCP$_G$	97.3	86.7	99.1	75.0	75.9	99.5	84.4
OVO-SOCP$_G$	**98.7**	87.1	**99.5**	76.3	**76.1**	99.5	85.1
MC-SOCP$_G$	**98.7**	**89.0**	**99.5**	**77.5**	75.9	99.3	85.0

achieves competitive results, with the highest balanced accuracy for Fish dataset.

The robustness analysis proposed in [12] was performed to assess the best overall performance. The relative performance of each strategy on a given dataset is computed as the ratio between its balanced accuracy and the highest one among all the methods compared. For a given method a and a dataset i, this ratio has the following form:

$$AccRatio_i(a) = \frac{bAcc(a)}{max_j \; bAcc(j)}. \quad (22)$$

The larger the (balanced) accuracy ratio for a given method a and a dataset i, the better the performance. The best method a^* will have $AccRatio_i(a^*) = 1$ for dataset i. The measure $\sum_i AccRatio_i(a)$ represents a measure of overall performance for a method a. A high value of $\sum_i AccRatio_i(a)$, close to the total number of datasets, provides a good indicator for the best overall performance and robustness. Figure 1 presents the distribution of $AccRatio_i(a)$ for all seven methods and all datasets.

It can be seen in Fig. 1 that the kernel-based SOCP approaches are indeed the best ones in terms of overall performance and robustness. The all-together MC-SOCP method achieves the best overall performance, followed by OvO-SOCP and then by OvR-SOCP. The same order can be observed for the standard SVM approaches, where OvO-SVM and all-together MC-SVM are better than OvR-SVM in terms of accuracy ratios. The OvR-TWSVM method achieves better results compared with the standard SVM approaches, demonstrating the virtues of twin SVM classification. In contrast, the optimized approaches AMM and BSGD have the lowest accuracy ratio among all methods. We can conclude that our proposals are positive contributions to the state of the art in maximum margin methods due to their powerful performance and appealing optimization schemes.

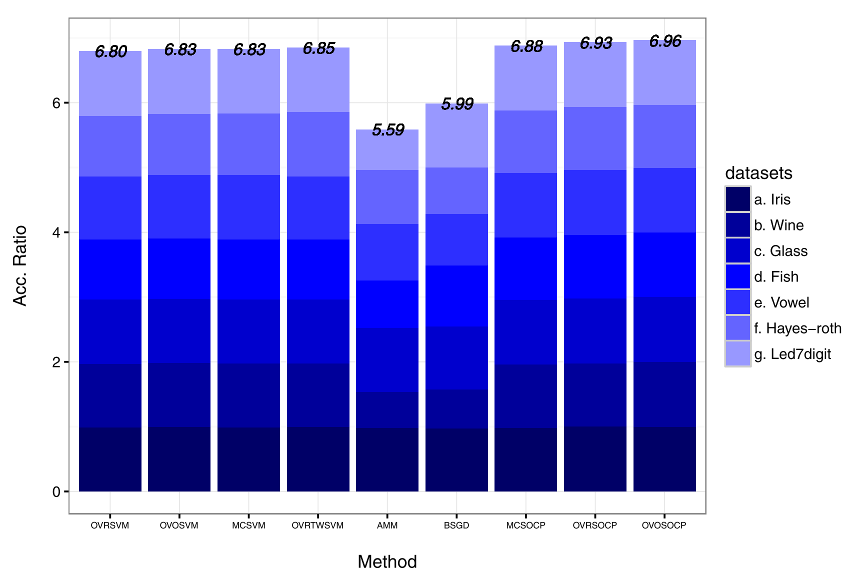

Fig. 1 Sum of accuracy ratios for all methods

6 Conclusions

The present study provides three kernel-based formulations based on second-order cone programming for multiclass maximum margin classification. These methods are extensions of the well-known One-vs.-The-Rest, One-vs.-One, and all-together MC-SVM methods. The main methodological contribution is the MC-SOCP method, which solves a single optimization problem for constructing all nonlinear classifiers, taking all available information into account. Our proposals have the following strenghts compared to these methods:

– They provide a robust framework, aiming at classifying the samples of each class correctly, up to a predefined rate, even for the worst data distribution. This robust scheme has proven to be very effective in binary and multiclass classification based on linear hyperplanes.
– The robust framework provides a balanced scheme that benefits the correct prediction of each class, since the margin maximization is performed separately for each training pattern.
– They show superior average performance compared to OvR SVM, OvO SVM, all-together MC-SVM, and other recently proposed SVM formulations. Although no method outperformed the others in terms of balanced accuracy, the robustness analysis proposed in [12] provides numerical evidence that the SOCP strategies described in this work are excellent alternatives for multiclass classification.

The main weakness of the proposals and, in particular, of the all-together multiclass classification strategy, is that the resulting problem can be very time-consuming on large scale datasets, and therefore there is a pressing need for efficient SOCP implementations. Our proposals were solved by using a generic solver like SeDuMi, in contrast to standard SVM approaches, for which ad-hoc optimization schemes like the Sequential Minimal Optimization (SMO) strategy [30] are used.

There are several opportunities for future work. First, the SOCP implementation can be improved further in order to reduce computational times, for example by speeding up algebraic operations like the computation of the kernel matrices [28] or by proposing incremental optimization schemes like the SMO approach for SVM [30] to SOCP. Secondly, the method can be extended to variations of the multiclass SVM problem, such as multiclass Twin SVM [35]. Finally, another problem that arises when facing several classes is the "class-imbalance problem", in which some of the labels are under-represented in the dataset, causing poorly balanced performance. The structure of the SOCP formulations allows us to control the different class recalls independently, providing an interesting framework for this problem [24].

Acknowledgments The first author was funded by FONDECYT project 1140831, while the second was supported by FONDECYT projects 1130905 and 1160894. The work reported in this paper has been partially funded by the Complex Engineering Systems Institute (ICM: P-05-004-F, CONICYT: FB016, www.sistemasdeingenieria.cl).

References

1. Alizadeh F, Goldfarb D (2003) Second-order cone programming. Math Progr 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. Optim Meth Soft 25(6):859–881
3. Asuncion A, Newman D (2007) UCI machine learning repository. http://archive.ics.uci.edu/ml/
4. Bhattacharyya C (2004) Second order cone programming formulations for feature selection. J Mach Learn Res 5:1417–1433
5. 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
6. Bottou L, Cortes C, Denker J, Drucker H, Guyon I, Jackel L, LeCun Y, Muller U, Sackinger E, Simard P, Vapnik V (1994) Comparison of classifier methods: a case study in handwritten digit recognition. In: Proceedings of international conference on pattern recognition, vol 2, pp 77-82
7. Bravo C, Thomas L, Weber R (2014) Improving credit scoring by differentiating defaulter behaviour. J Oper Res Soc 66:771–781
8. Bredensteiner EJ, Bennett KP (1999) Multicategory classification by support vector machines. Comp Optim Appl 12:53–79
9. 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(3):219–239
10. Djuric N, Lan L, Vucetic S, Wang Z (2013) Budgetedsvm: a toolbox for scalable svm approximations. J Mach Learn Res 14:3813–3817
11. Friedman J (1996) Another approach to polychotomous classification. Tech. rep., Department of Statistics, Stanford University, http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z
12. Geng X, Zhan DC, Zhou ZH (2005) Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Transactions on Systems, Man, and Cybernetics. Part B Cybernetics 35(6):1098–1107
13. Hao PY, Chiang JH, Lin YH (2009) A new maximal-margin spherical-structured multi-class support vector machine. Appl Intell 30:98–111
14. Hastie T, Tibshirani R, Friedman J (2008) The elements of statistical learning, Springer, chap 4.3
15. Hsu C, Lin C (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425
16. Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5):905–910
17. Kressel UG (1999) Advances in kernel methods, MIT Press, Cambridge, MA, USA, chap Pairwise classification and support vector machines:255–268
18. Lanckriet G, Ghaoui L, Bhattacharyya C, Jordan M (2003) A robust minimax approach to classification. J Mach Learn Res 3:555–582

19. Lee LH, Wan C, Rajkumar R, Isa D (2012) An enhanced support vector machine classification framework by using euclidean distance function for text document categorization. Appl Intell 37(1):80–99
20. Lin T, Liu R, Chen C, Chao Y, Chen S (2006) Pattern classification in DNA microarray data of multiple tumor types. Pattern Recogn 39(12):2426–2438
21. López J, Maldonado S (2015) Robust feature selection for multiclass support vector machines using second-order cone programming. Intell Data Anal 19(S1):S117–S133
22. López J, Maldonado S (2016) Multi-class second-order cone programming support vector machines. Inform Sci 330:328–341
23. López J, Maldonado S, Carrasco M (2016) A novel multiclass svm model using second-order cone constraints. Appl Intell 44(2):457–469
24. Maldonado S, López J (2014) Imbalanced data classification using second-order cone programming support vector machines. Pattern Recogn 47:2070–2079
25. Maldonado S, Weber R, Basak J (2011) Kernel-penalized SVM for feature selection. Inform Sci 181(1):115–128
26. Mangasarian O, Musicant D (1999) Successive overrelaxation for support vector machines. IEEE Trans Neural Netw 10(5):1032–1037
27. Mercer J (1909) Functions of positive and negative type, and their connection with the theory of integral equations. Phil Trans R Soc Lond 209:415–446
28. Michailidis P, Margaritis K (2013) Accelerating kernel density estimation on the gpu using the cuda framework. Appl Math Sci 7(30):1447–1476
29. Nath S, Bhattacharyya C (2007) Maximum margin classifiers with specified false positive and false negative error rates. Proceedings of the SIAM International Conference on Data mining
30. Platt J (1999) Advances in kernel Methods-Support vector learning, MIT press, cambridge, MA, chap Fast training of support vector machines using sequential minimal optimization
31. Rifkin R, Klautau A (2004) In defense of one-vs-all classification. J Mach Learn Res 5:101–141
32. Schölkopf B, Smola AJ (2002) Learning with Kernels. MIT Press
33. Shivaswamy PK, Bhattacharyya C, Smola AJ (2006) Second order cone programming approaches for handling missing and uncertain data. J Mach Learn Res 7:1283–1314
34. Sturm J (1999) Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optim Meth Softw 11(12):625–653
35. Tomar D, Agarwal S (2015) A comparison on multi-class classification methods based on least squares twin support vector machines. Knowl-Based Syst 81:131–147
36. Vapnik V (1998) Statistical Learning Theory. John Wiley and Sons
37. Wald V (1971) Statistical decision functions. Chelsea scientific books, Chelsea Pub. Co
38. Weston J, Watkins C (1999) Multi-class support vector machines. In: Proceedings of the Seventh European Symposium on Artificial Neural Networks
39. Weston J, Elisseeff A, BakIr G, Sinz F (2005) The spider machine learning toolbox. Software available at http://www.kyb.tuebingen.mpg.de/bs/people/spider/
40. Xie J, Hone K, Xie W, Gao X, Shi Y, Liu X (2013) Extending twin support vector machine classifier for multi-category classification problems. Intell Data Anal 17(4):649–664
41. Zhong P, Fukushima M (2007) Second-order cone programming formulations for robust multiclass classification. Neural Comp 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 Associate 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.