Information Sciences 330 (2016) 328–341

Contents lists available at ScienceDirect

Information Sciences

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

Multi-class second-order cone programming support vector machines

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

[a] Facultad de Ingeniería, Universidad Diego Portales, Ejército 441, Santiago, Chile
[b] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Monseñor Álvaro del Portillo 12455, Las Condes, Santiago, Chile

ARTICLE INFO

Article history:
Received 22 June 2014
Revised 19 August 2015
Accepted 2 October 2015
Available online 19 October 2015

Keywords:
Multi-class classification
Support vector machines
Second-order cone programming
Quadratic programming
Convex optimization

ABSTRACT

This paper presents novel second-order cone programming (SOCP) formulations that determine a linear multi-class predictor using support vector machines (SVMs). We first extend the ideas of OvO (One-versus-One) and OvA (One-versus-All) SVM formulations to SOCP-SVM, providing two interesting alternatives to the standard SVM formulations. Additionally, we propose a novel approach (MC-SOCP) that simultaneously constructs all required hyperplanes for multi-class classification, based on the multi-class SVM formulation (MC-SVM). The use of conic constraints for each pair of training patterns in a single optimization problem provides an adequate framework for a balanced and effective prediction.

1. Introduction

Second-order cone programming (SOCP) formulations have recently been proposed as an alternative optimization scheme for SVMs [1,2]. These formulations consider the worst-case setting for class-conditional densities with a given mean and covariance matrix, avoiding making assumptions about the distribution of these class-conditional densities. The SOCP formulations also provide a cost-sensitive framework for intuitively handling uneven misclassification costs in binary classification, since the methods are designed to construct a maximum margin classifier such that the false positive and false negative error rates do not exceed a predefined value [2]. SOCP problems are a special class of non-linear convex optimization problems which can be solved efficiently by interior point methods [3,4].

While SOCP-SVM has been successfully applied for binary classification, it has not yet been formalized for multi-category classification in this context, to the best of our knowledge. The only reference provided in the literature in the context of SOCP for multi-class classification is Zhong and Fukushima [5]. The method presented was used to study the problem of classification with noisy data (i.e. instances with measurement errors, see [6,7] for binary case), which is a completely different approach from the one used in this paper.

This paper is structured as follows. Section 2 introduces multi-class SVMs for classification. The proposed SOCP-SVM approaches are presented in Section 3. Section 4 provides experimental results using real-world and artificially-generated datasets. A summary of this paper can be found in Section 5, where we provide its main conclusions and address future developments.

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

http://dx.doi.org/10.1016/j.ins.2015.10.016
0020-0255/

2. Multi-class support vector machines

In this section we briefly describe the mathematical derivation of SVMs for multi-class classification (OvO, OvA and MC-SVM), which are closely related to our proposals and will be used as alternative methods in our experiments.

2.1. One-versus-All approach

This is the simplest and probably the earliest implementation for multi-class SVM [8]. This approach constructs K binary SVM classifiers, each one of which aims at separating one class from the remaining ones. Formally, for m training tuples of the form $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where $\mathbf{x}_i \in \Re^n$ represents the ith sample and $y_i \in \{1, 2, \ldots, K\}$ is the class label of $\mathbf{x}_i$, the kth SVM solves the following quadratic problem:

$$\begin{aligned} \min_{\mathbf{w}_k, b_k, \boldsymbol{\xi}^k} \quad & \frac{1}{2}\|\mathbf{w}_k\|^2 + C\sum_{i=1}^{m}\xi_i^k \\ \text{s.t.} \quad & \tilde{y}_i(\mathbf{w}_k^\top \cdot \mathbf{x}_i + b_i) \geq 1 - \xi_i^k, \\ & \xi_i^k \geq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{1}$$

where $\tilde{y}_i = 1$ means the object belongs to the target class ($y_i = k$), while $\tilde{y}_i = -1$ represents the opposite case (instance i belongs to a class different from k). Once all K hyperplanes are constructed, the decision function is given by $f_k(\mathbf{x}) = \mathbf{w}_k^\top \cdot \mathbf{x} + b_k$. Then, a new sample $\mathbf{x}$ is classified in the class with the greatest value of $f_k(\mathbf{x})$, that is, $\mathbf{x}$ is assigned to the k^*th class when $f_{k^*}(\mathbf{x}) = \max\{f_k(\mathbf{x}) : k = 1, \ldots, K\}$. Note that in the binary case (when $K = 2$), Problem (1) reduces to the classical SVM problem [9].

The OvA approach has proven to be successful and competitive compared with other multi-class approaches, according to various papers in the literature [10,11], but in some cases may lead to poor performance when the class distribution is skewed [12].

2.2. One-versus-One approach

A well-known classification approach is known as One-versus-One (OvO) SVM [13]. This method constructs $K(K-1)/2$ binary SVM classifiers, one for each pair of classes. Considering training points from the kth and the lth classes ($k < l$), OvO SVM solves the following quadratic formulation:

$$\begin{aligned} \min_{\mathbf{w}_{kl}, b_{kl}, \boldsymbol{\xi}^{kl}} \quad & \frac{1}{2}\|\mathbf{w}_{kl}\|^2 + C\sum_{r}\xi_r^{kl} \\ \text{s.t.} \quad & \mathbf{w}_{kl}^\top \cdot \mathbf{x}_r + b_{kl} \geq 1 - \xi_r^{kl}, \text{ if } y_r = k, \\ & -(\mathbf{w}_{kl}^\top \cdot \mathbf{x}_r + b_{kl}) \geq 1 - \xi_r^{kl}, \text{ if } y_r = l, \\ & \xi_r^{kl} \geq 0, \quad r = 1, \ldots, m_k + m_l. \end{aligned} \tag{2}$$

Once all $K(K-1)/2$ hyperplanes are constructed, the decision function for a new sample $\mathbf{x}$ is given by $f_{kl}(\mathbf{x}) = \mathbf{w}_{kl}^\top \cdot \mathbf{x} + b_{kl}$. Then, the Max-Wins voting strategy is used [14] in which each classifier assigns the data instances to one of the two classes, increasing the vote by one for the assigned class. The class with the majority of votes determines the classification of each data point.

2.3. MC-SVM approach

An "all-together" approach for multi-class SVMs by solving one single optimization problem was proposed in [15]. This approach constructs K binary classifiers simultaneously. The formulation of this approach is:

$$\begin{aligned} \min_{\mathbf{w}_k, b_k, \boldsymbol{\xi}^k} \quad & \frac{1}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|^2 + C\sum_{i=1}^{n}\sum_{k=1, k\neq y_i}^{K}\xi_i^k \\ \text{s.t.} \quad & (\mathbf{w}_{y_i}^\top \cdot \mathbf{x}_i + b_{y_i}) - (\mathbf{w}_k^\top \cdot \mathbf{x}_i + b_k) \geq 2 - \xi_i^k, \\ & \xi_i^k \geq 0, \quad i = 1, \ldots, m, \ k \in \{1, \ldots, K\} \setminus y_i. \end{aligned} \tag{3}$$

The decision function is similar to that of the OvA SVM formulation, that is, a new sample $\mathbf{x}$ belongs to the class k^* iff $k^* = \text{argmax}_{k=1,\ldots,K}\{\mathbf{w}_k^\top \cdot \mathbf{x} + b_k\}$. Different variations of this approach have been proposed in the literature. For instance, in [16] the SMO decomposition algorithm based on the dual formulation of SVM is extended to multi-class classification, leading to a fast and efficient kernel machine. An alternative multi-class formulation to MC-SVM can be found in [17].

3. Novel SOCP-SVM formulations for multi-class classification

In this section we formalize the proposed multi-class formulations using second-order cones. We first present the One-versus-All extension to multi-class SOCP-SVM classification. Second, the One-versus-One SOCP-SVM formulation is formalized. Finally, an "all-together" approach for multi-class SVM by using second-order cones is presented.

3.1. The One-versus-All SOCP-SVM approach

An extension of the OvA-SVM described in Section 2.1 can be derived from the SOCP-SVM formulation for binary classification. Let $\mathbf{X}_k$ be a random variable that generates samples of class k, with mean and covariance matrix given by $(\boldsymbol{\mu}_k, \Sigma_k)$; and let $\mathbf{X}_k^c$ be a random variable that generates samples of the remaining classes, having $(\boldsymbol{\mu}_k^c, \Sigma_k^c)$, where $\Sigma_k, \Sigma_k^c \in \Re^{n\times n}$ are symmetric positive semidefinite matrices. Let us denote a family of distributions which have a common mean and covariance by $\mathbf{X} \sim (\boldsymbol{\mu}, \Sigma)$. Subsequently, for each class $k = 1, \ldots, K$, we consider the following quadratic chance-constrained programming problem:

$$\begin{aligned} \min_{\mathbf{w}_k, b_k} \ & \frac{1}{2}\|\mathbf{w}_k\|^2 \\ \text{s.t.} \ & \inf_{\mathbf{X}_k \sim (\boldsymbol{\mu}_k, \Sigma_k)} \text{Prob}\{\mathbf{w}_k{}^\top \cdot \mathbf{X}_k \geq b_k + 1\} \geq \eta_k, \\ & \inf_{\mathbf{X}_k^c \sim (\boldsymbol{\mu}_k^c, \Sigma_k^c)} \text{Prob}\{\mathbf{w}_k{}^\top \cdot \mathbf{X}_k^c \leq b_k - 1\} \geq \eta_k^c, \end{aligned} \tag{4}$$

where $\eta_k, \eta_k^c \in (0, 1)$ is a predefined parameter that controls the misclassification rates for each class [2]. Formulation (4) can be rewritten as an SOCP problem thanks to the multivariate generalization of the Chebyshev–Cantelli inequality. This theorem suggests that a probabilistic approach such as the one presented in Formulation 4 can be cast into a deterministic problem since this inequality holds even for the distribution corresponding to the worst-case. The Chebyshev–Cantelli inequality follows:

Theorem 3.1. *[18, Lemma 1] Let* $\mathbf{X}$ *be a n-dimensional random variable with mean and covariance* $(\boldsymbol{\mu}, \Sigma)$*, where* Σ *is a positive semidefinite symmetric matrix. Given* $a \in \Re^n$, $b \in \Re$ *and* $\eta \in (0, 1)$*, the condition*

$$\inf_{\mathbf{X} \sim (\boldsymbol{\mu}, \Sigma)} \text{Prob}\{a^\top \mathbf{X} - b \geq 0\} \geq \eta,$$

holds if and only if

$$a^\top \boldsymbol{\mu} - b \geq \kappa \sqrt{a^\top \Sigma a},$$

where $\kappa = \sqrt{\frac{\eta}{1-\eta}}$.

Applying Theorem 3.1 to the chance-constrained programming problem presented in Formulation 4 results in the following SOCP quadratic problem (OvA-SOCP):

$$\begin{aligned} \min_{\mathbf{w}_k, b_k} \ & \frac{1}{2}\|\mathbf{w}_k\|^2 \\ \text{s.t.} \ & \mathbf{w}_k^\top \cdot \boldsymbol{\mu}_k - b_k \geq 1 + \kappa_k \|S_k^\top \mathbf{w}_k\|, \\ & b_k - \mathbf{w}_k^\top \cdot \boldsymbol{\mu}_k^c \geq 1 + \kappa_k^c \|S_k^{c\top} \mathbf{w}_k\|, \end{aligned} \tag{5}$$

with $\Sigma_k = S_k S_k^\top$, and $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$ (resp. $\kappa_k^c = \sqrt{\frac{\eta_k^c}{1-\eta_k^c}}$).

By introducing a new variable t_k and a constraint $\|\mathbf{w}_k\| \leq t_k$, Formulation (5) can be cast as the following problem:

$$\begin{aligned} \min_{\mathbf{w}_k, b_k, t_k} \ & t_k \\ \text{s.t.} \ & \|\mathbf{w}_k\| \leq t_k \\ & \mathbf{w}_k^\top \cdot \boldsymbol{\mu}_k - b_k \geq 1 + \kappa_k \|S_k^\top \mathbf{w}_k\|, \\ & b_k - \mathbf{w}_k^\top \cdot \boldsymbol{\mu}_k^c \geq 1 + \kappa_k^c \|S_k^{c\top} \mathbf{w}_k\|. \end{aligned} \tag{6}$$

This new problem is a convex formulation with a linear objective function and second-order cone (SOC) constraints [3]. An SOC constraint on the variable $\mathbf{x} \in \Re^n$ is of the form

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

where $d \in \Re$, $\mathbf{c} \in \Re^n$, $\mathbf{b} \in \Re^m$, $A \in \Re^{m \times n}$ are given. Thus, Formulation (6) can be considered as a linear second-order cone programming (SOCP) problem. This approach solves K linear SOCP problems with three SOC constraints.

The decision function is similar to the one used for the standard OvA-SVM formulation; that is, a new data point $\mathbf{x}$ belongs to the class k^* iff $k^* = \arg\max_{k=1,\ldots,K}\{\mathbf{w}_k^\top \mathbf{x} - b_k\}$.

Note that for the binary case (when $K = 2$), Problems (4) and (5) are equivalent to the formulations proposed in [2]. As reported in [19], we used the OvA version for SOCP-SVM (Formulation (5)) to classify fish schools, although the method was not formalized in the form of Formulations (4) and (5).

Since Formulation (5) solves K binary problems, we can deduce the corresponding dual formulation for each problem by following the ideas presented in [1,2], as follows:

$$\begin{aligned} \min_{\mathbf{z}_1,\mathbf{z}_2} \ & \frac{1}{2}\|\mathbf{z}_1 - \mathbf{z}_2\|^2 \\ \text{s.t.} \ & \mathbf{z}_1 \in \mathbf{B}_1(\boldsymbol{\mu}_k, S_k, \kappa_k),\ \mathbf{z}_2 \in \mathbf{B}_2(\boldsymbol{\mu}_k^c, S_k^c, \kappa_k^c), \end{aligned} \tag{7}$$

where

$$\begin{aligned} \mathbf{B}_1(\boldsymbol{\mu}_k, S_k, \kappa_k) &= \{\mathbf{z} : \mathbf{z} = \boldsymbol{\mu}_k - \kappa_k S_k \mathbf{u}_k,\ \|\mathbf{u}_k\| \le 1\}, \\ \mathbf{B}_2(\boldsymbol{\mu}_k^c, S_k^c, \kappa_k^c) &= \{\mathbf{z} : \mathbf{z} = \boldsymbol{\mu}_k^c + \kappa_k^c S_k^c \mathbf{u}_k^c,\ \|\mathbf{u}_k^c\| \le 1\}. \end{aligned}$$

3.2. The One-versus-One approach

Similar to the OvA-SOCP formulation, let $\mathbf{X}_k$ be a random variable that generates samples of class k, with mean and covariance matrix given by $(\boldsymbol{\mu}_k, \Sigma_k)$ for $k = 1, \ldots, K$, where $\Sigma_k \in \Re^{n\times n}$ are symmetric positive semidefinite matrices. Based on the idea of OvO-SVM described in Section 2.2, we can formulate an OvO version for SOCP-SVM. More precisely, for training examples from the kth and the lth classes ($k < l$), we solve the following quadratic chance-constrained programming problem:

$$\begin{aligned} \min_{\mathbf{w}_{kl}, b_{kl}} \ & \frac{1}{2}\|\mathbf{w}_{kl}\|^2 \\ \text{s.t.} \ & \inf_{\mathbf{x}_k \sim (\boldsymbol{\mu}_k, \Sigma_k)} \text{Prob}\{\mathbf{w}_{kl}{}^\top \cdot \mathbf{X}_k \ge b_{kl} + 1\} \ge \eta_{kl}, \\ & \inf_{\mathbf{x}_l \sim (\boldsymbol{\mu}_l, \Sigma_l)} \text{Prob}\{\mathbf{w}_{kl}{}^\top \cdot \mathbf{X}_l \le b_{kl} - 1\} \ge \eta_{lk}, \end{aligned} \tag{8}$$

where $\eta_{kl}, \eta_{lk} \in (0, 1)$. Again, thanks to an appropriate application of the multivariate Chebyshev–Cantelli inequality (see Theorem 3.1), Formulation (8) can be rewritten as the following quadratic SOCP problem (OvO-SOCP):

$$\begin{aligned} \min_{\mathbf{w}_{kl}, b_{kl}} \ & \frac{1}{2}\|\mathbf{w}_{kl}\|^2 \\ \text{s.t.} \ & \mathbf{w}_{kl}^\top \cdot \boldsymbol{\mu}_k - b_{kl} \ge 1 + \kappa_{kl}\|S_k^\top \mathbf{w}_{kl}\|, \\ & b_{kl} - \mathbf{w}_{kl}^\top \cdot \boldsymbol{\mu}_l \ge 1 + \kappa_{lk}\|S_l{}^\top \mathbf{w}_{kl}\|, \end{aligned} \tag{9}$$

with $\Sigma_k = S_k S_k^\top$, and $\kappa_{kl} = \sqrt{\frac{\eta_{kl}}{1-\eta_{kl}}}$ (resp. $\kappa_{lk} = \sqrt{\frac{\eta_{lk}}{1-\eta_{lk}}}$). Similarly to OvO-SVM, this method constructs $K(K-1)/2$ binary classifiers by solving a linear SOCP problem (one for each pair of classes) with three SOC constraints.

The decision function is given by $f_{kl}(\mathbf{x}) = \mathbf{w}_{kl}^\top \cdot \mathbf{x} - b_{kl}$, and the prediction of a new point $\mathbf{x}$ is done by the Max-Wins voting strategy (see Section 2.2).

Again, since Formulation (9) solves $K(K-1)/2$ binary problems, we can deduce the corresponding dual formulation for each problem by following the ideas presented in [1,2], as follows:

$$\begin{aligned} \min_{\mathbf{z}_1,\mathbf{z}_2} \ & \frac{1}{2}\|\mathbf{z}_1 - \mathbf{z}_2\|^2 \\ \text{s.t.} \ & \mathbf{z}_1 \in \mathbf{B}_1(\boldsymbol{\mu}_k, S_k, \kappa_{kl}),\ \mathbf{z}_2 \in \mathbf{B}_2(\boldsymbol{\mu}_l, S_l, \kappa_{lk}), \end{aligned} \tag{10}$$

where

$$\begin{aligned} \mathbf{B}_1(\boldsymbol{\mu}_k, S_k, \kappa_{kl}) &= \{\mathbf{z} : \mathbf{z} = \boldsymbol{\mu}_k - \kappa_{kl} S_k \mathbf{u}_k,\ \|\mathbf{u}_k\| \le 1\}, \\ \mathbf{B}_2(\boldsymbol{\mu}_l, S_l, \kappa_{lk}) &= \{\mathbf{z} : \mathbf{z} = \boldsymbol{\mu}_l + \kappa_{lk} S_l \mathbf{u}_l,\ \|\mathbf{u}_l\| \le 1\}. \end{aligned}$$

3.3. MC-SOCP, a novel k-class SOCP-SVM formulation

We present a novel multi-class SVM formulation using second-order cones, for which all classifiers are constructed via a single optimization problem. For each training point i, let $\mathcal{A}^i$ be a set of points in the n-dimensional space $\Re^n$ with cardinality m_i, for $i = 1, \ldots, K$. Let A^i be an $m_i \times n$ matrix whose rows are the points in $\mathcal{A}^i$. Denote by $\mathbf{e}^i$ the vector of ones of dimension m_i. For each i, let $\mathbf{w}_i$ be a vector in $\Re^n$ and b_i be a real number. We will assume that the sets $\mathcal{A}^i$, $i = 1, \ldots, K$, are piecewise-linearly separable [17], that is, that there exist $\mathbf{w}_i$ and b_i, $i = 1, \ldots, K$, such that

$$A^i \mathbf{w}_i - b_i \mathbf{e}^i > A^i \mathbf{w}_j - b_j \mathbf{e}^i, \quad i, j = 1, \ldots, K,\ i \ne j.$$

Let $\mathbf{X}_i$ be random vector variables that generate the samples $\mathcal{A}^i$, with a mean vector and a covariance matrix given by $(\boldsymbol{\mu}_i, \Sigma_i)$ for $i = 1, \ldots, K$, where $\Sigma_i \in \Re^{n\times n}$ are symmetric positive semidefinite matrices. In order to construct a maximum margin

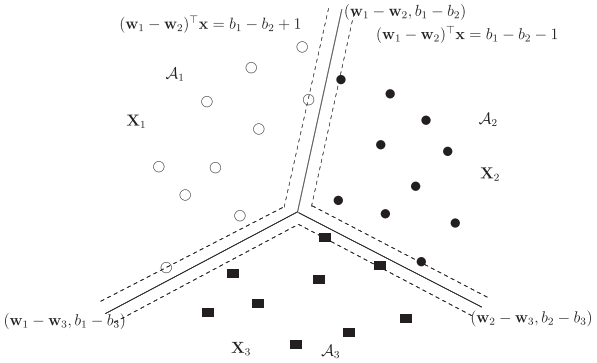

Fig. 1. Piecewise-linear separator with margins and random variables for three classes.

of separation between the classes i and j, such that the probability that the random variable $\mathbf{X}_i$ lies on the correct side of the piecewise-linear separator is greater than $\eta_{ij} \in (0,1], i, j = 1,\ldots,K,\ i \neq j$, we suggest considering the following quadratic chance-constrained programming problem:

$$\begin{aligned}\min_{\mathbf{w}_i,b_i}\quad & \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{i-1}\|\mathbf{w}_i-\mathbf{w}_j\|^2+\frac{1}{2}\sum_{i=1}^{K}\|\mathbf{w}_i\|^2\\ \text{s.t.}\quad & \text{Prob}\{(\mathbf{w}_i-\mathbf{w}_j)^\top\cdot\mathbf{X}_i-(b_i-b_j)-1\geq 0\}\geq\eta_{ij},\\ & i,j=1,\ldots,K,\ i\neq j.\end{aligned}\tag{11}$$

In this case, we want to be able to classify correctly, up to the rate η_{ij}, the instances that have common mean and covariance $\mathbf{X}_i \sim (\boldsymbol{\mu}_i, \Sigma_i)$, even for the *worst distribution* of the data [20]. The worst distribution refers to the distribution corresponding to the worst case regarding the Chebyshev inequality [18]. This inequality provides a bound that holds for a family of distributions having the same second order moments, and the worst case occurs when equality is attained for this bound [21]. For this purpose, the probabilistic constraints in (11) are replaced by their *robust* counterparts:

$$\inf_{\mathbf{X}_i\sim(\boldsymbol{\mu}_i,\Sigma_i)}\text{Prob}\{(\mathbf{w}_i-\mathbf{w}_j)^\top\cdot\mathbf{X}_i-(b_i-b_j)-1\geq 0\}\geq\eta_{ij}$$

Again, thanks to an appropriate application of the multivariate Chebyshev–Cantelli inequality (see Theorem 3.1), this worst distribution approach leads to the following deterministic problem:

$$\begin{aligned}\min_{\mathbf{w}_i,b_i}\quad & \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{i-1}\|\mathbf{w}_i-\mathbf{w}_j\|^2+\frac{1}{2}\sum_{i=1}^{K}\|\mathbf{w}_i\|^2\\ \text{s.t.}\quad & (\mathbf{w}_i-\mathbf{w}_j)^\top\cdot\boldsymbol{\mu}_i-(b_i-b_j)\geq 1+\kappa_{ij}\|S_i^\top(\mathbf{w}_i-\mathbf{w}_j)\|,\\ & i,j=1,\ldots,K,\ i\neq j.\end{aligned}\tag{12}$$

where $\Sigma_i = S_i S_i^\top$ (for instance, Cholesky factorization) and $\kappa_{ij} = \sqrt{\frac{\eta_{ij}}{1-\eta_{ij}}}$, for $i, j = 1,\ldots,K,\ i \neq j$. We named this formulation Multi-class SOCP-SVM (MC-SOCP). Fig. 1 presents a graphic representation of MC-SOCP for three classes:

Remark 1. Supposing that each training pattern $\mathbf{X}_i$ is normally distributed, i.e. $\mathbf{X}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$, then κ_{ij} can be redefined as $\kappa_{ij} = \Phi^{-1}(\eta_{ij})$, where Φ denotes the cumulative distribution function.

The decision rule follows: A new point $\mathbf{x} \in \Re^n$ belongs to the class i^*, iff

$$i^* = \arg\max_{i=1,\ldots,K} f_i(\mathbf{x}) = \arg\max_{i=1,\ldots,K}\{\mathbf{x}^\top\cdot\mathbf{w}_i - b_i\}.$$

In the next steps, we rewrite Formulation (12) in the form of a linear SOCP problem. Let us denote by:

$$\mathbf{w} = [(\mathbf{w}_1)^\top, (\mathbf{w}_2)^\top, \ldots, (\mathbf{w}_K)^\top]^\top \in \Re^{nK},$$

$$\mathbf{b} = [b_1, b_2, \ldots, b_K]^\top \in \Re^K,$$

$$\mathbf{Q} = (K+1)I_{nK} - \mathcal{J} \in \Re^{nK\times nK},$$

with

$$\mathcal{J} = \begin{bmatrix} I_n & I_n & \cdots & I_n \\ I_n & I_n & \cdots & I_n \\ \vdots & \vdots & \ddots & \vdots \\ I_n & I_n & \cdots & I_n \end{bmatrix} \in \Re^{nK\times nK},$$

where I_n and I_{nK} denote the identity matrix of size n and nK, respectively.

Note that the matrix $\mathbf{Q}$ is symmetric positive definite [see 22, Proposition 3.3]. Then, the objective function of problem (12) can be expressed as:

$$\frac{1}{2}\mathbf{w}^\top \mathbf{Q}\mathbf{w} = \frac{1}{2}\|\mathbf{Q}^{1/2}\mathbf{w}\|^2, \tag{13}$$

where

$$\mathbf{Q}^{1/2} = \sqrt{K+1}I_{nK} - \frac{\sqrt{K+1}-1}{K}\mathcal{J}.$$

Let H^{ij} be the $n \times nK$ matrix with all blocks being $n \times n$ zero matrices, except for the ith block being I_n, and the jth block being $-I_n$, that is,

$$H^{ij} = [0, \ldots, 0, I_n, 0, \ldots, 0, -I_n, 0, \ldots, 0], \quad i, j = 1, \ldots, K, \ i \neq j.$$

Then

$$\mathbf{w}_i - \mathbf{w}_j = H^{ij}\mathbf{w}. \tag{14}$$

Let $\mathbf{r}^{ij}$ be the K-dimensional vector with all components being zero, except for the ith component being 1 and the jth component being -1, that is,

$$\mathbf{r}^{ij} = [0, \ldots, 0, 1, 0, \ldots, 0, -1, 0, \ldots, 0]^\top.$$

Thus,

$$b_i - b_j = (\mathbf{r}^{ij})^\top \cdot \mathbf{b}. \tag{15}$$

By (13)–(15), Problem (12) can be rewritten as follows:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b}} \quad & \frac{1}{2}\|\mathbf{Q}^{1/2}\mathbf{w}\|^2 \\ \text{s.t.} \quad & \kappa_{ij}\|S_i^\top H^{ij}\mathbf{w}\| \leq (H^{ij}\mathbf{w})^\top \cdot \boldsymbol{\mu}_i - (\mathbf{r}^{ij})^\top \cdot \mathbf{b} - 1, \\ & i, j = 1, \ldots, K, \ i \neq j. \end{aligned} \tag{16}$$

By introducing a new variable t and a constraint $\|\mathbf{Q}^{1/2}\mathbf{w}\| \leq t$, Formulation (16) can be cast as the following linear second-order cone programming (SOCP) problem:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b},t} \quad & t \\ \text{s.t.} \quad & \|\mathbf{Q}^{1/2}\mathbf{w}\| \leq t \\ & \kappa_{ij}\|S_i^\top H^{ij}\mathbf{w}\| \leq (H^{ij}\mathbf{w})^\top \cdot \boldsymbol{\mu}_i - (\mathbf{r}^{ij})^\top \cdot \mathbf{b} - 1, \\ & i, j = 1, \ldots, K, \ i \neq j. \end{aligned} \tag{17}$$

It is important to note that our approaches have several differences compared with other SOCP methods in the literature. Debnath et al. [23], for instance, solve the standard SVM formulation via a SOCP optimization scheme. Our proposals extend the model proposed by Nath and Bhattacharyya [2] from binary classification to multi-class, which is a completely different formulation compared with standard SVM. Trafalis and Alwazzi [7] propose a method that deals with the issue of measurements errors, resulting in an SOCP model in which each data sample becomes an SOC constraint, contrary to our proposals. Our work is based on a robust setting that results in formulations in which each class becomes an SOC constraint.

Dual formulation of MC-SOCP and geometric interpretation

In the following steps we construct the dual formulation of MC-SOCP and provide a geometric interpretation of the solution obtained by this method. The Lagrangian function associated with problem (16) is given by:

$$L(\mathbf{w}, \mathbf{b}, \alpha_{ij}) = \frac{1}{2}\|\mathbf{Q}^{1/2}\mathbf{w}\|^2 + \sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij}(\kappa_{ij}\|S_i^\top H^{ij}\mathbf{w}\| - (H^{ij}\mathbf{w})^\top \boldsymbol{\mu}_i + \mathbf{r}^{ij\top}\mathbf{b} + 1).$$

Since the relationship $\|\mathbf{u}\|_2 = \max_{\|\mathbf{v}\|\leq 1} \mathbf{u}^\top \cdot \mathbf{v}$ holds for any $\mathbf{u} \in \Re^n$, we can modify the Lagrangian as follows:

$$L_1(\mathbf{w}, \mathbf{b}, \alpha_{ij}, \mathbf{u}^{ij}) = \frac{1}{2}\|\mathbf{Q}^{1/2}\mathbf{w}\|^2 + \sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij}(\kappa_{ij}(S_i^\top H^{ij}\mathbf{w})^\top \mathbf{u}^{ij} - (H^{ij}\mathbf{w})^\top \boldsymbol{\mu}_i + \mathbf{r}^{ij\top}\mathbf{b} + 1). \tag{18}$$

Then

$$L(\mathbf{w}, \mathbf{b}, \alpha_{ij}) = \max_{\mathbf{u}^{ij}}\{L_1(\mathbf{w}, \mathbf{b}, \alpha_{ij}, \mathbf{u}^{ij}) : \|\mathbf{u}^{ij}\| \leq 1, i, j = 1, \ldots, K, \ i \neq j\}.$$

Thus, Problem (16) can be written equivalently as:

$$\min_{\mathbf{w},\mathbf{b}} \max_{\alpha_{ij},\mathbf{u}^{ij}} \{L_1(\mathbf{w},\mathbf{b},\alpha_{ij},\mathbf{u}^{ij}) : \|\mathbf{u}^{ij}\| \leq 1,\ \alpha_{ij} \geq 0, i, j = 1,\ldots,K,\ i \neq j\}.$$

Hence, the dual problem of (16) is given by:

$$\max_{\alpha_{ij},\mathbf{u}^{ij}} \min_{\mathbf{w},\mathbf{b}} \{L_1(\mathbf{w},\mathbf{b},\alpha_{ij},\mathbf{u}^{ij}) : \|\mathbf{u}^{ij}\| \leq 1,\ \alpha_{ij} \geq 0,\ i, j = 1,\ldots,K,\ i \neq j\}.$$

The above expression now enables us to eliminate the primal variables to obtain the dual formulation of the problem. Computing the gradient of L_1 with respect to $\mathbf{w}$ and $\mathbf{b}$ yields:

$$\nabla_{\mathbf{w}} L_1 = \mathbf{Q}\mathbf{w} + \sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij}(\kappa_{ij} H^{ij\top} S_i \mathbf{u}^{ij} - H^{ij\top}\boldsymbol{\mu}_i),$$

$$\nabla_{\mathbf{b}} L_1 = \sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij}\mathbf{r}^{ij}.$$

Then, according to the Karush–Kuhn–Tucker conditions, we make the gradients of L_1 equals to zero, which gives:

$$\mathbf{Q}\mathbf{w} = \sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij} H^{ij\top}(\boldsymbol{\mu}_i - \kappa_{ij} S_i \mathbf{u}^{ij}), \tag{19}$$

$$\sum_{\substack{j=1 \\ j\neq i}}^{K} (\alpha_{ij} - \alpha_{ji}) = 0, \quad i = 1,\ldots,K. \tag{20}$$

Substituting the above expression in (18) subject to the relevant constraints yields the dual stated as follows:

$$\begin{aligned} \max_{\alpha_{ij},\mathbf{u}^{ij}} & \sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij} - \frac{1}{2}\|\mathbf{Q}^{-1/2} \sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij} H^{ij\top}(\boldsymbol{\mu}_i - \kappa_{ij} S_i \mathbf{u}^{ij})\|^2 \\ & \text{s.t. } \|\mathbf{u}^{ij}\| \leq 1,\ i, j = 1,\ldots,K,\ i \neq j, \\ & \sum_{\substack{j=1 \\ j\neq i}}^{K} (\alpha_{ij} - \alpha_{ji}) = 0, \quad i = 1,\ldots,K, \\ & \alpha_{ij} \geq 0. \end{aligned} \tag{21}$$

Since $\mathbf{Q}^{-1/2} = \frac{1}{\sqrt{K+1}} I_{nK} + \frac{\sqrt{K+1}-1}{K\sqrt{K+1}} \mathcal{J}$ [see 22, Proposition 3.4] and $\mathcal{J} H^{ij\top} = 0$, one has:

$$\mathbf{Q}^{-1/2} H^{ij\top} = \frac{1}{\sqrt{K+1}} H^{ij\top}. \tag{22}$$

Let

$$\boldsymbol{\alpha} = [\alpha_{12}, \alpha_{13}, \ldots, \alpha_{1K}, \ldots, \alpha_{K1}, \alpha_{K2}, \ldots, \alpha_{KK-1}]^\top \in \Re^{K(K-1)},$$

and

$$\tilde{E} = [E_1^\top\ E_2^\top\ \ldots\ E_K^\top]^\top \in \Re^{K(K-1)\times K},$$

with

$$E_i = \begin{bmatrix} -1 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -1 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & -1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & -1 \end{bmatrix} \in \Re^{K-1\times K}.$$

Then, the equality constraint in Formulation (21) can be rewritten as

$$\tilde{E}^\top \boldsymbol{\alpha} = 0. \tag{23}$$

Denote by

$$\Theta = [\Theta_1^\top \ \Theta_2^\top \ \dots \ \Theta_K^\top]^\top \in \Re^{K(K-1)\times nK},$$

where

$$\Theta_i = \begin{bmatrix} -\boldsymbol{\mu}_i^\top & \cdots & 0 & \boldsymbol{\mu}_i^\top & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -\boldsymbol{\mu}_i^\top & \boldsymbol{\mu}_i^\top & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \boldsymbol{\mu}_i^\top & -\boldsymbol{\mu}_i^\top & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \boldsymbol{\mu}_i^\top & 0 & \cdots & -\boldsymbol{\mu}_i^\top \end{bmatrix} \in \Re^{K-1\times nK}.$$

Then

$$\sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij} H^{ij^\top} \boldsymbol{\mu}_i = \Theta^\top \boldsymbol{\alpha}. \tag{24}$$

Let us denote by

$$\mathbf{U} = \begin{bmatrix} \mathbf{u^{12}} & 0 & \cdots & & 0 \\ 0 & \ddots & & & \\ & & \mathbf{u}^{1K} & & \\ \vdots & & \ddots & & \vdots \\ & & & \mathbf{u}^{K1} & \\ & & & & \ddots \\ 0 & & \cdots & & \mathbf{u}^{KK-1} \end{bmatrix} \in \Re^{nK(K-1)\times K(K-1)}$$

and by

$$\Phi = [\Phi_1^\top \ \Phi_2^\top \ \dots \ \Phi_K^\top]^\top \in \Re^{nK(K-1)\times nK},$$

with

$$\Phi_i = \begin{bmatrix} -\kappa_{i1}^\top S_i^\top & \cdots & 0 & \kappa_{i1} S_i^\top & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -\kappa_{ii-1} S_i^\top & \kappa_{ii-1} S_i^\top & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \kappa_{ii+1} S_i^\top & -\kappa_{ii+1} S_i^\top & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \kappa_{iK} S_i^\top & 0 & \cdots & -\kappa_{iK} S_i^\top \end{bmatrix} \in \Re^{n(K-1)\times nK}.$$

Then

$$\sum_{\substack{i,j=1 \\ j\neq i}}^{K} \alpha_{ij} \kappa_{ij} H^{ij^\top} S_i \mathbf{u}^{ij} = \Phi^\top \mathbf{U} \boldsymbol{\alpha}, \tag{25}$$

Hence, by using (22)–(25), we can express the dual problem (21) more compactly as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha},\mathbf{u}} \ & \boldsymbol{e}^\top \boldsymbol{\alpha} - \frac{1}{2(K+1)} \|\Theta^\top \boldsymbol{\alpha} - \Phi^\top \mathbf{U} \boldsymbol{\alpha}\|^2 \\ & \text{s.t. } \|\mathbf{u}^{ij}\| \le 1, \ i, j = 1, \dots, K, \ i \neq j, \\ & \tilde{E}^\top \boldsymbol{\alpha} = 0, \\ & \boldsymbol{\alpha} \ge 0, \end{aligned} \tag{26}$$

where $\boldsymbol{e}$ denotes a vector of ones of dimension $K(K-1)$ and

$$\mathbf{u} = [(\mathbf{u}^{12})^\top, \cdots, (\mathbf{u}^{1K})^\top, \dots, (\mathbf{u}^{K1})^\top, \cdots, (\mathbf{u}^{KK-1})^\top]^\top \in \Re^{nK(K-1)}.$$

Remark 2. Let us denote by $\mathbf{z}_{ij} = \boldsymbol{\mu}_i - \kappa_{ij} S_i \mathbf{u}^{ij}$, for $i, j = 1, \ldots, K$, $i \neq j$. Then, taking (24) and (25) into account, Problem (26) can be written as:

$$\begin{aligned}
&\max_{\alpha_{ij}, \mathbf{z}_{ij}} \sum_{\substack{i,j=1 \\ j \neq i}}^{K} \alpha_{ij} - \frac{1}{2(K+1)} \| \sum_{\substack{i,j=1 \\ j \neq i}}^{K} \alpha_{ij} H^{ij\top} \mathbf{z}_{ij} \|^2 \\
&\quad \text{s.t. } \mathbf{z}_{ij} \in \mathbf{B}_{ij}(\boldsymbol{\mu}_i, S_i, \kappa_{ij}),\ i, j = 1, \ldots, K,\ i \neq j, \\
&\quad \sum_{\substack{j=1 \\ j \neq i}}^{K} (\alpha_{ij} - \alpha_{ji}) = 0, \quad i = 1, \ldots, K, \\
&\quad \alpha_{ij} \geq 0,
\end{aligned} \tag{27}$$

where $\mathbf{B}(\boldsymbol{\mu}, S, \kappa)$ is given by:

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

The set $\mathbf{B}(\mu, S, \kappa)$ denotes an ellipsoid centered at $\boldsymbol{\mu}$ whose shape is determined by S and size by κ. This means that, for a fixed $\mathbf{z}_{ij}$, Problem (26) maximizes a quadratic concave function, at variable $\boldsymbol{\alpha}$ (see Proposition Appendix A.1), over the intersection of an affine linear space with the nonnegative orthant, and, for a fixed $\boldsymbol{\alpha}$, Problem (26) maximizes a quadratic concave function, at variable $\mathbf{z}_{ij}$ (see Proposition Appendix A.1), over the Cartesian product of ellipsoids.

Remark 3. Taking $K = 2$, Formulation (27) is reduced to

$$\begin{aligned}
&\max_{\alpha, \mathbf{z}_1, \mathbf{z}_2} 2\alpha - \frac{\alpha^2}{3} \|\mathbf{z}_1 - \mathbf{z}_2\|^2 \\
&\quad \text{s.t. } \mathbf{z}_i \in \mathbf{B}_i(\boldsymbol{\mu}_i, S_i, \kappa_i),\ i = 1, 2, \\
&\quad \alpha \geq 0.
\end{aligned}$$

Note that the objective function of this problem is maximized when $\alpha = \frac{3}{\|\mathbf{z}_1 - \mathbf{z}_2\|^2}$, and its maximum value is $\frac{3}{\|\mathbf{z}_1 - \mathbf{z}_2\|^2}$. Then, the above formulation can be rewritten as:

$$\begin{aligned}
&\min_{\mathbf{z}_1, \mathbf{z}_2} \frac{1}{3} \|\mathbf{z}_1 - \mathbf{z}_2\|^2 \\
&\quad \text{s.t. } \mathbf{z}_i \in \mathbf{B}_i(\boldsymbol{\mu}_i, S_i, \kappa_i),\ i = 1, 2.
\end{aligned}$$

This formulation can be interpreted as finding the minimum distance between two ellipsoids (see [2]).

Remark 4. From (19), (22), (24) and (25), we obtain:

$$\mathbf{w} = \frac{1}{K+1} (\Theta^\top - \Phi^\top \mathbf{U}) \boldsymbol{\alpha} = \frac{1}{K+1} \sum_{\substack{i,j=1 \\ j \neq i}}^{K} \alpha_{ij} H^{ij\top} (\boldsymbol{\mu}_i - \kappa_{ij} S_i \mathbf{u}^{ij}).$$

Then,

$$\mathbf{w}_i = \frac{1}{K+1} \sum_{\substack{j=1 \\ j \neq i}}^{K} \left[(\alpha_{ij} \boldsymbol{\mu}_i - \alpha_{ji} \boldsymbol{\mu}_j) - (\alpha_{ij} \kappa_{ij} S_i \mathbf{u}^{ij} - \alpha_{ji} \kappa_{ji} S_j \mathbf{u}^{ji}) \right], \quad i = 1, \ldots, K.$$

Hence, the decision functions are given by

$$\begin{aligned}
f_i(\mathbf{x}) = \frac{1}{K+1} \sum_{\substack{j=1 \\ j \neq i}}^{K} \Big[&(\alpha_{ij} \mathbf{x}^\top \cdot \boldsymbol{\mu}_i - \alpha_{ji} \mathbf{x}^\top \cdot \boldsymbol{\mu}_j) \\
&- (\alpha_{ij} \kappa_{ij} \mathbf{x}^\top \cdot S_i \mathbf{u}^{ij} - \alpha_{ji} \kappa_{ji} \mathbf{x}^\top \cdot S_j \mathbf{u}^{ji}) \Big] - b_i, \quad i = 1, \ldots, K.
\end{aligned}$$

4. Experimental results

We applied the proposed SOCP-SVM approaches (MC-SOCP, OvO-SOCP, and OvA-SOCP) to eight benchmark data sets for multi-class classification. We also studied their original versions based on standard SVM (MC-SVM, OvO-SVM, and OvA-SVM) for comparison purposes.

We provide a description of the data sets in Section 4.1, while Section 4.2 presents a summary of the performance obtained for all the proposed and alternative approaches.

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

Dataset	#examples	#variables	#classes
IRIS	150	4	3
WINE	178	13	3
GLASS	214	13	6
FISH	762	12	3
SEGMENT	2310	19	7
WAVEFORM	5000	21	3
HAYES-ROTH	160	4	3
LED7DIGIT	500	7	10

Table 2
Performance summary for different classification approaches. All datasets.

	Iris	Wine	Glass	Fish	Segm	Wave	Hayes	Led7
MC-SVM	93.3**	98.6	53.1**	69.7**	88.2**	**87.2**	57.9**	75.1
OVA-SVM	94.7	98.2	60.5**	74.4	92.7**	87.0	61.5**	72.1
OVO-SVM	**96.7**	98.6	66.1*	**80.0**	95.5	87.0	64.9	74.3
MC-SOCP	**96.7**	**99.0**	**73.4**	76.1	94.4	86.6	**71.6**	75.4
OVA-SOCP	**96.7**	**99.0**	61.4**	67.3**	90.2**	86.6	66.5	**75.7**
OVO-SOCP	**96.7**	**99.0**	72.6	77.2	**96.9**	87.1	62.5*	75.0

4.1. Datasets and experimental settings

In this section we briefly describe the data sets used for benchmark and provide the classification results using different feature selection methods. These sets have already been used for benchmarks in feature selection (see, for example [10]).

We studied five real-world datasets from the UCI Machine Learning Repository [24]: Iris, Wine, Glass, Segment, and Waveform; one dataset used in a previous research project for classification of fish schools (Fish, see [19] for more details); and two artificially-generated datasets, Hayes-Roth and LED Display Domain (LED7digit), also available from the UCI Repository. Table 1 summarizes the relevant information for each benchmark data set:

For model evaluation we chose a nested cross-validation (CV) strategy (also referred as repeated double CV, see e.g. [25]): training and test subsets are obtained using a 10-fold CV (outer loop), and the training subset is further split in training and validation subsets in order to find the right hyperparameter setting (parameters C). The average of the 100 outcomes of the model evaluations is used to select the best model configuration. The final classification is then performed with the full training subsets from the outer loop and for the best configuration of parameters. The average of the 10 outcomes of the model evaluations is used as predictor of the performance metric. The classification performance is computed by averaging the 10 test results, whose samples remains unseen during the hyperparameter selection procedure. We limited ourselves to linear classifiers.

For this work we studied balanced accuracy as the main performance metric to assess predictive performance. The balanced accuracy corresponds to the Recall for each class, averaged over the number of classes. The Recall for a given class k is computed from the number of correct class k matches divided by the total number of actual class k cases.

A grid search was performed to study the influence of the parameters C for soft-margin models and η for SOCP approaches. We studied the following values of $\eta_{kl} \in \{0.2, 0.4, 0.6, 0.8\}$ (MC-SOCP and One-versus-One SOCP-SVM), and η_k, $\eta_k^c \in \{0.2, 0.4, 0.6, 0.8\}$ (One-versus-All classification). For standard SVM approaches, we used the following set of values for parameter C: $\{2^{-7}, 2^{-6}, \ldots, 2^0, \ldots, 2^6, 2^7\}$.

For the above procedure, we used the Spider Toolbox for Matlab [26] for standard SVM approaches, and the SeDuMi Matlab Toolbox for SOCP-based classifiers [27].

4.2. Classification performance summary

Table 2 summarizes the results obtained from the model selection procedure for each classification approach and for all data sets. We select the best combination of parameter C for standard SVM approaches and parameter η for SOCP-SVM approaches using balanced accuracy. The best performance among all methods in terms of this metric is highlighted in bold type. We also indicate with one asterisk where the performance is significantly lower than the best method at a 10% significance level, and with two asterisks at a 5% significance level. A t-test is used to make pairwise comparisons between the mean of each approach and the best method for a given dataset.

In Table 2 it can be seen that no method outperformed others in all experiments, although MC-SOCP and OvO-SOCP achieve notably better results on the Glass and Segment datasets, respectively, and their performance is never significantly lower than the best method. For the other datasets the differences are not conclusive. We also observe that SOCP methods performed better in general, achieving best results in six out of eight datasets.

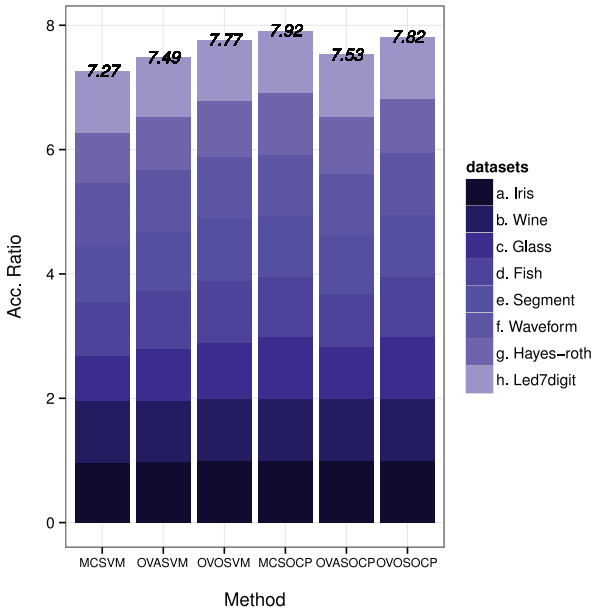

Fig. 2. Sum of accuracy ratios for all methods.

We used the robustness analysis procedure proposed in [28] to assess the overall performance of our approaches. The relative performance of a given method on a dataset is represented by the ratio between its balanced accuracy and the highest among all the compared strategies. Formally, the (balanced) accuracy ratio for method a and dataset i is:

$$AccRatio_i(a) = \frac{bAcc(a)}{max_j\ bAcc(j)}, \tag{28}$$

where $bAcc(j)$ is the balanced accuracy for method j when trained over dataset i. The larger the value of $AccRatio_i(a)$, the better the performance of a in dataset i. The best method a^* will have $AccRatio_i(a^*) = 1$ for dataset i. The measure $\Sigma_i AccRatio_i(a)$ represents a good measure of robustness and overall performance for an algorithm a, and the larger its value, the better the overall performance and robustness. Fig. 2 presents the distribution of $AccRatio_i(a)$ for all six methods and all datasets.

In Fig. 2 we observe that MC-SOCP has the best average performance, being very close to the optimal performance measure of 8. Comparing the different types of classification strategies (MC, OvA and OvO), standard SVM approaches always have lower overall performance than their robust counterparts. The most remarkable improvement is achieved for the "all-together" approaches, since MC-SVM has the lowest overall performance, while MC-SOCP has the best. We conclude that all three proposed approaches contribute to improving the performance of SVM multi-class classifiers.

4.3. Complexity and running times

The proposed approaches are based on SOCP formulations, which are known to be more time-consuming than standard SVM and therefore, in general, less suitable for machine learning. Table 3 provides a comparison for one run of each method (the average running time for one fold using 10-fold cross-validation). The experiments were performed on an HP Envy dv6 with 16 GB RAM, 750 GB SSD, an i7-2620M processor with 2.70 GHz, and using Microsoft Windows 8.1 Operating System (64-bits). We used the SeDuMI solver for Matlab 7.12 for the proposed SOCP approaches, and the spider toolbox [26] and LIBSVM [29] were used for the multi-class SVM approaches to solve the quadratic optimization problem.

Table 3
Average running times, in seconds, for all datasets.

	Iris	Wine	Glass	Fish	Segm	Wave	Hayes	Led7
MC-SVM	0".48	0".56	6".33	14627".13	6323".20	23".85	441".73	0".48
OVA-SVM	0".37	0".43	1".16	59".77	14".78	0".58	0".82	0".38
OVO-SVM	0".20	0".25	0".90	9".15	5".22	0".37	4".66	0".25
MC-SVM	0".41	0".42	0".99	6".76	1".85	0".84	0".997	0".28
OVA-SVM	0".73	0".76	1".57	3".36	1".94	1".04	1".85	0".63
OVO-SVM	0".27	0".66	3".02	5".63	1".64	0".98	7".43	0".47

It is important to notice that, for the proposed approaches, all running times are tractable and reasonable (all running times less than 10 s). Our approaches are relatively similar to the alternative approaches OVA-SVM and OVO-SVM in terms of running times. In contrast, MC-SVM has prohibitive running times under the implementation used in this work (Spider Toolbox).

Regarding complexity, the OvA approach solves K SOCP problems with three SOC constraints each; the OvO approach solves $K(K-1)/2$ SOCP problems with three SOC constraints each; and the proposed all-together approach solves a single SOCP problem with $K(K-1)/2+1$ SOC constraints. In this context, the complexity of OvO-SOCP and MC-SOCP is similar, while the OvA approach is the most efficient one (although it is the one with the worst performance among the three methods).

5. Conclusions

In this work, we present three multi-class SVM methods based on second-order cone programming formulations. Our work extends the ideas of One-versus-One, One-versus-All, and MC-SVM to second-order cones, conferring robustness to the methods, given the ability of SOCP-SVM to generalize the class patterns better by following a robust optimization approach. The SOCP-SVM method also has a balanced design since each constraint corresponds to a particular training pattern that must be correctly classified up to a rate of η, assuring adequate classification performance for all available classes. Empirically, we observed that the proposed approaches achieve better overall results on eight benchmark data sets. The gain is particularly important in the two most overlapped datasets (Glass and Hayes-Roth), where MC-SOCP achieves an improvement of 7% compared with other SVM methods.

Our main contribution is the development of the MC-SOCP approach for the simultaneous classification of all classes in one single SOCP problem. The OvO-SOCP and OvA-SOCP methods are novel formulations with interesting advantages, such as more accurate results than the traditional versions and greater simplicity than the "all-together" approach. Their derivation, however, is relatively straightforward, given the state-of-the-art. For MC-SOCP, on the other hand, the construction of one single optimization problem that includes all comparisons between training patterns, and the proof that this formulation can be written in a linear SOCP problem, presented a more challenging task.

We identified the following research opportunities for future work:

- There is a pressing need for more efficient implementations of second-order cone programming formulations. Faster SOCP implementations that exploit the structure of the SVM problem are needed for them to become real alternatives to traditional SVM for large scale datasets. Recently, a very fast multi-class SVM implementation based on parallel programming has been proposed [30], and similar extensions can be made to our proposals.
- The extension of these approaches to kernel approaches may lead to better performance, thanks to their ability to construct non-linear classifiers.
- Second-order cone programming formulations have interesting properties for class-imbalanced classification. Since the parameter η controls Type I/II errors, a differentiated value of this parameter may help to construct better classification functions that help to achieve accurate results in under-represented classes. This is particularly relevant in multi-class domains, where it is relatively common to find skewed class distributions, and classifiers tend to favor the better-represented classes and produce classifiers with poorly balanced performance [31]. Recently, we proposed a novel SOCP methodology to exploit this virtue for binary classification [32], and we are currently working on its extension to multi-class SOCP.
- One-versus-One and the proposed "all-together" strategy required the construction of several classifiers (one for each pair of classes), and therefore the running times and complexity grow quadratically with the number of classes. Although finding applications with more than 10 classes is unlikely, reducing the complexity of such approaches is an interesting challenge. Recently, the OvO-SVM method has been adapted to filter out non-competent classifiers [33]. Such extensions could make the proposed approaches faster and scalable to multi-class problems with several categories.
- Multi-class classification via SVM has strong potential in several domains. One example is credit scoring, where multi-class SVM have been used to deal with two types of defaulters: those who cannot pay because of cash flow problems, and those that lack of willingness to pay [34]. We strongly believe that practitioners can benefit from the performance of the proposed SOCP-based methods, and we consider their application in different domains as future work.

Acknowledgments

The first author was funded by FONDECYT project 11110188 and by CONICYT Anillo ACT1106, while the second was supported by FONDECYT project 11121196. The authors are grateful to the anonymous reviewers who contributed to improving the quality of the original paper. The work reported in this paper has been partially funded by the Complex Engineering Systems Institute (ICM: P-05-004-F, CONICYT: FB016).

Appendix A. Concavity of the objective function of problem (26)

Proposition A.1. *Let* $f : \Re^{K(K-1)} \times \Re^{nK(K-1)} \to \Re$ *be a function defined by* $f(\boldsymbol{\alpha}, \mathbf{u}) = \boldsymbol{e}^\top \boldsymbol{\alpha} - \frac{1}{2(K+1)} \|\Theta^\top \boldsymbol{\alpha} - \Phi^\top \mathbf{U}\boldsymbol{\alpha}\|^2$. *Then, the functions* $f(\boldsymbol{\alpha}, \cdot)$ *and* $f(\cdot, \mathbf{u})$ *are concave.*

Proof. On the one hand, the gradient and the Hessian of $f(\cdot, \mathbf{u})$, with respect to $\boldsymbol{\alpha}$, are given by:

$$\nabla_{\boldsymbol{\alpha}} f = \mathbf{e} - \frac{1}{K+1}(\Theta - \mathbf{U}^\top \Phi)(\Theta^\top - \Phi^\top \mathbf{U})\boldsymbol{\alpha}$$

and

$$\nabla^2_{\boldsymbol{\alpha}\boldsymbol{\alpha}} f = -\frac{1}{K+1}(\Theta - \mathbf{U}^\top \Phi)(\Theta - \mathbf{U}\Phi)^\top,$$

respectively. On the other hand, the gradient and the Hessian of $f(\boldsymbol{\alpha}, \cdot)$, with respect to $\mathbf{u}$, are given by

$$\nabla_{\mathbf{u}} f = \frac{1}{K+1}\Psi_{\boldsymbol{\alpha}}^\top \Phi(\Theta^\top \boldsymbol{\alpha} - \Phi^\top \Psi_{\boldsymbol{\alpha}} \mathbf{u})$$

and

$$\nabla^2_{\mathbf{u}\mathbf{u}} f = -\frac{1}{K+1}\Psi_{\boldsymbol{\alpha}}^\top \Phi\Phi^\top \Psi_{\boldsymbol{\alpha}},$$

respectively, where

$$\Psi_{\boldsymbol{\alpha}} = \begin{bmatrix} \alpha_{12} I_n & 0 & \cdots & & 0 \\ 0 & \ddots & & & \\ & & \alpha_{1K} I_n & & \\ \vdots & & \ddots & & \vdots \\ & & & \alpha_{K1} I_n & \\ & & & & \ddots \\ 0 & & \cdots & & \alpha_{KK-1} I_n \end{bmatrix} \in \Re^{nK(K-1) \times nK(K-1)}.$$

Clearly, both Hessian matrices are negative semi-definite symmetric. Therefore, functions $f(\boldsymbol{\alpha}, \cdot)$ and $f(\cdot, \mathbf{u})$ are concave. □

References

[1] S. Maldonado, J. López, Alternative second-order cone programming formulations for support vector classification, Inf. Sci. 268 (2014) 328–341.
[2] S. Nath, C. Bhattacharyya, Maximum margin classifiers with specified false positive and false negative error rates, in: Proceedings of the SIAM International Conference on Data Mining, 2007.
[3] F. Alizadeh, D. Goldfarb, Second-order cone programming, Math. Program. 95 (2003) 3–51.
[4] F. Alvarez, J. López, H. Ramírez C., Interior proximal algorithm with variable metric for second-order cone programming: applications to structural optimization and support vector machines, Optim. Methods Softw. 25 (6) (2010) 859–881.
[5] P. Zhong, M. Fukushima, Second-order cone programming formulations for robust multiclass classification, Neural Comput. 19 (2007) 258–282.
[6] D. Goldfarb, G. Iyengar, Robust convex quadratically constrained programs, Math. Program. 97 (3) (2003) 495–515.
[7] T. Trafalis, S. Alwazzi, Support vector machine classification with noisy data: a second order cone programming approach, Int. J. Gen. Syst. 39 (7) (2010) 757–781.
[8] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, V. Vapnik, Comparison of classifier methods: a case study in handwritten digit recognition, in: Proceedings of International Conference on Pattern Recognition, vol. 2, 1994, pp. 77–82.
[9] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[10] C. Hsu, C. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.
[11] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[12] G.M. Fung, O.L. Mangasarian, Multicategory proximal support vector machine classifiers, Mach. Learn. 59 (1–2) (2005) 77–97.
[13] U.-G. Kressel, Advances in Kernel Methods, MIT Press, Cambridge, MA, USA, 1999, pp. 255–268.
[14] J. Friedman, Another Approach to Polychotomous Classification, Technical Report, Department of Statistics, Stanford University, 1996.
[15] J. Weston, C. Watkins, Multi-class support vector machines, in: Proceedings of the Seventh European Symposium on Artificial Neural Networks., 1999.
[16] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, J. Mach. Learn. Res. 2 (2001) 265–292.
[17] E.J. Bredensteiner, K.P. Bennett, Multicategory classification by support vector machines, Comput. Optim. Appl. 12 (1999) 53–79.
[18] G. Lanckriet, L. Ghaoui, C. Bhattacharyya, M. Jordan, A robust minimax approach to classification, J. Mach. Learn. Res. 3 (2003) 555–582.
[19] P. Bosch, J. López, H. Ramírez, H. Robotham, Support vector machine under uncertainty: an application for hydroacoustic classification of fish-schools in Chile, Expert Syst. Appl. 40 (10) (2013) 4029–4034.
[20] V. Wald, Statistical Decision Functions, Chelsea Scientific Books, Chelsea Publishing Co., 1971.
[21] P.K. Shivaswamy, C. Bhattacharyya, A.J. Smola, Second order cone programming approaches for handling missing and uncertain data, J. Mach. Learn. Res. 7 (2006) 1283–1314.
[22] Y. Yajima, Linear programming approaches for multicategory support vector machines, Eur. J. Oper. Res. 162 (2) (2005) 514–531.
[23] R. Debnath, M. Muramatsu, H. Takahashi, An efficient support vector machine learning method with second-order cone programming for large-scale problems, Appl. Intell. 23 (3) (2005) 219–239.
[24] A. Asuncion, D. Newman, UCI Machine Learning Repository, 2007.
[25] D. Krstajic, L. Buturovic, D. Leahy, S. Thomas, Cross-validation pitfalls when selecting and assessing regression and classification models, J. Cheminformat. 6 (10) (2014) 1–15.
[26] J. Weston, A. Elisseeff, G. BakIr, F. Sinz, The Spider Machine Learning Toolbox. Software available at http://www.kyb.tuebingen.mpg.de/bs/people/spider/, 2005.
[27] J. Sturm, Using SEDUMI 1.02, a MATLAB toolbox for optimization over symmetric cones, Optim. Methods Softw. 11 (12) (1999) 625–653.
[28] X. Geng, D.-C. Zhan, Z.-H. Zhou, Supervised nonlinear dimensionality reduction for visualization and classification, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 35 (6) (2005) 1098–1107.
[29] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27.software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[30] F. Lauer, Y. Guermeur, MSVMPACK: a multiclass support vector machine package, J. Mach. Learn. Res. 12 (2011) 2293–2296.
[31] K. Yang, Z. Cai, J. Li, G. Lin, A stable gene selection in microarray data analysis, BMC Bioinformat. 7 (2006) 228.
[32] S. Maldonado, J. López, Imbalanced data classification using second-order cone programming support vector machines, Pattern Recognit. 47 (2014) 2070–2079.
[33] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Dynamic classifier selection for one-vs-one strategy: avoiding non-competent classifiers, Pattern Recognit. 46 (12) (2013) 3412–3424.
[34] C. Bravo, L. Thomas, R. Weber, Improving credit scoring by differentiating defaulter behaviour, J. Oper. Res. Soc. 66 (2014) 771–781.