Pattern Recognition 48 (2015) 1598–1607

Contents lists available at ScienceDirect

Pattern Recognition

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

A multi-class SVM approach based on the l_1-norm minimization of the distances between the reduced convex hulls

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

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

ARTICLE INFO

Article history:
Received 25 July 2014
Received in revised form
29 November 2014
Accepted 8 December 2014
Available online 17 December 2014

Keywords:
Multi-class classification
Support vector machines
Linear programming

ABSTRACT

Multi-class classification is an important pattern recognition task that can be addressed accurately and efficiently by Support Vector Machine (SVM). In this work we present a novel SVM-based multi-class classification approach based on the center of the configuration, a point which is equidistant to all classes. The center of the configuration is obtained from the dual formulation by minimizing the distances between the reduced convex hulls using the l_1-norm, while the decision functions are subsequently constructed from this point. This work also extends the ideas of Zhou et al. (2002) [37] to multi-class classification. The use of l_1-norm provides a single linear programming formulation, which reduces the complexity and confers scalability compared with other multi-class SVM methods based on quadratic programming formulations. Experiments on benchmark datasets demonstrate the virtues of our approach in terms of classification performance and running times compared with various other multi-class SVM methods.

1. Introduction

Support Vector Machine (SVM) is a well-known machine learning method, which has been used frequently in classification problems because of its strong theoretical foundation and its good performance in practice. Its extension to multi-class learning has been reported extensively in the literature: while some methods solve a series of binary problems, others attempt to construct a single optimization formulation to obtain the classification hyperplanes simultaneously [9]. However, training times can be prohibitively long for both cases, even with specially tailored quadratic programming solvers, growing exponentially with the number of classes in the problem.

In this work we propose an SVM-based linear programming formulation to solve the multi-class classification problem efficiently. While most l_1-norm SVM formulations are related to the primal form of SVM, where the Euclidean norm of the weight vector is replaced by the l_1-norm (also known as LASSO penalty), here we follow a completely different strategy. We based our work on the concept of the *center of the configuration* [1,16] to obtain a point which is equidistant to all classes, while the classification functions are constructed based on this point. This novel formulation provides a geometrically grounded interpretation of the model, while the use of the l_1-norm to find the center of the configuration results in a convex programming problem that can be reduced to an efficient linear programming model (see e.g. [5,12]).

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

2. Multi-class support vector machines

In this section we describe the multi-class Support Vector Machines approach in its three most common forms (one-versus-all SVM, one-versus-one SVM, and k-class SVM). Additionally, we present recently developed SVM approximations for multi-class classification, which are highly optimized implementations designed to achieve reduced training times [10].

2.1. One-versus-all support vector machines

One-versus-all SVM is the simplest and probably the earliest formulation for multi-class SVM [9]. This approach constructs K binary SVM classifiers, where each one separates one class from

* Corresponding author. Tel.: +56 9 61704167.
E-mail addresses: micarrasco@uandes.cl (M. Carrasco), julio.lopez@udp.cl (J. López), smaldonado@uandes.cl (S. Maldonado).

http://dx.doi.org/10.1016/j.patcog.2014.12.006
0031-3203/

the remaining training patterns. The k-th SVM classifier is trained with all examples of the k-th class, labeled as the positive class, while the remaining ones have a negative label. Formally, for m training points of the form $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where $\mathbf{x}_i \in \Re^n$ is a feature vector representing the i-th sample, and $y_i \in \{1, 2, \ldots, K\}$ is the class label of $\mathbf{x}_i$, the k-th SVM solves the following 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 $\tilde{y}_i = 1$ if $y_i = k$ and $\tilde{y}_i = -1$ otherwise. The decision function is given by $f_k(\mathbf{x}) = \mathbf{w}_k^\top \cdot \mathbf{x} + b_k$. A new sample $\mathbf{x}$ will be classified in the class which attains the greatest value of $f_k(\mathbf{x})$, that is, $\mathbf{x}$ is in the k^*-th class when $f_{k^*}(\mathbf{x}) = \max\{f_k(\mathbf{x}) : k = 1, \ldots, K\}$. In the exceptional case when this maximum is attained in more than one class, sample $\mathbf{x}$ is classified in the class associated with the lowest index k^* by convention.

Note that in the binary case (when $K=2$), Problem (1) reduces to the classical SVM problem [31].

2.2. One-versus-one support vector machines

Another important SVM-based multi-class classification method is known as one-versus-one (OvO) Support Vector Machine [17]. This method constructs $K(K-1)/2$ binary SVM classifiers, one for every pair of classes. For training data from the k-th and the l-th classes, $k \neq l$ $(k < l)$, OvO-SVM solves the following binary classification problem:

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

where m_k denotes the number of elements of the class k. The decision function is given by $f_{kl}(\mathbf{x}) = \mathbf{w}_{kl}^\top \cdot \mathbf{x} + b_{kl}$.

Classification of new examples is performed by a *max-wins* voting strategy [13], in which each data point is assigned to one of the two classes, increasing the vote for the assigned class by one. Finally, the class with the maximum number of votes determines the classification of each instance. This strategy can also be adapted to filter out non-competent classifiers [14].

2.3. k-Class support vector machines

In Weston and Watkins [36], an *all-together* approach for multi-class SVM by solving one single optimization problem was proposed. This approach constructs K binary classifiers simultaneously. The formulation of this approach (k-class SVMs) is given by

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

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 $k^* = \arg\max_{k=1,\ldots,K}\{\mathbf{w}_k^\top \cdot \mathbf{x} + b_k\}$. Different variations of this approach have been proposed in the literature. For instance, Crammer and Singer [7] extend the SMO decomposition algorithm based on the dual formulation of SVM to multi-class classification, leading to a fast and efficient kernel machine. An alternative multi-class formulation to k-class SVMs can be found in Lee et al. [18].

2.4. Optimized SVM approximations

In this section we briefly describe three highly optimized SVM approximations, which are used for benchmarking purposes in the experimental section. These state-of-the-art approaches are Pegasos, Adaptive Multi-Hyperplane Machine (AMM), and Budgeted Stochastic Gradient Descent (BSGD).

The first approach, Pegasos, is an iterative algorithm that alternates between stochastic sub-gradient descent steps and projection steps. This algorithm was proposed by Shalev-Shwartz et al. [29] for binary classification, and then extended to multi-class by Wang et al. [32]. The AMM method approximates a non-linear decision boundary via multiple linear classifiers [34]. The method is trained via Stochastic Gradient Descent (SGD). Finally, the BSGD method maintains a fixed number of support vectors in the model, and incrementally updates them during the Stochastic Gradient Descent training [33].

3. Proposed multi-class SVM approaches based on the center of the configuration

In this section, we present a novel multi-class classification approach based on the l_1-norm minimization of all distances with respect to the center of the configuration. Geometrically speaking, the idea behind this approach is to minimize the distances between all reduced convex hulls using the l_1-norm. We first revisit the concept of the center of the configuration presented in two approaches, namely AD-SVMs [1] for standard SVM, and Scatter SVM [16] for ν-SVM, and present a novel geometric approach based on the l_2-norm based on the center of the configuration and the concept of reduced convex hulls, highlighting the differences and similarities with these works. Two linear formulations that can be derived from this geometric approach are subsequently presented: one considers $\mathbf{p}$ as an additional decision variable of the optimization problem (l_1-CCSVM$_p$), and the other uses an explicit value for $\mathbf{p}$ (l_1-CCSVM$_e$). The kernel version related to the latter linear formulation is described subsequently. Some extensions and properties regarding the relationship between our proposals and other approaches are discussed at the end of the section.

3.1. Center of the configuration: notation and preliminaries

The idea of the center of the configuration was introduced by Ñanculef et al. [1] for standard SVM (AD-SVM formulation) and [16] for ν-SVM (Scatter-SVM). Let us consider $X_k = [\mathbf{x}_1^k \ldots \mathbf{x}_{m_k}^k] \in \mathbb{R}^{n \times m_k}$ a data matrix, for $k = 1, \ldots, K$. Each column $\mathbf{x}_i^k \in \mathbb{R}^n$ of X_k corresponds to a feature vector representing the i-th sample related to the class k. We denote the reduced convex hull of the class associated with X by RCo(X), that is,

$$\text{RCo}(X) := \{X\boldsymbol{\mu} : \mathbf{e}^\top \boldsymbol{\mu} = 1, \ \mathbf{0} \leq \boldsymbol{\mu} \leq C\mathbf{e}\},$$

where $C < 1$ is a real fixed parameter and $\mathbf{e}$ is the vector with all one entries (see [4,8] for details about reduced convex hulls).

We start by finding a point equidistant to all classes that minimizes the distance between their reduced convex hulls. This can be obtained by solving the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mu}_k, \mathbf{p}} \quad & \frac{1}{2}\sum_{k=1}^{K}\|X_k\boldsymbol{\mu}_k - \mathbf{p}\|^2 \\ \text{s.t.} \quad & \mathbf{e}^\top \boldsymbol{\mu}_k = 1, \quad \mathbf{0} \leq \boldsymbol{\mu}_k \leq C\mathbf{e}, \quad k = 1, \ldots, K. \end{aligned} \tag{4}$$

The constraints of this problem require that $C \geq 1/N_{min}$, where N_{min} denotes the number of points in the smallest class.

Graphically, the center of the configuration for a two-dimensional toy example with three classes can be represented as in Fig. 1.

In this formulation we call $\mathbf{p}$ the center of the configuration of the multi-class problem [1]. Jenssen et al. [16] refer to it as the *arithmetic mean* because Problem (4) is unconstrained with respect to $\mathbf{p}$, so then, $\mathbf{p}$ can be computed in terms of $\boldsymbol{\mu}_k$ as

$$\mathbf{p} = \frac{1}{K}\sum_{k=1}^{K} X_k \boldsymbol{\mu}_k. \tag{5}$$

The inclusion of this explicit value for $\mathbf{p}$ leads to the following formulation, which will allow the use of kernel functions:

$$\min_{\boldsymbol{\mu}_k} \quad \frac{1}{2}\sum_{k=1}^{K} \| X_k \boldsymbol{\mu}_k - \frac{1}{K}\sum_{l=1}^{K} X_l \boldsymbol{\mu}_l \|^2 \tag{6}$$

$$\text{s.t.} \quad \mathbf{e}^\top \boldsymbol{\mu}_k = 1, \quad \mathbf{0} \le \boldsymbol{\mu}_k \le C\mathbf{e}, \quad k = 1, \ldots, K. \tag{7}$$

Remark 1. The proposed formulation differs from the one proposed by Ñanculef et al. [1] in the optimization strategy (they minimize the distances among all the reduced convex hulls), and in the computation of the center of the configuration (according to Ñanculef et al. [1], $\mathbf{p}$ is obtained heuristically).

In order to obtain a kernel-based formulation of Problem (6)–(7), we replace the inner products $(\mathbf{x}_q^{k\top}\mathbf{x}_r^l)$, $q = 1, \ldots, m_k$, $r = 1, \ldots, m_l$, that result from expanding the quadratic term of the objective function, with any function $\mathcal{K} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ satisfying the Mercer condition [23]. Using this kernel function the quantity $(\mathbf{x}_q^{k\top}\mathbf{x}_r^l)$ is replaced by $(\mathbf{K}_{kl})_{qr}$ computed as

$$(\mathbf{K}_{kl})_{qr} = \mathcal{K}(\mathbf{x}_q^k, \mathbf{x}_r^l).$$

The *Gaussian kernel*, defined by $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u}-\mathbf{v}\|^2/2\sigma^2)$ with $\sigma \in \mathbb{R}$, and the polynomial kernel $\mathcal{K}(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^\top \mathbf{v} + 1)^d$ with $d \in \mathbb{N}$ are kernel choices (see e.g. [22,28]). Let $\mathbf{K} \in \mathbb{R}^{m \times m}$ denote a symmetric matrix formed with the blocks $\mathbf{K}_{kl}$. It should be noted that the Mercer condition ensures the positive semidefiniteness of the matrix $\mathbf{K}$. Following the ideas of Ñanculef et al. [1] and Jenssen et al. [16], the kernel-based formulation follows

$$\min_{\boldsymbol{\mu}_k} \quad \frac{1}{2}\left(\sum_{k=1}^{K} \boldsymbol{\mu}_k^\top \mathbf{K}_{kk} \boldsymbol{\mu}_k - \frac{1}{K}\boldsymbol{\mu}^\top \mathbf{K} \boldsymbol{\mu} \right)$$

$$\text{s.t.} \quad \mathbf{e}^\top \boldsymbol{\mu}_k = 1, \quad \mathbf{0} \le \boldsymbol{\mu}_k \le C\mathbf{e}, \quad k = 1, \ldots, K, \tag{Pker}$$

where $\boldsymbol{\mu} = [\boldsymbol{\mu}_1;\ \boldsymbol{\mu}_2;\ \ldots; \boldsymbol{\mu}_K] \in \mathbb{R}^m$.

The classification functions are given for each class by

$$f_k(\mathbf{x}) = \sum_{q=1}^{m_k} \mu_{kq} \mathcal{K}(\mathbf{x}, \mathbf{x}_q^k) + b_k, \quad k = 1, \ldots, K. \tag{8}$$

Remark 2. It is not difficult to see that (P_{ker}) is equivalent to that studied by Ñanculef et al. [1]. In that work, the authors introduce linear and nonlinear kernels and consider the classification function $f_k(\mathbf{x}) = \mathbf{w}_k^\top(\mathbf{x} - \mathbf{p})$ with $\mathbf{p} = (1/K)\sum_{k=1}^{K} \mathbf{w}_k$ and $\mathbf{w}_k = X_k \boldsymbol{\mu}_k$. Additionally, adding the constraint $\boldsymbol{\mu}^\top \mathbf{e} = K$ to the Formulation (6)–(7) and (P_{ker}), they coincide with that given by Jenssen et al. [15,16]. This constraint is related to the ν-SVM formulation. The authors also propose the following classification function: $f_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} / \|\mathbf{w}_k\|$. This function corresponds to the angular spread with respect to the *class representatives* [15,16], i.e. the data points that support the center of the configuration ($X_1\boldsymbol{\mu}_1$, $X_2\boldsymbol{\mu}_2$, and $X_3\boldsymbol{\mu}_3$ in Fig. 1).

3.2. Proposed linear formulations

We first propose studying a linear programming formulation based on the l_1-norm where the center of the configuration is a part of the optimization problem. We recall that for a given $\mathbf{x} \in \mathbb{R}^n$

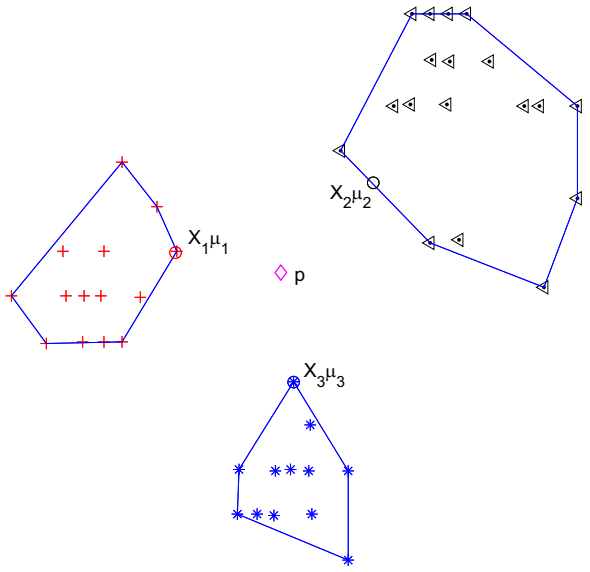

Fig. 1. Geometric illustration of Problem (4).

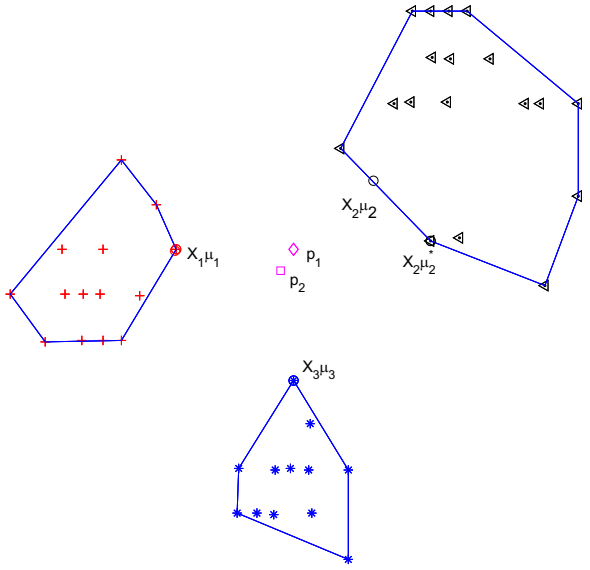

Fig. 2. Comparison between the l_2-norm and the l_1-norm formulations based on the center of the configuration.

we define $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$. The method l_1-CCSVM$_p$ can be formulated as follows:

$$\min_{\boldsymbol{\mu}_k, \mathbf{p}} \quad \sum_{k=1}^{K} \|X_k \boldsymbol{\mu}_k - \mathbf{p}\|_1 \tag{9}$$

$$\text{s.t.} \quad \mathbf{e}^\top \boldsymbol{\mu}_k = 1, \quad \mathbf{0} \le \boldsymbol{\mu}_k \le C\mathbf{e}, \quad k = 1, \ldots, K. \tag{10}$$

Fig. 2 provides a graphic representation of the previous formulation, where the solution of the l_2-norm from Formulation (4) (p_2) is compared with the l_1-norm (p_1).

In Fig. 2 we observe a slight change in the center of the configuration compared to the one obtained with the l_2-norm formulation since the support vector related to the second pattern shifted from $X_2\boldsymbol{\mu}_2$ to $X_2\boldsymbol{\mu}_2^*$. Both centers assure the correct classification of all training patterns using an adequate classification function. The l_1-CCSVM$_p$ formulation corresponds to a convex programming problem, but the objective function $\|\cdot\|_1$ is not differentiable at 0, which can be a difficulty in practice. Nevertheless, l_1-CCSVM$_p$ can be transformed to a linear programming problem by making the

following substitutions (see [5,12]):

$$X_k\boldsymbol{\mu}_k-\mathbf{p}=\mathbf{u}_k-\mathbf{v}_k,\quad \|X_k\boldsymbol{\mu}_k-\mathbf{p}\|_1=\mathbf{e}^\top(\mathbf{u}_k+\mathbf{v}_k),\quad \mathbf{u}_k,\mathbf{v}_k\geq\mathbf{0}.$$

Then, a solution of (9)–(10) can be obtained by solving the following equivalent linear formulation:

$$\begin{aligned}\min_{\boldsymbol{\mu}_k,\mathbf{p},\mathbf{u}_k,\mathbf{v}_k}\ & \sum_{k=1}^{K}\mathbf{e}^\top(\mathbf{u}_k+\mathbf{v}_k)\\ \text{s.t.}\ & \mathbf{u}_k-\mathbf{v}_k=X_k\boldsymbol{\mu}_k-\mathbf{p},\quad k=1,\ldots,K,\\ & \mathbf{e}^\top\boldsymbol{\mu}_k=1,\ \mathbf{0}\leq\boldsymbol{\mu}_k\leq C\mathbf{e},\quad k=1,\ldots,K,\\ & \mathbf{u}_k,\mathbf{v}_k\geq\mathbf{0},\quad k=1,\ldots,K.\end{aligned}\tag{PLp}$$

In order to derive a dual formulation for the problem above we computed its optimality conditions. The Lagrange function of PL^p is given by

$$\begin{aligned}L(\boldsymbol{\mu}_k,\mathbf{p},\mathbf{u}_k,\mathbf{v}_k,b_k,\mathbf{s}_k,\boldsymbol{\xi}_k,\mathbf{w}_k,\boldsymbol{\alpha}_k,\boldsymbol{\beta}_k)=&\sum_{k=1}^{K}\mathbf{e}^\top(\mathbf{u}_k+\mathbf{v}_k)+b_k(\mathbf{e}^\top\boldsymbol{\mu}_k-1)\\ &-\mathbf{s}_k^\top\boldsymbol{\mu}_k+\boldsymbol{\xi}_k^\top(\boldsymbol{\mu}_k-C\mathbf{e})-\boldsymbol{\alpha}_k^\top\mathbf{u}_k\\ &-\boldsymbol{\beta}_k^\top\mathbf{v}_k-\mathbf{w}_k^\top(\mathbf{u}_k-\mathbf{v}_k-X_k\boldsymbol{\mu}_k+\mathbf{p}).\end{aligned}\tag{11}$$

Therefore, the optimality conditions of (PL^p) are given by ($k=1,\ldots,K$)

$$\nabla_{\boldsymbol{\mu}_k}L=b_k\mathbf{e}-\mathbf{s}_k+\boldsymbol{\xi}_k+X_k^\top\mathbf{w}_k=\mathbf{0},\tag{12}$$

$$\nabla_{\mathbf{p}}L=\sum_k\mathbf{w}_k=\mathbf{0},\tag{13}$$

$$\nabla_{\mathbf{u}_k}L=\mathbf{e}-\mathbf{w}_k-\boldsymbol{\alpha}_k=\mathbf{0},\tag{14}$$

$$\nabla_{\mathbf{v}_k}L=\mathbf{e}+\mathbf{w}_k-\boldsymbol{\beta}_k=\mathbf{0},\tag{15}$$

with

$$\mathbf{s}_k^\top\boldsymbol{\mu}_k=0,\quad \boldsymbol{\xi}_k^\top(\boldsymbol{\mu}_k-C\mathbf{e})=0,\quad \boldsymbol{\alpha}_k^\top\mathbf{u}_k=0,\quad \boldsymbol{\beta}_k^\top\mathbf{v}_k=0,$$

where $\mathbf{s}_k,\boldsymbol{\xi}_k,\boldsymbol{\alpha}_k,\boldsymbol{\beta}_k\geq 0$.

Since $\boldsymbol{\alpha}_k,\boldsymbol{\beta}_k\geq\mathbf{0}$, from (14) and (15) it follows that

$$-\mathbf{e}\leq\mathbf{w}_k\leq\mathbf{e},\quad k=1,\ldots,K.$$

On the other hand, from (12) and the fact that $\mathbf{s}_k\geq\mathbf{0}$ we get

$$X_k^\top\mathbf{w}_k+b_k\mathbf{e}+\boldsymbol{\xi}_k\geq\mathbf{0},\quad k=1,\ldots,K.$$

By using (12)–(14), we obtain the dual formulation of (PL^p) as follows:

$$\begin{aligned}\min_{\mathbf{w}_k,b_k,\boldsymbol{\xi}_k}\ & \sum_{k=1}^{K}(b_k+C\boldsymbol{\xi}_k^\top\mathbf{e})\\ \text{s.t.}\ & X_k^\top\mathbf{w}_k+b_k\mathbf{e}+\boldsymbol{\xi}_k\geq\mathbf{0},\quad k=1,\ldots,K,\\ & -\mathbf{e}\leq\mathbf{w}_k\leq\mathbf{e},\quad k=1,\ldots,K,\\ & \boldsymbol{\xi}_k\geq\mathbf{0},\quad k=1,\ldots,K,\\ & \sum_{k=1}^{K}\mathbf{w}_k=\mathbf{0}.\end{aligned}\tag{DLp}$$

Since formulations (PL^p) and (DL^p) are feasible, we conclude by the strong duality theorem for linear programming (see [19, Chapter 4]) that both problems have optimal solutions and also satisfy $v(PL^p)+v(DL^p)=0$, where $v(PL^p)$ and $v(DL^p)$ are the optimal values of PL^p and DL^p, respectively. This conclusion is important since we can then use either (PL^p) or (DL^p) and achieve the same solution.

As mentioned earlier in this section, an alternative formulation is proposed based on the following expression for the center of the configuration:

$$\mathbf{p}=\frac{1}{K}\sum_{k=1}^{K}X_k\boldsymbol{\mu}_k.$$

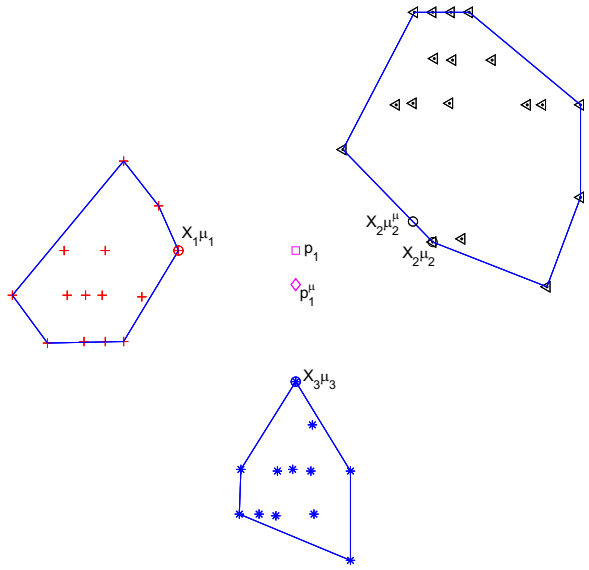

Fig. 3. Comparison between formulations l_1-CCSVM$_e$ and l_1-CCSVM$_p$.

Including this equation in problem (9)–(10), we obtain the following model (l_1-CCSVM$_e$):

$$\begin{aligned}\min_{\boldsymbol{\mu}_k}\ & \sum_{k=1}^{K}\|X_k\boldsymbol{\mu}_k-\mathbf{p}\|_1\\ \text{s.t.}\ & \mathbf{e}^\top\boldsymbol{\mu}_k=1,\quad \mathbf{0}\leq\boldsymbol{\mu}_k\leq C\mathbf{e},\quad k=1,\ldots,K,\\ & \mathbf{p}=\frac{1}{K}\sum_{k=1}^{K}X_k\boldsymbol{\mu}_k.\end{aligned}\tag{PLe}$$

We observe that the previous formulation is strongly related to Formulation (6)–(7), but the l_1-norm is used instead of the l_2-norm to minimize all distances with respect to the center of the configuration, conferring scalability to the approach. Fig. 3 illustrates the difference between methods l_1-CCSVM$_e$ and l_1-CCSVM$_p$ in terms of the position of the center of the configuration. Again, we observe only a small change in its position, demonstrating the robustness of the l_1-norm in computing distances between all data points. Using an appropriate classification function, all training instances can be shattered adequately.

Similar to the derivation of the dual formulation for l_1-CCSVM$_p$, the dual problem of the formulation l_1-CCSVM$_e$ is the following linear program:

$$\begin{aligned}\min_{\mathbf{w}_k,b_k,\boldsymbol{\xi}_k}\ & \sum_{k=1}^{K}(b_k+C\boldsymbol{\xi}_k^\top\mathbf{e})\\ & X_k^\top(\mathbf{w}_k-\overline{\mathbf{w}})+b_k\mathbf{e}+\boldsymbol{\xi}_k\geq\mathbf{0},\quad k=1,\ldots,K,\\ & -\mathbf{e}\leq\mathbf{w}_k\leq\mathbf{e},\quad k=1,\ldots,K,\\ & \boldsymbol{\xi}_k\geq\mathbf{0},\quad k=1,\ldots,K,\\ & \overline{\mathbf{w}}=\frac{1}{K}\sum_{k=1}^{K}\mathbf{w}_k.\end{aligned}\tag{DLe}$$

Again, thanks to the strong duality theorem for linear programming, both the primal and the dual formulation can be used interchangeably.

Remark 3. Note that if $(\mathbf{w}_k,b_k,\boldsymbol{\xi}_k)$ is a feasible solution of formulation (DL^e) then the point $\frac{1}{2}((\mathbf{w}_k-\overline{\mathbf{w}}),b_k,\boldsymbol{\xi}_k)$ is a feasible solution of (DL^p). The inverse is not true in general.

3.3. Kernel-based formulation

Based on the kernel-based formulation presented in Section 3.1, we propose a nonlinear kernel formulation for l_1-CCSVM$_e$. Let

us assume that

$$\mathbf{w}_k = X_k\boldsymbol{\mu}_k - \frac{1}{K}\sum_{l=1}^{K} X_l\boldsymbol{\mu}_l \quad \text{for } k=1,\ldots,K,$$

then it is easy to see that $\sum_k \mathbf{w}_k = 0$. Using this result, the inequality constraints of the dual form of l_1-CCSVM$_e$ can be rewritten as

$$X_k^\top X_k\boldsymbol{\mu}_k - \frac{1}{K}\sum_{l=1}^{K} X_k^\top X_l\boldsymbol{\mu}_l + b_k\mathbf{e} + \boldsymbol{\xi}_k \geq \mathbf{0}, \quad k=1,\ldots,K.$$

Similar to Section 3.1, we replace the dot products of the form $X_k^\top X_l$, by a kernel matrix $\mathbf{K}_{kl}$. The (q,r) entry of the kernel matrix is

$$(\mathbf{K}_{kl})_{qr} = \mathcal{K}(\mathbf{x}_q^k, \mathbf{x}_r^l), \quad q=1,\ldots,m_k, \quad r=1,\ldots,m_l,$$

where $\mathcal{K}: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is any function satisfying the Mercer condition. Taking previous equations into account, we propose the following kernel-based formulation (l_1-CCSVM$_K$):

$$\begin{aligned}
\min_{\boldsymbol{\mu}_k, b_k, \boldsymbol{\xi}_k} \quad & \sum_{k=1}^{K} (b_k + C\boldsymbol{\xi}_k^\top \mathbf{e}) \\
\text{s.t.} \quad & \mathbf{K}_{kk}\boldsymbol{\mu}_k - \frac{1}{K}\sum_{l=1}^{K} \mathbf{K}_{kl}\boldsymbol{\mu}_l + b_k\mathbf{e} + \boldsymbol{\xi}_k \geq 0, \quad k=1,\ldots,K, \\
& -\mathbf{e} \leq \boldsymbol{\mu}_k \leq \mathbf{e}, \quad k=1,\ldots,K, \\
& \boldsymbol{\xi}_k \geq \mathbf{0}, \quad k=1,\ldots,K.
\end{aligned} \tag{PLK}$$

3.4. Test rules for classification

In this work we study three classification functions found in the literature. First, we consider the standard OVA function based on the hyperplanes constructed from the weight vector, in their linear form, and in the kernel-based version:

$$f_k(\mathbf{x}) = \mathbf{w}_k^\top \cdot \mathbf{x} + b_k, \quad k=1,\ldots,K, \tag{16}$$

$$f_k(\mathbf{x}) = \sum_{q=1}^{m_k} \mu_{kq}\mathcal{K}(\mathbf{x}, \mathbf{x}_q^k) + b_k, \quad k=1,\ldots,K. \tag{17}$$

We also studied the classification function presented by Ñanculef et al. [1] based on the center of the configuration, which can be obtained either as the arithmetic mean of the weight vectors $\mathbf{p} = (1/K)\sum_{k=1}^{K}\mathbf{w}_k$ for l_1-CCSVM$_e$ or as a part of the optimization problem for l_1-CCSVM$_p$:

$$f_k(\mathbf{x}) = \mathbf{w}_k^\top \cdot (\mathbf{x} - \mathbf{p}), \quad k=1,\ldots,K, \tag{18}$$

$$f_k(\mathbf{x}) = \sum_{q=1}^{m_k} \mu_{kq}\left(\mathcal{K}(\mathbf{x}, \mathbf{x}_q^k) - \frac{1}{K}\sum_{l=1}^{K}\sum_{r=1}^{m_l} \mathcal{K}(\mathbf{x}_q^k, \mathbf{x}_r^l)\mu_{qr}\right), \quad k=1,\ldots,K. \tag{19}$$

Finally, we study the classification function related to the work proposed by Jenssen et al. [15], named *Test Rule* 1 in that work and which is based on the angular spread with respect to the class representatives as

$$f_k(\mathbf{x}) = \frac{\mathbf{w}_k^\top \cdot \mathbf{x}}{\|\mathbf{w}_k\|}, \quad k=1,\ldots,K, \tag{20}$$

$$f_k(\mathbf{x}) = \frac{1}{\sqrt{\boldsymbol{\mu}_k^\top \mathbf{K}_{kk}\boldsymbol{\mu}_k}} \sum_{q=1}^{m_k} \mu_{kq}\mathcal{K}(\mathbf{x}, \mathbf{x}_q^k), \quad k=1,\ldots,K. \tag{21}$$

3.5. Relation to other SVM-based approaches and extensions

In this section we study the relationship between our proposal and other SVM models for binary and multiclass classification. First, we recall the LP-SVM formulation proposed by Zhou et al. [37] for binary classification. In this formulation, the bound of the VC dimension is loosened properly, using the l_∞−norm [37, Theorem 2.2], resulting in an LP formulation that controls the margin maximization directly by including a margin variable r. This variable is maximized while the empirical risk is minimized simultaneously. The LP-SVM soft-margin formulation follows

$$\begin{aligned}
\min_{\hat{\mathbf{w}}, \hat{b}_k, \hat{\boldsymbol{\xi}}_k, r} \quad & -r + C[\hat{\boldsymbol{\xi}}_1^\top \mathbf{e} + \hat{\boldsymbol{\xi}}_2^\top \mathbf{e}] \\
\text{s.t.} \quad & X_1^\top \hat{\mathbf{w}} + \hat{b}_1\mathbf{e} \geq r\mathbf{e} - \hat{\boldsymbol{\xi}}_1, \\
& -X_2^\top \hat{\mathbf{w}} - \hat{b}_1\mathbf{e} \geq r\mathbf{e} - \hat{\boldsymbol{\xi}}_2, \\
& \hat{\boldsymbol{\xi}}_k \geq \mathbf{0}, \quad k=1,2, \ r \geq 0, \\
& -\mathbf{e} \leq \hat{\mathbf{w}} \leq \mathbf{e}.
\end{aligned}$$

Geometrically, the previous optimization problem in its dual form is equivalent to finding the closest points on the reduced convex hulls by using 1-norm [4,21]. Zhou et al. [37] also proposed the following kernel-based formulation:

$$\begin{aligned}
\min_{\boldsymbol{\mu}_k, b, r, \boldsymbol{\xi}_k} \quad & -r + C(\boldsymbol{\xi}_1^\top \mathbf{e} + \boldsymbol{\xi}_2^\top \mathbf{e}) \\
& \mathbf{K}_{11}\boldsymbol{\mu}_1 - \mathbf{K}_{12}\boldsymbol{\mu}_2 + b\mathbf{e} \geq r - \boldsymbol{\xi}_1, \\
& -\mathbf{K}_{21}\boldsymbol{\mu}_1 + \mathbf{K}_{22}\boldsymbol{\mu}_2 + b\mathbf{e} \geq r - \boldsymbol{\xi}_2 \\
& \boldsymbol{\xi}_k \geq \mathbf{0}, \quad k=1,2, \\
& -\mathbf{e} \leq \boldsymbol{\mu}_k \leq \mathbf{e}, \quad k=1,2.
\end{aligned} \tag{22}$$

The proposed work can be seen as an extension of the LP-SVM method proposed by Zhou et al. [37] to multi-class. First, it can be seen that when $K=2$, the method l_1-CCSVM$_e$ reduces to the following problem:

$$\begin{aligned}
\min_{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2} \quad & \|X_1\boldsymbol{\mu}_1 - X_2\boldsymbol{\mu}_2\|_1 \\
\text{s.t.} \quad & \mathbf{e}^\top\boldsymbol{\mu}_k = 1, \ \mathbf{0} \leq \boldsymbol{\mu}_k \leq C\mathbf{e}, \quad k=1,2.
\end{aligned}$$

The previous formulation is similar to the dual form of LP-SVM [21]. Based on the LP-SVM method, the following multi-class SVM formulation can be derived from the soft-margin LP-SVM model:

$$\begin{aligned}
\min_{\hat{\mathbf{w}}_k, \hat{b}_k, \hat{\boldsymbol{\xi}}_k, r} \quad & -r + C\sum_{k=1}^{K} \hat{\boldsymbol{\xi}}_k^\top \mathbf{e} \\
\text{s.t.} \quad & X_k^\top \hat{\mathbf{w}}_k + \hat{b}_k\mathbf{e} \geq r\mathbf{e} - \hat{\boldsymbol{\xi}}_k, \quad k=1,\ldots,K, \\
& -\mathbf{e} \leq \hat{\mathbf{w}}_k \leq \mathbf{e}, \quad k=1,\ldots,K, \\
& r \geq 0, \quad \hat{\boldsymbol{\xi}}_k \geq \mathbf{0}, \quad k=1,\ldots,K, \\
& \sum_{k=1}^{K} \hat{\mathbf{w}}_k = \mathbf{0}, \quad \sum_{k=1}^{K} \hat{b}_k = 0.
\end{aligned} \tag{23}$$

We first note that taking $K=2$, and denoting by $\hat{\mathbf{w}} = \hat{\mathbf{w}}_1 = -\hat{\mathbf{w}}_2$, we obtain the soft-margin LP-SVM model proposed by Zhou et al. [37]. Additionally, the following proposition relates Formulation (23) with l_1-CCSVM$_p$, demonstrating that, in fact, l_1-CCSVM$_p$ extends LP-SVM to multi-class classification.

Proposition 3.1. *Formulations (23) and l_1-CCSVM$_p$ are equivalent. More precisely, $(\mathbf{w}_k, b_k, \boldsymbol{\xi}_k)$ is a solution of l_1-CCSVM$_p$ if and only if*

$$r := -\frac{1}{K}\sum_k b_k, \quad \hat{\mathbf{w}}_k := \mathbf{w}_k, \quad \hat{b}_k := b_k + r \quad \text{and} \quad \hat{\boldsymbol{\xi}}_k := \boldsymbol{\xi}_k \ \text{ or } k=1,\ldots,K \tag{24}$$

solves Formulation (23).

The proof of Proposition 3.1 is presented in Appendix A. The following remark also demonstrates that l_1-CCSVM$_K$, the kernel-based method proposed in this work, is an extension to the kernel-based LP-SVM formulation proposed by Zhou et al. [37].

Remark 4. Taking $K=2$, the formulation (PL_K) reduces to

$$\begin{aligned}
\min_{\boldsymbol{\mu}_k, b_k, \boldsymbol{\xi}_k} \quad & b_1 + b_2 + C(\boldsymbol{\xi}_1^\top \mathbf{e} + \boldsymbol{\xi}_2^\top \mathbf{e}) \\
& \mathbf{K}_{11}\boldsymbol{\mu}_1 - \mathbf{K}_{12}\boldsymbol{\mu}_2 + 2b_1\mathbf{e} + 2\boldsymbol{\xi}_1 \geq \mathbf{0},
\end{aligned}$$

$$-\mathbf{K}_{21}\boldsymbol{\mu}_1+\mathbf{K}_{22}\boldsymbol{\mu}_2+2b_1\mathbf{e}+2\boldsymbol{\xi}_2\geq\mathbf{0},$$
$$\boldsymbol{\xi}_k\geq\mathbf{0},\quad k=1,2,$$
$$-\mathbf{e}\leq\boldsymbol{\mu}_k\leq\mathbf{e},\quad k=1,2. \tag{25}$$

If we choose $b_1=(b-r)/2$, $b_2=-(b+r)/2$, and $\hat{\boldsymbol{\xi}}_k=2\boldsymbol{\xi}_k$, with $r\geq 0$, the above formulation is then equivalent to the kernel-based LP-SVM formulation proposed by Zhou et al. [37].

In addition to LP-SVM, several SVM extensions have been proposed for efficient classification. The Sphere Support Vector Machine [30], for example, solves the minimal enclosing ball problem efficiently via an adaptation of the Sequential Minimization Approach (SMO) [27] algorithm. The Ellipsoidal Support Vector Machine [24] uses the center of an ellipsoid to approximate the Bayes point, instead of approximating it by a sphere center, as the standard SVM formulation does. This new approach leads to a convex quadratic problem that can be solved efficiently by a variant of the SMO method. Another strategy that leads to efficient SVM-based implementations is the concept of mixture-of-experts, which splits the input data into a number of subregions and trains an SVM classifier within each region [38]. One of these approaches is the Infinite Support Vector Machine, which is based on Dirichlet Process [11] for constructing the mixture of large margin classifiers. The main differences among these approaches compared with our proposal are the geometric principles behind the methods: while our approach is based on finding the center of the configuration via one-norm minimization, the Sphere Support Vector Machines, Infinite Support Vector Machines, and Ellipsoidal Support Vector Machines follow other geometrical approaches, as described above. Another important difference is the optimization scheme: Sphere Support Vector Machines, Infinite Support Vector Machines, and Ellipsoidal Support Vector Machines are quadratic problems, while our method is based on a convex linear formulation, being, therefore, potentially faster and more suitable for large scale machine learning than quadratic methods.

4. Experimental results

We applied the proposed multi-class approach to seven well-known benchmark datasets for multi-class classification, studying its different variations. We compare the proposal with standard SVM (OvO SVM, OvA SVM and *k*-class SVM), Scatter SVM [16], AD-SVM [1], Pegasos, AMM, and BSGD [10]. In Section 4.1 we provide a description of the datasets, while Section 4.2 provides a summary of the performance obtained for all the proposed and alternative approaches.

4.1. Datasets and experimental settings

In this section we briefly present the datasets used in this work. We studied four datasets from the UCI Machine Learning Repository [3]: Iris, Wine, Glass, and Vowel; one dataset from the Statlog Project Databases, Segment dataset, also available from UCI Repository; and two high-dimensional microarray datasets: MLL [2], and Glioma [26]. Table 1 summarizes the relevant metadata for each dataset:

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

Dataset	# examples	# variables	# classes
IRIS	150	4	3
WINE	178	13	3
GLASS	214	13	6
VOWEL	528	10	11
SEGMENT	2310	19	7
GLIOMA	50	4433	4
MLL	72	5848	3

We performed the following model selection procedure: The dataset was split into different training and test subsets using 10-fold cross-validation for the first five datasets, while leave-one-out validation was used for the microarray datasets. For this work we studied the balanced accuracy as the main performance metric to assess predictive performance. This metric corresponds to the Recall for each class, averaged over the number of different classes. We used the following set of values for parameter C (λ for the methods Pegasos, AMM, and BSGD) and σ:

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

Table 2
Predictive performance for all variations of the proposed approach for all datasets.

Method	Iris	Wine	Glass	Segment	Vowel	Glioma	MLL
l_1-CCSVM$_p$, d.r. (16)	90.7	95.6	42.7	53.0	27.5	72.4	**97.6**
l_1-CCSVM$_p$, d.r. (18)	81.3	87.9	59.7	72.8	30.3	68.3	95.5
l_1-CCSVM$_p$, d.r. (20)	88.0	95.3	44.3	43.2	25.6	25.0	53.6
l_1-CCSVM$_e$, d.r. (16)	73.3	95.9	46.8	68.5	30.2	74.1	96.2
l_1-CCSVM$_e$, d.r. (18)	82.7	95.3	59.6	69.9	33.8	65.2	95.0
l_1-CCSVM$_e$, d.r. (20)	82.0	93.0	37.8	47.9	29.1	25.0	58.1
l_1-CCSVM$_K$, d.r. (17)	**96.7**	**98.5**	60.3	93.8	97.4	78.7	96.0
l_1-CCSVM$_K$, d.r. (19)	46.7	43.3	33.3	14.3	67.9	41.1	54.1
l_1-CCSVM$_K$, d.r. (21)	**96.7**	98.1	**69.8**	**97.1**	**99.5**	**80.5**	**97.6**

4.2. Classification performance summary

We first study the performance of the different variations of our proposal, which involves:

- Method l_1-CCSVM$_p$ using decision rule given by Formula (16).
- Method l_1-CCSVM$_p$ using decision rule given by Formula (18).
- Method l_1-CCSVM$_p$ using decision rule given by Formula (20).
- Method l_1-CCSVM$_e$ using decision rule given by Formula (16).
- Method l_1-CCSVM$_e$ using decision rule given by Formula (18).
- Method l_1-CCSVM$_e$ using decision rule given by Formula (20).
- Method l_1-CCSVM$_K$ using decision rule given by Formula (17).
- Method l_1-CCSVM$_K$ using decision rule given by Formula (19).
- Method l_1-CCSVM$_K$ using decision rule given by Formula (21).

Table 2 summarizes the results obtained from the model selection procedure for each variation of our proposal and for all five datasets. The best performance among all variations in terms of balanced accuracy is highlighted in bold type.

In Table 2 we first observe that the best performance is achieved using kernel functions for all datasets. The method l_1-CCSVM$_K$ using the decision rule given by Formula (21) performs better in six out of seven datasets, while l_1-CCSVM$_K$ with the decision rule given by Formula (17) has better performance on the Wine dataset. The decision rule given by Formula (19) tends to fail at constructing adequate classifiers for the kernel-based method l_1-CCSVM$_K$. For linear methods, all classification functions tend to perform consistently, but kernel methods outperform the linear ones, especially for the more complex datasets (Glass, Segment, Vowel and Glioma) in terms of number of instances, number of classes, and class overlap.

Next, the results obtained from the model selection procedure for all alternative approaches are presented in Table 3. We studied the linear and kernel-based formulation for standard SVM (OvO SVM, OvA SVM and *k*-class SVM), Scatter SVM, AD-SVM, Pegasos (only available as a linear model), AMM (only available as a kernel-based model), and BSGD (only available as a kernel-based model).

Table 3
Predictive performance summary for all alternative approaches and for all datasets.

Method	Iris	Wine	Glass	Segment	Vowel	Glioma	MLL
Linear k-class SVM	96.0	99.0	57.3	90.9	72.1	**80.5**	97.1
Linear OVA-SVM	94.7	98.6	60.7	92.7	56.4	78.7	98.3
Linear OVO-SVM	**98.0**	98.6	66.1	95.6	90.0	78.7	97.1
Linear Scatter-SVM	82.0	94.5	39.1	60.4	40.1	25.0	63.6
Linear AD-SVM	76.0	95.4	52.1	57.3	41.3	65.4	98.3
Pegasos (linear)	97.3	98.3	51.4	81.5	51.7	30.0	69.3
Kernel k-class SVM	97.3	99.0	71.4	**98.3**	99.0	78.7	**98.8**
Kernel OVA-SVM	97.3	**99.5**	71.8	97.5	**99.6**	**80.5**	98.3
Kernel OVO-SVM	**98.0**	99.0	72.2	97.4	**99.6**	78.7	98.3
Kernel Scatter-SVM	96.7	98.1	69.9	97.3	99.5	78.7	**98.8**
Kernel AD-SVM	96.0	98.6	59.8	95.0	99.3	78.7	98.3
AMM (kernel-based)	96.7	98.3	57.0	83.9	61.7	78.0	98.3
BSGD (kernel-based)	96.0	96.7	**73.3**	95.9	98.3	78.0	98.6

Table 4
Predictive performance summary for all approaches and for all datasets.

Method	Iris	Wine	Glass	Segment	Vowel	Glioma	MLL
k-Class SVM	97.3	99.0	71.4	**98.3**	99.0*	78.7	**98.8**
OVA-SVM	97.3	**99.5**	71.8	97.5	99.6	**80.5**	98.3
OVO-SVM	**98.0**	99.0	72.2	97.4	**99.6**	78.7	98.3
AD-SVM	96.0	98.6	59.8**	95.0*	99.3	78.7	98.3
Scatter SVM	96.7	98.1	69.9	97.3	99.5	78.7	**98.8**
Pegasos	97.3	98.3	51.4**	81.5**	51.7**	30.0**	69.3**
AMM	96.7	98.3	57.0**	83.9**	61.7**	78.0	98.3
BSGD	96.0	96.7**	**73.3**	95.9	98.3	78.0	98.6
l_1-CCSVM	96.7	98.5	69.8	97.1	99.5	**80.5**	97.6

The best performance among all methods in terms of balanced accuracy is highlighted in bold type.

Again, we observed that the kernel-based versions perform better than the linear versions of each method. Table 4 summarizes the best performance for each method in all datasets. The best performance among all methods in terms of balanced accuracy 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 particular dataset.

From Table 4 we observe that the proposed methods are never outperformed by the best approach for each dataset. There is also no approach that performs consistently better than the others: OVO-SVM, OVA-SVM, and k-class SVM are the best method two out of seven times, while Scatter SVM, BSGD, and the proposed l_1-CCSVM are the best methods one time each. The worst performance is obtained with Pegasos since five out of seven times performs significantly worse than the best method once at a 5% significance level, followed by AMM with significantly lower accuracy three out of seven times. AD-SVM is significantly lower than the best method once at a 5% significance level and once at a 10% significance level, while k-class SVM and BSGD are significantly lower than the best method once at a 10% significance level.

The proposed approach is based on a linear programming formulation, which is more efficient and less time consuming than quadratic programming, used in standard SVM and AD-SVM. This efficiency can be very useful in large scale machine learning where huge datasets are to be analyzed. Table 5 provides a comparison for each method including the average running time using 10-fold cross-validation, and considering the best set of parameters obtained using the model selection procedure. The experiments were performed on an HP Envy dv6 with 16 GB RAM, 750 GB SSD, a i7-2620M processor with 2.70 GHz, and using Microsoft Windows 8.1 Operating System (64-bits). We used the LINPROG solver for Matlab 7.12 for the proposed approach; the QUADPROG solver for Matlab 7.12 for AD-SVM and Scatter SVM; the Budgeted SVM toolbox [10] for Pegasos, AMM, and BSGD; and the spider toolbox [35] and LIBSVM [6] were used for the multiclass SVM approaches to solve the quadratic optimization problem. Training times for AD-SVM and Scatter SVM are similar since they have essentially the same formulation but consider different classification functions, as mentioned in Remark 2. Training times for l_1-CCSVM$_e$ and l_1-CCSVM$_K$ are also similar since l_1-CCSVM$_e$ corresponds to l_1-CCSVM$_K$'s using a linear kernel.

Table 5
Average running times, in seconds, for all datasets.

Method	Iris	Wine	Glass	Segment	Vowel	Glioma	MLL
k-Class SVM	0".48	0".56	6".33	14627".13	529".23	0".31	0".24
OVA-SVM	0".37	0".43	1".16	59".77	0".98	0".57	0".57
OVO-SVM	0".20	0".25	0".90	9".15	5".02	0".65	0".46
AD-SVM	0".15	0".12	0".21	45".21	0".95	0".01	0".12
Scatter-SVM	0".15	0".12	0".21	45".21	0".95	0".01	0".12
Pegasos	0".05	0".04	0".04	0".05	0".05	0".06	0".10
AMM	0".05	0".07	0".08	0".09	0".11	0".23	0".42
BSGD	0".02	0".02	0".08	2".68	0".73	0".21	0".61
l_1-CCSVM$_p$	0".03	0".06	0".06	0".87	0".22	2".64	5".91
l_1-CCSVM$_{e\&K}$	0".04	0".08	0".10	3".07	0".53	0".19	0".37

In Table 5 we observe that our approach is consistently faster than the alternative QP approaches which, in the case of k-class SVM, may have prohibitive running times under the implementation used in this work. Furthermore, the running times achieved by our approach are comparable with the ones achieved by the most efficient SVM approximations in the literature in most cases.

Although the running times for one SVM training are below one second in most cases, the gain can be significant if an exhaustive grid search is used to tune the hyperparameters. It is also interesting to note that l_1-CCSVM$_e$ is several times faster than AD-SVM, since both formulations studied the concept of the center of the configuration in a similar fashion, but in our work the l_1-norm is used instead of the l_2-norm. The methods AD-SVM and Scatter behave faster than the proposed approaches only for the microarray datasets. It is also important to highlight that running times decrease significantly when using l_1-CCSVM$_K$ for high-dimensional datasets (such as microarray data) instead of l_1-CCSVM$_p$ or l_1-CCSVM$_e$ as in Formulation DL^e. The reason is that the number of variables for l_1-CCSVM$_K$ does not increase with the dimensionality, in contrast to l_1-CCSVM$_e$ and l_1-CCSVM$_p$, which are faster than l_1-CCSVM$_K$ when the number of cases is larger than the number of variables.

5. Conclusions

In this work, we presented two multi-class classification approaches based on Support Vector Machines and the concept of the center of the configuration. The l_1-norm is used to find a point that is equidistant to all classes, which is subsequently used to construct the classification functions. The two methods differ mainly in the computation of the center of the configuration: l_1-CCSVM$_p$, on one hand, obtains the center of the configuration directly from the optimization process, while l_1-CCSVM$_e$ provides an explicit value for it based on the arithmetic mean of the weights, on the other. From the latter method we derive a kernel-based formulation (l_1-CCSVM$_K$), conferring flexibility to the classification process by allowing non-linear classifiers, and achieving the best results among our proposals.

A comparison with other multi-class SVM classification approaches shows the advantages of the proposed methods:

- They provide a geometrically grounded framework for multi-class classification, which allows an adequate interpretation of the classification process based on the concept of reduced convex hulls.
- They achieved competitive results compared to other SVM-based methods, never being significantly below the best method for each dataset.
- They provide more efficient formulations based on linear programming, leading to an important reduction in terms of running times.

Some conclusions can be drawn from the experimental section of this work, which can be useful for practitioners:

- Predictive performance is significantly improved with kernel-based approaches, compared to linear methods. This result demonstrates the advantage of constructing nonlinear classification functions, especially in the context of models based on the center of the configuration.
- The usage of the test rule proposed by Jenssen et al. [15] based on the angular spread of the class representatives led to better results compared to other decision functions studied in this work, leading to conclusions about the importance of considering this test rule for prediction.
- There is a tradeoff between predictive performance and training times for SVM-based approaches: while best average performance can be achieved with the standard OVO-SVM, running times can be prohibitive for large datasets. Highly optimized SVM approximations, on the other hand, are extremely efficient with large-scale problems, but they may perform significantly worst than standard SVM in terms of predictive accuracy. Our proposal has the best compromise between classification permeance and efficiency in terms of running times, being the recommended method for medium-size datasets. For small problems, OVO-SVM is suggested, while the method BSGD is recommended for very large datasets.

There are several opportunities for future work. The work of Jenssen et al. [16] for ν-SVM can be adapted to the l_1-norm as presented in this work, while the concept of the center of the configuration can be extended further to other SVM-based classification approaches, such as Second-order cone programming Support Vector Machine (SOCP-SVM) [20,25]. Additionally, the proposed approach presents an interesting property for classification on highly imbalanced datasets, namely the minimization of the distances between the reduced convex hulls. Given that each training pattern is studied separately, adaptations of the model can be performed on others to favor relevant classes that could be less represented in the training sample. Similar attempts have been done recently for SOCP-SVM [21].

Conflict of Interest

None declared.

Acknowledgments

The authors are grateful to the anonymous reviewers who contributed to improving the quality of the original paper. The first author was supported by FONDECYT Project 1130905, the second by FONDECYT Project 11110188 and CONICYT Anillo ACT1106, and the third by FONDECYT Project 11121196.

Appendix A. Proof of Proposition 3.1

Proof. The Lagrangian function for (DL^p) is given by

$$L(\mathbf{w}_k, b_k, \boldsymbol{\xi}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k, \boldsymbol{\beta}_k, \mathbf{s}_k, \boldsymbol{\gamma}) = \sum_{k=1}^{K} \Big[(b_k + C\boldsymbol{\xi}_k^\top \mathbf{e}) - \boldsymbol{\mu}_k^\top (X_k^\top \mathbf{w}_k + b_k \mathbf{e} + \boldsymbol{\xi}_k) - \boldsymbol{\alpha}_k (\mathbf{w}_k + \mathbf{e}) - \boldsymbol{\beta}_k^\top (\mathbf{e} - \mathbf{w}_k) - \mathbf{s}_k^\top \boldsymbol{\xi}_k + \boldsymbol{\gamma}^\top \mathbf{w}_k \Big].$$

Then, the KKT conditions of the linear programming problem (DL^p) are given by

$$\nabla_{\mathbf{w}_k} L = -X_k \boldsymbol{\mu}_k - \boldsymbol{\alpha}_k + \boldsymbol{\beta}_k + \boldsymbol{\gamma} = \mathbf{0}, \quad \frac{\partial L}{\partial b_k} = 1 - \boldsymbol{\mu}_k^\top \mathbf{e} = 0, \tag{A.1}$$

$$\nabla_{\boldsymbol{\xi}_k} L = C\mathbf{e} - \boldsymbol{\mu}_k - \mathbf{s}_k = \mathbf{0}, \quad \sum_k \mathbf{w}_k = \mathbf{0}, \tag{A.2}$$

$$\boldsymbol{\mu}_k^\top (X_k^\top \mathbf{w}_k + b_k \mathbf{e} + \boldsymbol{\xi}_k) = 0, \quad \mathbf{s}_k^\top \boldsymbol{\xi}_k = 0, \tag{A.3}$$

$$\boldsymbol{\beta}_k^\top (\mathbf{e} - \mathbf{w}_k) = 0, \quad \boldsymbol{\alpha}_k^\top (\mathbf{w}_k + \mathbf{e}) = 0, \tag{A.4}$$

$$X_k^\top \mathbf{w}_k + b_k \mathbf{e} + \boldsymbol{\xi}_k \geq \mathbf{0}, \quad \boldsymbol{\mu}_k \geq \mathbf{0}, \tag{A.5}$$

$$-\mathbf{e} \leq \mathbf{w}_k \leq \mathbf{e}, \quad \boldsymbol{\alpha}_k \geq \mathbf{0}, \quad \boldsymbol{\beta}_k \geq \mathbf{0}, \quad \boldsymbol{\xi}_k \geq \mathbf{0}, \quad \mathbf{s}_k \geq \mathbf{0}. \tag{A.6}$$

From (A.3) and the second expression of (A.1), we get

$$\mathbf{w}_k^\top X_k \boldsymbol{\mu}_k + b_k + \boldsymbol{\mu}_k^\top \boldsymbol{\xi}_k = 0.$$

Multiplying the first equality of (A.1) by $\mathbf{w}_k$ and replacing in the above equality one has

$$b_k = \mathbf{w}_k^\top (\boldsymbol{\alpha}_k - \boldsymbol{\beta}_k - \boldsymbol{\gamma}) - \boldsymbol{\mu}_k^\top \boldsymbol{\xi}_k.$$

Summing over i in the above equality and using (A.2) and (A.4), we have

$$\sum_{k=1}^{K} b_k = \sum_{k=1}^{K} (\mathbf{w}_k^\top (\boldsymbol{\alpha}_k - \boldsymbol{\beta}_k) - \boldsymbol{\mu}_k^\top \boldsymbol{\xi}_k) = -\sum_{k=1}^{K} ((\boldsymbol{\alpha}_k + \boldsymbol{\beta}_k)^\top \mathbf{e} + \boldsymbol{\mu}_k^\top \boldsymbol{\xi}_k), \tag{A.7}$$

from which we deduce that $\sum_{k=1}^{K} b_k \leq 0$.

In a similar way, we define the Lagrangian function associated with (23) as

$$\hat{L}(\hat{\mathbf{w}}_k, \hat{b}_k, \hat{\boldsymbol{\xi}}_k, r, \boldsymbol{\xi}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\alpha}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\mathbf{s}}_k, \hat{\boldsymbol{\gamma}}, t, \theta) = -r + \sum_{k=1}^{K} \Big[C\hat{\boldsymbol{\xi}}_k^\top \mathbf{e} - \hat{\boldsymbol{\mu}}_k^\top (X_k^\top \hat{\mathbf{w}}_k + \hat{b}_k \mathbf{e}) - \hat{\boldsymbol{\mu}}_k^\top (-r\mathbf{e} + \hat{\boldsymbol{\xi}}_k) - \hat{\boldsymbol{\alpha}}_k (\hat{\mathbf{w}}_k + \mathbf{e}) - rt - \hat{\boldsymbol{\beta}}_k^\top (\mathbf{e} - \hat{\mathbf{w}}_k) - \hat{\mathbf{s}}_k^\top \hat{\boldsymbol{\xi}}_k + \hat{\boldsymbol{\gamma}}^\top \hat{\mathbf{w}}_k + \theta \hat{b}_k \Big].$$

Then, the KKT conditions for Problem (23) are the following:

$$\nabla_{\hat{\mathbf{w}}_k} \hat{L} = -X_k \hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\alpha}}_k + \hat{\boldsymbol{\beta}}_k + \hat{\boldsymbol{\gamma}} = \mathbf{0}, \quad \frac{\partial \hat{L}}{\partial b_k} = \theta - \hat{\boldsymbol{\mu}}_k^\top \mathbf{e} = 0, \tag{A.8}$$

$$\nabla_{\hat{\boldsymbol{\xi}}_k} \hat{L} = C\mathbf{e} - \hat{\boldsymbol{\mu}}_k - \hat{\mathbf{s}}_k = \mathbf{0}, \quad \frac{\partial \hat{L}}{\partial r} = -1 + \hat{\boldsymbol{\mu}}_k^\top \mathbf{e} - t = 0, \tag{A.9}$$

$$\hat{\boldsymbol{\mu}}_k^\top (X_k^\top \hat{\mathbf{w}}_k + \hat{b}_k \mathbf{e} - r\mathbf{e} + \hat{\boldsymbol{\xi}}_k) = 0, \quad \hat{\mathbf{s}}_k^\top \hat{\boldsymbol{\xi}}_k = 0, \quad rt = 0, \tag{A.10}$$

$$\hat{\boldsymbol{\beta}}_k^\top (\mathbf{e} - \hat{\mathbf{w}}_k) = 0, \quad \hat{\boldsymbol{\alpha}}_k (\hat{\mathbf{w}}_k + \mathbf{e}) = 0, \tag{A.11}$$

$$X_k^\top \hat{\mathbf{w}}_k + \hat{b}_k \mathbf{e} - r\mathbf{e} + \hat{\boldsymbol{\xi}}_k \geq \mathbf{0}, \quad \hat{\boldsymbol{\mu}}_k \geq \mathbf{0}, \quad r \geq 0, \tag{A.12}$$

$$-\mathbf{e} \leq \hat{\mathbf{w}}_k \leq \mathbf{e}, \quad \hat{\boldsymbol{\alpha}}_k \geq \mathbf{0}, \quad \hat{\boldsymbol{\beta}}_k \geq \mathbf{0}, \quad \hat{\boldsymbol{\xi}}_k \geq \mathbf{0}, \quad \hat{\mathbf{s}}_k \geq \mathbf{0}, \tag{A.13}$$

$$\sum_{k=1}^{K} \hat{\mathbf{w}}_k = \mathbf{0}, \quad \sum_{k=1}^{K} \hat{b}_k = 0, \quad t \geq 0. \tag{A.14}$$

We note that $\theta \geq 1$, as a direct consequence of the second equality in (A.8)–(A.9) and the feasibility (A.14). Additionally, from (A.8)–(A.10)

one has $r(\theta-1)=0$. If we assume $r>0$, then $\theta=1$, and therefore

$$\mathbf{w}_k=\hat{\mathbf{w}}_k,\quad b_k=\hat{b}_k-r,\quad \boldsymbol{\xi}_k=\hat{\boldsymbol{\xi}},\quad \boldsymbol{\mu}_k=\hat{\boldsymbol{\mu}}_k,\quad \boldsymbol{\alpha}_k=\hat{\boldsymbol{\alpha}}_k,\quad \boldsymbol{\beta}_k=\hat{\boldsymbol{\beta}}_k,\quad \mathbf{s}_k=\hat{\mathbf{s}}_k,\quad \boldsymbol{\gamma}=\hat{\boldsymbol{\gamma}},$$

satisfy the KKT conditions of the problem (DL^p).

On the other hand, from (A.10) and the second equality of (A.8), we have

$$\hat{\mathbf{w}}_k^\top X_k\hat{\boldsymbol{\mu}}_k+\theta(\hat{b}_k-r)+\boldsymbol{\mu}_k^\top\hat{\boldsymbol{\xi}}_k=0.$$

Multiplying the first equality of (A.8) by $\mathbf{w}_k$ and replacing in the above equality we get

$$\theta(r-\hat{b}_k)=\hat{\mathbf{w}}_k^\top(-\hat{\boldsymbol{\alpha}}_k+\hat{\boldsymbol{\beta}}_k+\hat{\boldsymbol{\gamma}})+\boldsymbol{\mu}_k^\top\hat{\boldsymbol{\xi}}_k$$

Summing the above equality over $k=1,\ldots,K$ and using (A.11) and (A.14), we have

$$Kr\theta=\sum_{k=1}^{K}(\hat{\mathbf{w}}_k^\top(-\hat{\boldsymbol{\alpha}}_k+\hat{\boldsymbol{\beta}}_k)+\hat{\boldsymbol{\mu}}_k^\top\hat{\boldsymbol{\xi}}_k)=\sum_{k=1}^{K}(\hat{\boldsymbol{\alpha}}_k^\top\mathbf{e}+\hat{\boldsymbol{\beta}}_k^\top\mathbf{e}+\hat{\boldsymbol{\mu}}_k^\top\hat{\boldsymbol{\xi}}_k).\quad(A.15)$$

If we assume that $r=0$, using the above expression, we deduce that

$$\hat{\boldsymbol{\alpha}}_k=\mathbf{0},\quad \hat{\boldsymbol{\beta}}_k=\mathbf{0},\quad \hat{\boldsymbol{\mu}}_k^\top\hat{\boldsymbol{\xi}}_k=0,\quad k=1,\ldots,K.$$

Now, multiplying the first equality of (A.9) by $\hat{\boldsymbol{\xi}}_k$, using the above relation and (A.10), we get

$$C\hat{\boldsymbol{\xi}}_k^\top\mathbf{e}=0,\quad k=1,\ldots,K,$$

therefore $\hat{\boldsymbol{\xi}}_k=0$, for $k=1,\ldots,K$. Finally, taking

$$\mathbf{w}_k=\hat{\mathbf{w}}_k,\quad b_k=\hat{b}_k,\quad \boldsymbol{\xi}_k=\mathbf{0},\quad \boldsymbol{\mu}_k=\frac{\hat{\boldsymbol{\mu}}_k}{\theta},\quad \boldsymbol{\alpha}_k=\mathbf{0},\quad \boldsymbol{\beta}_k=\mathbf{0},$$
$$\mathbf{s}_k=\left(1-\frac{1}{\theta}\right)\hat{\boldsymbol{\mu}}_k+\frac{1}{\theta}\hat{\mathbf{s}}_k,\quad \boldsymbol{\gamma}=\frac{\hat{\boldsymbol{\gamma}}}{\theta},\quad(A.16)$$

satisfy the KKT conditions of the problem (DL^p).

Similarly, supposing that $(\mathbf{w}_k,b_k,\boldsymbol{\xi}_k,\boldsymbol{\mu}_k,\boldsymbol{\alpha}_k,\boldsymbol{\beta}_k,\mathbf{s}_k,\boldsymbol{\gamma})$ satisfies the KKT conditions of the problem (DL^p). Take $\theta=1$ and let us define

$$r=-\frac{1}{K}\sum_{k=1}^{K}b_k\quad(A.17)$$

From (A.7), we obtain that $r\geq 0$ and therefore

$$\hat{\mathbf{w}}_k=\mathbf{w}_k,\quad \hat{b}_k=b_k+r,\quad \hat{\boldsymbol{\xi}}_k=\boldsymbol{\xi},\quad \hat{\boldsymbol{\mu}}_k=\boldsymbol{\mu}_k,\quad \hat{\boldsymbol{\alpha}}_k=\boldsymbol{\alpha}_k,\quad \hat{\boldsymbol{\beta}}_k=\boldsymbol{\beta}_k,$$
$$\hat{\mathbf{s}}_k=\mathbf{s}_k,\quad \hat{\boldsymbol{\gamma}}=\boldsymbol{\gamma},\quad t=0,\quad \theta=1,\quad(A.18)$$

satisfy the relations (A.8)–(A.14). □

References

[1] R. Ñanculef, C. Concha, H. Allende, D. Candel, C. Moraga, Ad-svms: a light extension of svms for multicategory classification, Int. J. Hybrid Intell. Syst. 6 (2009) 69–79.

[2] S. Armstrong, J. Staunton, L. Silverman, R. Pieters, M. den Boer, M. Minden, S. Sallan, E. Lander, T. Golub, S. Korsmeyer, MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia, Nat. Genet. 30 (2002) 41–47.

[3] A. Asuncion, D. Newman, UCI Machine Learning Repository, 2007.

[4] K. Bennett, E. Bredensteiner, Duality and geometry in svm classifiers, in: Proceedings of 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, 2000, pp. 57–64.

[5] C. Bhattacharyya, Second order cone programming formulations for feature selection, J. Mach. Learn. Res. 5 (2004) 1417–1433.

[6] 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⟩.

[7] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, J. Mach. Learn. Res. 2 (2001) 265–292.

[8] D.J. Crisp, C. Burges, A geometric interpretation of v-svm classifiers, in: Advances in Neural Information Processing Systems, The MIT Press, 1999, pp. 244–250.

[9] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, New York, 2000.

[10] N. Djuric, L. Lan, S. Vucetic, Z. Wang, Budgetedsvm: a toolbox for scalable svm approximations, J. Mach. Learn. Res. 14 (2013) 3813–3817.

[11] T.A. Ferguson, Bayesian analysis of some nonparametric problems, Ann. Stat. 1 (1973) 209–230.

[12] R. Fletcher, Practical Methods of Optimization, John Wiley and Sons, New York, 1989.

[13] J. Friedman, Another Approach to Polychotomous Classification, Technical Report, Department of Statistics, Stanford University, 1996.

[14] 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 (2013) 3412–3424.

[15] R. Jenssen, M. Kloft, A. Zien, S. Sonnenburg, K. Müller, A Multi-Class Support Vector Machine Based on Scatter Criteria, Technical Report 014-2009, Technische Universität Berlin, 2009.

[16] R. Jenssen, M. Kloft, A. Zien, S. Sonnenburg, K. Müller, A scatter-based prototype framework and multi-class extension of support vector machines, PLoS ONE 7 (2012) e42947.

[17] U.G. Kressel, Advances in Kernel Methods, MIT Press, Cambridge, MA, USA (1999) 255–268.

[18] Y. Lee, Y. Lin, W. G, Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, J. Am. Stat. Assoc. 99 (2004) 67–81.

[19] D.G. Luenberger, Linear and Nonlinear Programming, 2nd ed., Kluwer Academic Publishers, Boston, MA, 2003.

[20] S. Maldonado, J. López, Alternative second-order cone programming formulations for support vector classification, Inf. Sci. 268 (2014) 328–341.

[21] S. Maldonado, J. López, Imbalanced data classification using second-order cone programming support vector machines, Pattern Recognit. 47 (2014) 2070–2079.

[22] S. Maldonado, R. Weber, J. Basak, Kernel-penalized SVM for feature selection, Inf. Sci. 181 (2011) 115–128.

[23] J. Mercer, Functions of positive and negative type, and their connection with the theory of integral equations, Philos. Trans. R. Soc. Lond. 209 (1909) 415–446.

[24] M. Momma, K. Hatano, H. Nakayama, Ellipsoidal support vector machines, in: Proceedings of the Asian Conference on Machine Learning (ACML-10), 2010.

[25] 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.

[26] C. Nutt, D. Mani, R. Betensky, P. Tamayo, J. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. McLaughlin, T. Batchelor, P. Black, A. von Deimling, S. Pomeroy, T. Golub, D. Louis, Gene expression-based classification of malignant gliomas correlates better with survival than histological classification, Cancer Res. 63 (2003) 1602–1607.

[27] J. Platt, Advances in Kernel Methods-Support Vector Learning, MIT Press, Cambridge, MA (1999) 185–208.

[28] B. Schölkopf, A.J. Smola, Learning with Kernels, MIT Press, Cambridge, 2002.

[29] S. Shalev-Shwartz, Y. Singer, N. Srebro, A. Cotter, Pegasos: primal estimated sub-gradient solver for svm, Math. Programm. 127 (2011) 3–30.

[30] R. Strack, V. Kecman, B. Strack, Q. Li, Sphere support vector machines for large classification tasks, Neurocomputing 101 (2013) 59–67.

[31] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.

[32] Z. Wang, K. Crammer, S. Vucetic, Multi-class pegasos on a budget, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress, 2010, pp. 1143–1150.

[33] Z. Wang, K. Crammer, S. Vucetic, Breaking the curse of kernelization: budgeted stochastic gradient descent for large-scale SVM training, J. Mach. Learn. Res. 13 (2012) 3103–3131.

[34] Z. Wang, N. Djuric, K. Crammer, S. Vucetic, Trading representability for scalability: adaptive multi-hyperplane machine for nonlinear classification, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 24–32.

[35] J. Weston, A. Elisseeff, G. Baklr, F. Sinz, The Spider Machine Learning Toolbox, 2005.

[36] J. Weston, C. Watkins, Multi-class support vector machines, in: Proceedings of the Seventh European Symposium on Artificial Neural Networks, 1999.

[37] W. Zhou, L. Zhang, L. Jiao, Linear programming support vector machines, Pattern Recognit. 35 (2002) 2927–2936.

[38] J. Zhu, N. Chen, E.P. Xing, Infinite svm: a Dirichlet process mixture of large-margin kernel machines, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 617–624.

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

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 Modeling 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.

Sebastián Maldonado received his B.S. and M.S. degrees from the University of Chile, in 2007, and his Ph.D. degree from the University of Chile, in 2011. He is currently a 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.