Pattern Recognition 47 (2014) 2070–2079

Contents lists available at ScienceDirect

Pattern Recognition

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

Imbalanced data classification using second-order cone programming support vector machines

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

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

A R T I C L E I N F O

Article history:
Received 7 June 2013
Received in revised form
2 October 2013
Accepted 23 November 2013
Available online 3 December 2013

Keywords:
Class-imbalanced data
Support Vector Machines
LP-SVM
SOCP-SVM

A B S T R A C T

Learning from imbalanced data sets is an important machine learning challenge, especially in Support Vector Machines (SVM), where the assumption of equal cost of errors is made and each object is treated independently. Second-order cone programming SVM (SOCP-SVM) studies each class separately instead, providing quite an interesting formulation for the imbalanced classification task. This work presents a novel second-order cone programming (SOCP) formulation, based on the LP-SVM formulation principle: the bound of the VC dimension is loosened properly using the l_∞-norm, and the margin is directly maximized using two margin variables associated with each class. A regularization parameter C is considered in order to control the trade-off between the maximization of these two margin variables. The proposed method has the following advantages: it provides better results, since it is specially designed for imbalanced classification, and it reduces computational complexity, since one conic restriction is eliminated. Experiments on benchmark imbalanced data sets demonstrate that our approach accomplishes the best classification performance, compared with the traditional SOCP-SVM formulation and with cost-sensitive formulations for linear SVM.

1. Introduction

The class imbalance problem is a relatively new challenge that has attracted growing attention in both industry and academia, since it negatively affects classification performance. This issue arises when the class distribution is too skewed [37]. Technically speaking, any data set with unequal distribution between the two classes can be considered imbalanced. However, class ratios of 5:1 or higher have often been used in experiments under the category of imbalanced data sets [14]. When this situation occurs, standard classification methods such as Support Vector Machines (SVM) will generate a model that predicts everything to the majority class. Attempts have been made to deal with this problem in the context of business analytics, such as churn prediction [17], credit scoring [27], and fraud detection [12]; and also various domains such as text categorization [39], spam filtering [33] and anomaly detection [24].

Recently, second-order cone programming (SOCP) formulations have been proposed as an alternative optimization scheme for SVM. These formulations consider all possible choices of class-conditional densities with a given mean and covariance matrix, that is, in a worst-case setting, and hence they avoid making assumptions about the class-conditional densities, which could cast the generality and validity of such an approach in doubt [3,7]. Moreover, these formulations provide a cost-sensitive framework for handling uneven misclassification costs in binary classification [22], for instance, in medical diagnosis. These special types of non-linear convex optimization problems can be solved efficiently by interior point algorithms [2,3].

This work presents a novel second-order cone programming (SOCP) formulation based on the LP-SVM principle: the bound of the VC dimension is loosened properly using the l_∞-norm, and the margin is maximized directly using two margin variables associated with each class. A regularization parameter, C, is included to control the trade-off between the maximization of these two margin variables.

This paper is organized as follows: in Section 2, we briefly introduce the methods Support Vector Machines, LP-SVM and SOCP-SVM for binary classification. Section 3 provides an overview of the class imbalance problem. Section 4 presents the proposed SOCP-SVM formulation for imbalanced data classification. Experimental results using benchmark data sets are given in Section 5. A summary of this paper can be found in Section 6, where we provide its main conclusions and address future developments.

2. Support Vector Machines for binary classification

This section introduces SVM for binary classification, as developed by Vapnik [35] for hard margin and Cortes and Vapnik [10] for soft margin, and its variation based on second-order cone

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

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.11.021

programming [22,29]. Given a set of tuples $(\mathbf{x}_i, y_i)$ of training examples and their respective labels, where $\mathbf{x}_i \in \Re^n$, $i=1,\dots,m$ and $y_i \in \{-1,+1\}$, the linear version of SVM aims at finding the *maximum-margin hyperplane*, i.e., it finds a classifier of the form $f(\mathbf{x}) = \mathbf{w}^\top \cdot \mathbf{x} + b$ that maximizes the distance from it to the nearest training point on each class. To maximize this measure, SVM minimizes the Euclidean norm of coefficients $\mathbf{w}$ [35]. Additionally, we want to classify the training vectors $\mathbf{x}_i$ correctly into two different classes y_i. For linearly separable problems, this can be formulated as follows (*hard margin* SVM formulation):

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

Notice that if there is no hyperplane that can split both classes, formulation (1) becomes unfeasible. Cortes and Vapnik [10] suggested a modified formulation that allows misclassification by balancing the structural (minimization of the Euclidean norm), and the empirical risk (minimization of misclassification errors) by introducing slack variables ξ_i, $i=1,\dots,m$, which measure the degree of misclassification for an instance $\mathbf{x}_i$, and a penalty parameter C that controls this trade-off. For a linear penalty function, the *soft margin* SVM formulation becomes

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

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

Formulation (2) can be solved efficiently in the dual space using the Sequential Minimal Optimization (SMO) technique [25], among others. Some studies have also been proposed for solving the primal SVM formulation efficiently, using a Newton-based algorithm [21] or back-fitting strategies [23], for example.

2.1. Linear programming Support Vector Machine

In linear programming SVM, the bound of the VC dimension is loosened properly, using the l_∞-norm, to improve the speed of the training time [40, Theorem 2.2], resulting in a linear programming formulation that controls the margin maximization directly by including a margin variable r. This variable is then maximized while simultaneously minimizing the empirical risk, by penalizing the slack variables similar to that in standard SVM. The LP-SVM soft-margin formulation follows:

$$\begin{aligned} \min_{\mathbf{w},r,b,\xi} \quad & -r + C\sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y_i \cdot (\mathbf{w}^\top \cdot \mathbf{x}_i + b) \geq r - \xi_i, \quad i=1,\dots,m, \\ & -1 \leq w_j \leq 1, \quad j=1,\dots,n, \\ & \xi_i \geq 0, \quad i=1,\dots,m, \\ & r \geq 0. \end{aligned} \tag{3}$$

The formulation (3) is simpler than the one in (2), especially for large-scale problems. The dual formulation associated with (3) can be stated as follows (see Appendix A for details):

$$\begin{aligned} \min_{\mathbf{z}_1,\mathbf{z}_2} \quad & \|A\mathbf{z}_1 - B\mathbf{z}_2\|_1 \\ \text{s.t.} \quad & \mathbf{e}^\top \cdot \mathbf{z}_1 = \mathbf{e}^\top \cdot \mathbf{z}_2 = 1, \\ & 0 \leq \mathbf{z}_1 \leq C\mathbf{e}, \quad 0 \leq \mathbf{z}_2 \leq C\mathbf{e}. \end{aligned} \tag{4}$$

Geometrically, this means that the optimization problem (3) is equivalent to finding the closest points on the reduced convex hulls formed by the positive and negative labeled data points, by using 1-norm (see [6] for alternative norm variations).

2.2. SOCP Support Vector Machines

Suppose that $\mathbf{X}_1$ and $\mathbf{X}_2$ are the random vectors that generate samples of the positive and negative classes respectively, with means and covariance matrices given by $(\boldsymbol{\mu}_i, \Sigma_i)$ for $i=1,2$, where $\Sigma_i \in \Re^{n\times n}$ are symmetric positive semi-definite matrices.

In order to construct a maximum margin linear classifier, so that the probability of false-negative and false-positive errors does not exceed $\eta_1 \in (0,1]$ and $\eta_2 \in (0,1]$ respectively, Nath and Bhattacharyya [22] suggested considering the following quadratic chance-constrained programming problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \Pr\{\mathbf{w}^\top \cdot \mathbf{X}_1 - b \geq 0\} \geq \eta_1, \\ & \Pr\{\mathbf{w}^\top \cdot \mathbf{X}_2 - b \leq 0\} \geq \eta_2. \end{aligned} \tag{5}$$

In other words, the model requires that the random variable $\mathbf{X}_i$ lies on the correct side of the hyperplane, with greater probability than η_k for $k=1,2$. In this case, we want to be able to classify for each training pattern correctly, up to the rate η_k, even for the *worst distribution* in the class of the ones that have common mean and covariance $\mathbf{X}_k \sim (\boldsymbol{\mu}_k, \Sigma_k)$. For this purpose, the probability constraints in (5) are replaced by their *robust* counterparts:

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

Thanks to an appropriate application of the multivariate Chebyshev inequality [18, Lemma 1], this worst distribution approach leads to the following deterministic problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \mathbf{w}^\top \cdot \boldsymbol{\mu}_1 - b \geq 1 + \kappa_1 \|S_1^\top \mathbf{w}\|, \\ & b - \mathbf{w}^\top \cdot \boldsymbol{\mu}_2 \geq 1 + \kappa_2 \|S_2^\top \mathbf{w}\|, \end{aligned} \tag{6}$$

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

By introducing a new variable t and a constraint $\|\mathbf{w}\| \leq t$, the formulation (6) can be casted as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & t \\ \text{s.t.} \quad & \|\mathbf{w}\| \leq t \\ & \mathbf{w}^\top \cdot \boldsymbol{\mu}_1 - b \geq 1 + \kappa_1 \|S_1^\top \mathbf{w}\|, \\ & b - \mathbf{w}^\top \cdot \boldsymbol{\mu}_2 \geq 1 + \kappa_2 \|S_2^\top \mathbf{w}\|. \end{aligned} \tag{7}$$

This problem is an instance of linear SOCP with three blocks [2]. A linear SOCP problem is a convex optimization problem with a linear objective function, and second-order cone (SOC) constraints. An SOC constraint on the variable $\mathbf{x} \in \Re^n$ is of the form

$$\|A\mathbf{x} + \mathbf{b}\| \leq \mathbf{c}^\top \cdot \mathbf{x} + d, \tag{8}$$

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

3. The class imbalance problem

Several developments have been made in the class imbalance problem, mainly in three subareas: data resampling, cost-sensitive learning and one-class learning. Other areas, such as feature selection [14] and extraction [36] have been studied. In this section, we briefly describe these topics, referring also to the assessment techniques used to evaluate these tasks at the end of this section.

3.1. Data resampling

The two most common and intuitive data resampling techniques are *random oversampling* and *undersampling*. The first one replicates randomly selected examples of the minority class, while the second discards instances from the majority class randomly, downsizing this class. Both cases help to balance the class distribution, but no new information is added to the data set and this may lead to overfitting [16]. Additionally, oversampling increases the training size, causing longer model training times, while undersampling may lead to an important loss of information [16].

SMOTE [8] is an intelligent oversampling method that generates new examples for the minority class. These are created artificially by interpolating the preexisting minority instances, which may help to improve the classification performance, reducing the risk of overfitting [13,17].

3.2. Cost-sensitive learning

Classification methods can also be trained from imbalanced data sets without data resampling. Cost-sensitive techniques are based on the concept of the cost matrix, which can be considered as a numerical representation of the penalty for misclassification. For example, we define C_- as the cost of misclassifying a majority class instance as a minority one, and let C_+ represent the cost of misclassification in the target class, which is usually higher, i.e., $C_+ > C_-$.

There are strategies for cost-sensitive learning. One group of techniques applies misclassification costs to the data set as a form of data weighting, for example, by introducing costs into the weight updating strategy used in AdaBoost [32]. Other approaches consider cost-sensitive adjustment of different classification methods, which can be applied to the decision threshold or modifying their formulation [14]. In the case of SVM, the total misclassification cost $C\sum_{i=1}^{m}\xi_i$ can be replaced by two terms, one for each class, leading to the following formulation (Cost-sensitive SVM or CS-SVM):

$$\begin{aligned}\min_{\mathbf{w},b,\xi}\quad & \tfrac{1}{2}\|\mathbf{w}\|^2+\mathbf{C}_+\sum_{\mathbf{i}\in\mathbf{I}^+}\xi_i+\mathbf{C}_-\sum_{\mathbf{i}\in\mathbf{I}^-}\xi_i\\ \text{s.t.}\quad & y_i\cdot(\mathbf{w}^\top\cdot\mathbf{x}_i+b)\ge 1-\xi_i,\quad i=1,\dots,m,\\ & \xi_i\ge 0,\quad i=1,\dots,m.\end{aligned}\tag{9}$$

where I^+ and I^- are the sets of positive and negative examples respectively [5,28].

3.3. One-class learning

When negative examples greatly outnumber the positive ones, certain classifiers based on discriminative approaches tend to overfit [9]. In this case, one-class strategies may lead to better predictive performance [26]. A one-class SVM formulation for unbalanced data has been proposed by Tax and Duin [34], namely Support Vector Data Description (SVDD). This method finds the smallest sphere of radius R that contains most of the data instances. Outliers in the training set result in quite a large sphere which will not represent the data very well, therefore slack variables ξ are introduced.

3.4. Assessment metrics for imbalanced classification

Traditionally, the most frequently used metric for binary classification performance is the *accuracy*, which represents the proportion of true results:

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN}\tag{10}$$

where TP=true positives, TN=true negatives, FP=false positives and FN=false negatives. This measure provides a simple way of describing the classification performance. However, it is not appropriate for imbalanced classification [14]. For instance, if a given data set includes 1% of the minority class examples and 99% of majority instances, a naive approach that classifies every example as a majority class instance would provide an accuracy of 99%, which can be considered quite accurate. However, this metric fails to reflect the fact that all target examples are misclassified, which is assumed to have a higher misclassification cost [14].

Alternatively, a few assessment metrics are frequently adopted in the research community for imbalanced learning problems. The most common one is the Area Under the Curve (AUC) [30], in which version defined by one run is explained as follows:

$$AUC=\frac{1+\dfrac{TP}{TP+FN}-\dfrac{FP}{FP+TN}}{2}\tag{11}$$

We consider this metric, which is widely known as balanced accuracy [30], to be the main performance measure in this work.

4. The proposed SOCP-SVM approach for imbalanced data classification

In this section, a cost-sensitive formulation for SOCP-SVM is presented. The reasoning behind this approach is that we can improve classification performance in unbalanced data by controlling the distance to both training patterns separately. The main idea is to adapt the LP-SVM formulation to a robust one based on second-order cones, and, in a second step, to modify it by splitting the problem into two margin variables.

Let us consider the following linear chance-constrained programming problem:

$$\begin{aligned}\min_{\mathbf{w},b,r}\quad & -r\\ \text{s.t.}\quad & \inf_{\mathbf{X}_1\sim(\boldsymbol{\mu}_1,\Sigma_1)}\text{Prob}\{\mathbf{w}^\top\cdot\mathbf{X}_1\ge b+r\}\ge\eta_1,\\ & \inf_{\mathbf{X}_2\sim(\boldsymbol{\mu}_2,\Sigma_2)}\text{Prob}\{\mathbf{w}^\top\cdot\mathbf{X}_2\le b-r\}\ge\eta_2,\\ & -1\le w_j\le 1,\quad j=1,\dots,n.\\ & r\ge 0.\end{aligned}\tag{12}$$

Formulation 12 relates to LP-SVM since the Euclidean norm minimization is replaced by a margin variable r, maximized in the objective function, while the chance constraints impose the probability that the random variable $\mathbf{X}_i$ lies on the correct side of the hyperplane, including variable r, does not exceed η_i for i=1,2. Equal to the derivation of SOCP-SVM, the problem (12) can be stated as the following linear SOCP formulation, thanks to an appropriate application of the multivariate Chebyshev inequality:

$$\begin{aligned}\min_{\mathbf{w},b,r}\quad & -r\\ \text{s.t.}\quad & \mathbf{w}^\top\cdot\boldsymbol{\mu}_1-b\ge r+\kappa_1\|S_1^\top\mathbf{w}\|,\\ & b-\mathbf{w}^\top\cdot\boldsymbol{\mu}_2\ge r+\kappa_2\|S_2^\top\mathbf{w}\|,\\ & -1\le w_j\le 1,\quad j=1,\dots,n.\\ & r\ge 0,\end{aligned}\tag{13}$$

where $\Sigma_i=S_iS_i^\top$ and $\kappa_i=\sqrt{\eta_i/(1-\eta_i)}$ for i=1,2. We refer to the above formulation as r-SOCP-SVM.

Comparing SOCP-SVM (Formulation (6)) with the previous one, the latter replaces the Euclidean norm minimization for a (positive) margin variable r, which works as a slack variable for the conic constraints (the first two in Formulation 13). Additional constraints are introduced in order to bound the weights. This approach has the advantage that only two conic constraints are needed for classification instead of three, as in the case of the standard SOCP-SVM.

As we described previously in this section, the main idea is to adapt the previous formulation to imbalanced classification. The margin variable r is then replaced by r_1 and r_2, one for each conic constraint, and simultaneously maximized in the objective function. The trade-off between both values is managed by a positive parameter C:

$$\begin{aligned} \min_{\mathbf{w},b,r_1,r_2} \quad & -r_1 - Cr_2 \\ \text{s.t.} \quad & \mathbf{w}^\top \cdot \boldsymbol{\mu}_1 - b \geq r_1 + \kappa_1 \| S_1^\top \mathbf{w} \|, \\ & b - \mathbf{w}^\top \cdot \boldsymbol{\mu}_2 \geq r_2 + \kappa_2 \| S_2^\top \mathbf{w} \|, \\ & -1 \leq w_j \leq 1, \quad j = 1, \dots, n, \\ & r_1, r_2 \geq 0. \end{aligned} \tag{14}$$

We refer to formulation (14) as $r_1 r_2 - \text{SOCP} - \text{SVM}$, which is inspired by the cost-sensitive SVM formulation for imbalanced data (cf. (9)). The penalty parameter C can be set using crossvalidation, and relates to the costs of misclassification for both classes. The advantage of this formulation compared to CS-SVM is that only two objectives are considered instead of three, making the model selection step more tractable.

The dual formulation associated with $r_1 r_2 - \text{SOCP} - \text{SVM}$ (see Appendix B for derivation) is given by

$$\begin{aligned} \min_{\mathbf{z}_1,\mathbf{z}_2} \quad & \hat{\lambda} \| \mathbf{z}_1 - \mathbf{z}_2 \|_1 \\ \text{s.t.} \quad & \mathbf{z}_i \in \mathbf{B}_i(\boldsymbol{\mu}_i, S_i, \kappa_i), \quad i = 1, 2, \end{aligned} \tag{15}$$

where

$$\mathbf{B}_i(\boldsymbol{\mu}_i, S_i, \kappa_i) = \{ \mathbf{z}_i : \mathbf{z}_i = \boldsymbol{\mu}_i + (-1)^i S_i \mathbf{u}_i, \ \| \mathbf{u}_i \| \leq 1 \}, \quad i = 1, 2. \tag{16}$$

The sets $\mathbf{B}_i(\boldsymbol{\mu}_i, S_i, \kappa_i)$ are ellipsoids centered at $\boldsymbol{\mu}_i$, whose shape is determined by S_i and sized by κ_i. Thus, the dual formulation minimizes the distance between two ellipsoids using the 1-norm.

5. Experimental results

We applied the r-SOCP-SVM and $r_1 r_2 - \text{SOCP} - \text{SVM}$ approaches on class-imbalanced data sets to assess their performance compared to well-known SVM-based classification methods. The main goal was to study whether our approaches perform better in terms of AUC compared to SVM in its standard version, to the cost-sensitive version of this approach, and finally to SOCP-SVM. The importance of this comparison rests on having a broad enough range of classification approaches to verify if the proposed modifications have a positive effect on the classification performance, compared to the original methods. In this context, the comparison with CS-SVM, a method designed for the class imbalance problem, and SOCP-SVM, whose design provides a natural way to adequately obtain balanced classifiers, is particularly important.

We first provide a description of the data sets in Section 5.1, while Section 5.2 presents a summary of the predictive performance obtained for the proposed and alternative approaches. Finally, an empirical study regarding the influence of the different parameters and an extended discussion of the results are presented in Section 5.3.

5.1. Description of data sets and validation procedure

The proposed approach has been applied to six benchmark data sets from the UCI data repository [4]. Two data sets are class-imbalanced binary-classification problems, while the remaining four are adapted multiclass classification problems, in which the target and majority classes were constructed by grouping some of the labels, as described in [1] and [11]. Next, we briefly describe these data sets:

- *Ecoli*: This data set studies the localization site of the E. coli protein in eukaryotic cells. The original data set from the UCI Repository is studied since it is a natural two-class imbalanced problem.
- *Abalone_7*: This data set studies the classification of 29 types of Abalone, according to variables such as sex, length, diameter, and weight. We study class 7 against all other Abalone types.
- *Balance*: The Balance Scale Weight and Distance data base were generated to model this psychological phenomena. Each example is classified as having the balance scale tipped to the right, the left, or in balance. We study the balanced scale class against both the tipped left and right scale.
- *Car_34*: The Car Evaluation data set studies the four levels of acceptability of various used cars. Classes 3 (good) and 4 (very good) are studied together against the other classes.
- *Solar_M*: The Solar Flare data set studies the number of times a certain type of solar flare occurs in 1 day. We considered M-class flares (moderate ones), and two classes were obtained by studying zero M-class flares in 24 h against one or more.
- *Yeast_ME2*: This multi-class data set also considers the problem of protein localization, and class ME2 (membrane protein with cleaved signal) was studied as the target class against all the remaining labels.

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

In this section, we study the following classification approaches: SVM in its standard version, cost-sensitive SVM, and the proposed approaches r-SOCP-SVM (Formulation (13)) and $r_1 r_2$-SOCP-SVM (Formulation (14)). The following model selection procedure was performed: training and test subsets were constructed using 10-fold crossvalidation, the average accuracy (proportion of true results, Eq. (10)) and the AUC was computed.

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

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

For the above procedure, we used the Spider Toolbox for Matlab [38] for standard SVM approaches, and the SeDuMI Matlab Toolbox for SOCP-based classifiers [31].

5.2. Summary of classification results for imbalanced datasets

In this section, we present a summary of the results obtained from our experiments to facilitate assessing the best performance of the respective approaches. Table 2 summarizes the best performance (using the AUC measure) of all the approaches along the different values of C and η for all six data sets.

In previously mentioned table, we observe that the best predictive results were achieved using $r_1 r_2$-SOCP-SVM in four out of six data sets, while r-SOCP-SVM, SOCP-SVM and CS-SVM

Table 1
Number of variables, number of examples, percentage of each class and imbalanced ratio for all six data sets.

Dataset	# variables	# examples (min.,maj.)	IR
Ecoli	7	(35,301)	8.6
Abalone_7	8	(391,3786)	9.7
Balance	4	(49,576)	11.8
Car_34	6	(134,1594)	11.9
Solar_M	10	(68,1321)	19.4
Yeast_ME2	8	(51,1433)	28.1

Table 2
Mean AUC, in percentage, for all data sets.

	Ecoli	Abalone_7	Balance	Car_34	Solar_M	Yeast_ME2
SVM	78.6	50.0	50.0	98.3	50.0	50.0
CS-SVM	73.8	50.1	50.0	**98.7**	50.6	50.0
SOCP-SVM	89.2	78.4	71.6	98.3	**73.4**	85.0
r-SOCP-SVM	90.0	77.9	**74.6**	98.5	72.0	84.7
$r_1r_2-\mathrm{SOCP}-\mathrm{SVM}$	**90.8**	**78.7**	74.1	**98.7**	71.6	**85.1**

Table 3
Max, Min and Mean AUC along different values of η, and t test for model selection stability, for all data sets.

	Ecoli	Abalone_7	Balance	Car_34	Solar_M	Yeast_ME2
SOCP-SVM						
Max	89.2	78.4	71.6	98.3	73.4	85.0
Min	82.4	67.4	33.9	79.9	60.8	57.3
Mean	87.0	75.0	49.2	93.5	67.7	77.7
p value	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
r-SOCP-SVM						
Max	90.0	77.9	74.6	98.5	72.0	84.7
Min	80.3	66.4	29.7	80.2	61.9	48.6
Mean	86.3	73.7	51.5	93.6	67.8	74.9
p value	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
$r_1r_2-\mathrm{SOCP}-\mathrm{SVM}$						
Max	90.8	78.7	74.1	98.7	71.6	85.1
Min	84.6	50.0	51.1	80.3	64.4	71.3
Mean	87.7	71.6	64.9	93.5	68.7	80.5
p value	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

performed best in one out of six data sets each. Additionally, both proposed approaches are among the three best methods in all the data sets, demonstrating their effectiveness and robustness. We also observe quite a poor performance for SVM and CS-SVM for imbalanced and overlapped data. For those data sets with higher levels of noise (Abalone_7, Balance, Solar_M and Yeast_ME2, where the best predictive performance is below an AUC of 0.9), it was impossible to find a linear classifier that correctly discriminates between both classes, leading to an AUC close to 0.5 (all instances classified as the majority class). CS-SVM does not represent a real improvement in such cases, requiring the use of resampling techniques in order to achieve a reasonable performance [14]. In contrast, SOCP-based approaches achieved excellent performance without the need of data re-balancing.

5.3. Influence of the parameters and discussion

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

Table 3 summarizes the predictive performance in terms of the AUC for SOCP-SVM, r-SOCP-SVM and r_1r_2-SOCP-SVM (best performance along the different values of C). The average, the minimum, and the maximum performance along different values of η are computed and presented in this table. Additionally, a Student's t test is constructed to assess if the maximum value is significantly higher than the respective mean value. The detailed results of the model selection procedure are presented in Appendix C.

An important influence of parameter η can be observed in Table 3. There is an important gap between the minimum and the maximum AUC in all methods and for each data set, and the maximum value is always significantly higher than the mean, according to the p values associated with the Student's t tests. In this experiment we wanted to test the null hypothesis in which the mean performance is similar to the maximum value.

The optimal η value varies among the different methods, and different data sets, where predictive performance is strongly affected by this parameter. Since no clear rule can be defined in order to obtain this value, it is important to set it using cross-validation considering the values presented in this work (or a broader rank of values).

In order to facilitate studying the influence of hyperparameter C in the solution, Fig. 1 presents the predictive performance in terms of the AUC for standard SVM, CS-SVM, and $r_1r_2-\mathrm{SOCP}-\mathrm{SVM}$ by varying this parameter along the set of different values described earlier.

Fig. 1 shows very stable results for $r_1r_2-\mathrm{SOCP}-\mathrm{SVM}$ along the different values of C in four out of six data sets. Only in the two most overlapped data sets (Balance and Solar_M), the parameter C shows quite a strong influence in the final outcome of the proposed method, and the wrong choice of this parameter may lead to poor performance (AUC below 0.5). The gain using of the proposed model compared to SVM was significant, with the only exception of the Car_34 dataset, where almost perfect classification performance is achieved with all methods. Notice that for Car_34, the results with the suggested approach are much more stable. We still conclude that it is highly recommended to perform an adequate grid search, varying the parameters C along the suggested values in order to obtain desired results for $r_1r_2-\mathrm{SOCP}-\mathrm{SVM}$.

6. Conclusions

In this work, we present a cost-sensitive approach for classifying imbalanced data via direct margin maximization, where this task is

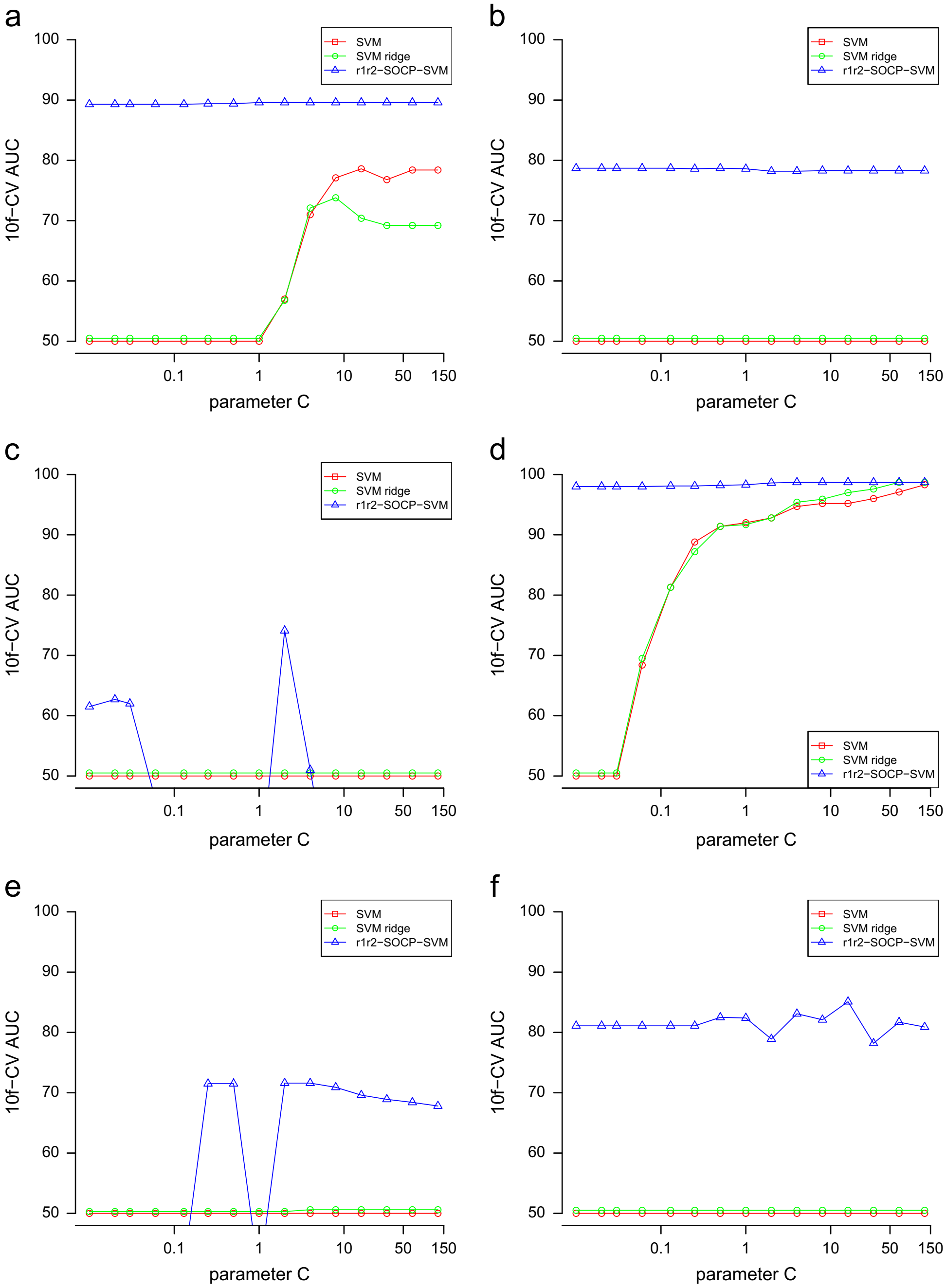

Fig. 1. CV AUC by varying parameter C for standard SVM, CS-SVM and r_1r_2−SOCP−SVM, in all datasets. (a) Ecoli dataset. (b) Abalone_7 dataset. (c) Balance dataset. (d) Car_34 dataset. (e) Solar_M dataset. (f) Yeast_ME2 dataset.

performed separately for each class, and which benefits the correct prediction of the target class.

Empirically, we observed that the proposed approach achieves better results in five out of six benchmark data sets. The gain is particularly important compared to standard and cost-sensitive SVM, since these methods fail at constructing a discriminative function that correctly classifies the target class, leading to high accuracy but an AUC close to 0.5. This phenomenon occurs when the degree of noise in the data set is high, and the best hyperplane under the SVM criterion is the one that predicts all instances as the majority class. These methods performed well only in data sets where the classification accuracy that can be achieved is near 100%.

A comparison with other SOCP-SVM formulations shows the advantages of the proposed method. Even if standard SOCP-SVM achieves significantly better results compared to the standard SVM formulation, and given the structure of the problem, still a gain in terms of predictive performance can be obtained by considering r_1r_2−SOCP − SVM, the proposed method. Additionally, the improvement that this approach presents compared to r-SOCP-SVM demonstrates the importance of classification models that are specially designed for imbalanced data.

The model selection procedure is studied extensively in order to measure and understand the influence of the hyperparameters C and η. We conclude that model selection should be performed carefully, using the suggested values presented in Section 5, in order to obtain the best predictive performance, although results are relatively stable with variations of both parameters.

There are several opportunities for future work. First, the extension of these methods to kernel approaches may lead to better performance, thanks to the ability of constructing non-linear classifiers. Secondly, there is a pressing need for more efficient implementations of second order cone programming formulations. While r-SOCP-SVM represents one step in that direction, faster implementations are necessary for the method to become a real alternative to traditional SVM for large scale datasets. Finally, some approaches have been proposed for feature selection in imbalanced data (see, e.g. Van Hulse et al. [15]). An interesting challenge, however, is to consider advanced feature selection strategies, such as wrapper and embedded methods [19,20], in order to address the interactions between SOCP-SVM and the predictors when facing imbalanced data.

Conflict of interest

None declared.

Acknowledgements

The first author was supported by FONDECYT project 11121196, while the second was funded by FONDECYT project 11110188 and by CONICYT Anillo ACT1106.

Appendix A. Dual formulation for LP-SVM

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

$$\begin{aligned} &\min_{\mathbf{w},r,b,\xi_1,\xi_2} \quad -r + C(\mathbf{e}^\top \cdot \xi_1 + \mathbf{e}^\top \cdot \xi_2) \\ &\text{s.t.} \quad A^\top \mathbf{w} + (b-r)\mathbf{e} + \xi_1 \geq 0, \\ &\quad -B^\top \mathbf{w} - (b+r)\mathbf{e} + \xi_2 \geq 0, \\ &\quad -1 \leq w_j \leq 1, \;\; j = 1,\ldots,n, \\ &\quad \xi_1 \geq 0, \;\; \xi_2 \geq 0, \;\; r \geq 0. \end{aligned} \tag{17}$$

The Lagrangian function associated with the problem (17) is given by

$$\begin{aligned} &L(\mathbf{w},b,r,\xi_i,t,\mathbf{u},\mathbf{v},\mathbf{z}_i,\mathbf{s}_i) = -r + C(\mathbf{e}^\top \cdot \xi_1 + \mathbf{e}^\top \cdot \xi_2) - rt \\ &-\langle A^\top \mathbf{w} + (b-r)\mathbf{e} + \xi_1, \mathbf{z}_1 \rangle \\ &-\langle -B^\top \mathbf{w} - (b+r)\mathbf{e} + \xi_2, \mathbf{z}_2 \rangle - \langle \mathbf{s}_1, \xi_1 \rangle \\ &-\langle \mathbf{s}_2, \xi_2 \rangle - \langle \mathbf{e} - \mathbf{w}, \mathbf{u} \rangle - \langle \mathbf{e} + \mathbf{w}, \mathbf{v} \rangle, \end{aligned} \tag{18}$$

where $t, \mathbf{u}, \mathbf{v}, \mathbf{z}_i, \mathbf{s}_i \geq 0$, $i = 1,2$. Thus, the formulation (17) can be equivalently written as

$$\min_{\mathbf{w},r,b,\xi_i} \max_{t,\mathbf{u},\mathbf{v},\mathbf{z}_i,\mathbf{s}_i} \{ L(\mathbf{w},b,r,\xi_i,t,\mathbf{u},\mathbf{v},\mathbf{z}_i,\mathbf{s}_i) : t,\mathbf{u},\mathbf{v},\mathbf{z}_i,\mathbf{s}_i \geq 0, \;\; i = 1,2 \}. \tag{19}$$

Hence, the dual problem of (17) is given by

$$\max_{t,\mathbf{u},\mathbf{v},\mathbf{z}_i,\mathbf{s}_i} \min_{\mathbf{w},r,b,\xi_i} \{ L(\mathbf{w},b,r,\xi_i,t,\mathbf{u},\mathbf{v},\mathbf{z}_i,\mathbf{s}_i) : t,\mathbf{u},\mathbf{v},\mathbf{z}_i,\mathbf{s}_i \geq 0, \;\; i = 1,2 \}. \tag{20}$$

This expression now enables us to eliminate the primal variables. Taking partial derivatives of L with respect to $\mathbf{w}$, r, b, ξ_1, ξ_2, and using the first order optimality conditions we get

$$\nabla_{\mathbf{w}} L = -A\mathbf{z}_1 + B\mathbf{z}_2 + \mathbf{u} - \mathbf{v} = 0, \tag{21}$$

$$\frac{\partial L}{\partial b} = -\mathbf{e}^\top \cdot \mathbf{z}_1 + \mathbf{e}^\top \cdot \mathbf{z}_2 = 0, \tag{22}$$

$$\frac{\partial L}{\partial r} = -1 - t + \mathbf{e}^\top \cdot \mathbf{z}_1 + \mathbf{e}^\top \cdot \mathbf{z}_2 = 0, \tag{23}$$

$$\nabla_{\xi_1} L = C\mathbf{e} - \mathbf{z}_1 - \mathbf{s}_1 = 0, \tag{24}$$

$$\nabla_{\xi_2} L = C\mathbf{e} - \mathbf{z}_2 - \mathbf{s}_2 = 0, \tag{25}$$

where $\nabla_{\mathbf{w}}$, ∇_{ξ_1}, ∇_{ξ_2} denote the gradient of L with respect to the vectors $\mathbf{w}$, ξ_1 and ξ_2, respectively.

Using (22) and (23) and the constraint $t \geq 0$, one has

$$\mathbf{e}^\top \cdot \mathbf{z}_1 = \mathbf{e}^\top \cdot \mathbf{z}_2 \geq \tfrac{1}{2}. \tag{26}$$

Also, by using (24) and (25) and the constraints $\mathbf{s}_i \geq 0$, for i=1,2, we get

$$0 \leq \mathbf{z}_1 \leq C\mathbf{e}, \quad 0 \leq \mathbf{z}_2 \leq C\mathbf{e}. \tag{27}$$

Then, substituting (21)–(25) in (18) subject to the relevant constraints together with (26) and (27) yields the dual formulation stated as follows:

$$\begin{aligned} &\max_{\mathbf{u},\mathbf{v},\mathbf{z}_1,\mathbf{z}_2} \quad -\mathbf{e}^\top(\mathbf{u}+\mathbf{v}) \\ &\text{s.t.} \quad \mathbf{u} - \mathbf{v} = A\mathbf{z}_1 - B\mathbf{z}_2, \\ &\quad \mathbf{e}^\top \cdot \mathbf{z}_1 = \mathbf{e}^\top \cdot \mathbf{z}_2 \geq \tfrac{1}{2}, \\ &\quad \mathbf{u} \geq 0, \;\; \mathbf{v} \geq 0, \;\; 0 \leq \mathbf{z}_1 \leq C\mathbf{e}, \;\; 0 \leq \mathbf{z}_2 \leq C\mathbf{e}. \end{aligned} \tag{28}$$

Since the relationship

$$\mathbf{z} = \mathbf{u} - \mathbf{v}, \quad \|\mathbf{z}\|_1 = \mathbf{e}^\top \cdot (\mathbf{u}+\mathbf{v}), \quad \mathbf{u},\mathbf{v} \geq 0, \tag{29}$$

holds, the dual problem (28) can be written as

$$\begin{aligned} &\max_{\mathbf{z}_1,\mathbf{z}_2} \quad -\|A\mathbf{z}_1 - B\mathbf{z}_2\|_1 \\ &\text{s.t.} \quad \mathbf{e}^\top \cdot \mathbf{z}_1 = \mathbf{e}^\top \cdot \mathbf{z}_2 \geq \tfrac{1}{2}, \\ &\quad 0 \leq \mathbf{z}_1 \leq C\mathbf{e}, \quad 0 \leq \mathbf{z}_2 \leq C\mathbf{e}. \end{aligned} \tag{30}$$

Remark 1. Without loss of generality, we can suppose that $\mathbf{e}^\top \cdot \mathbf{z}_1 = \mathbf{e}^\top \cdot \mathbf{z}_2 = 1$, thus (30) can be cast as

$$\begin{aligned} &\max_{\mathbf{z}_1,\mathbf{z}_2} \quad -\|A\mathbf{z}_1 - B\mathbf{z}_2\|_1 \\ &\text{s.t.} \quad \mathbf{e}^\top \cdot \mathbf{z}_1 = 1, \;\; \mathbf{e}^\top \cdot \mathbf{z}_2 = 1, \\ &\quad 0 \leq \mathbf{z}_1 \leq C\mathbf{e}, \;\; 0 \leq \mathbf{z}_2 \leq C\mathbf{e}. \end{aligned} \tag{31}$$

Appendix B. Dual formulation for $r_1 r_2$-SOCP-SVM

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

$$\begin{aligned} &L(\mathbf{w},b,r_i,t_i,\lambda_i,\mathbf{z}_i) = -r_1 - Cr_2 - t_1 r_1 - t_2 r_2 \\ &-\lambda_1(\mathbf{w}^\top \cdot \boldsymbol{\mu}_1 - b - r_1 - \kappa_1 \|S_1^\top \mathbf{w}\|) \\ &-\lambda_2(b - \mathbf{w}^\top \cdot \boldsymbol{\mu}_2 - r_2 - \kappa_2 \|S_2^\top \mathbf{w}\|) \\ &-\langle \mathbf{e} - \mathbf{w}), \mathbf{z}_1 \rangle - \langle \mathbf{e} + \mathbf{w}), \mathbf{z}_2 \rangle, \end{aligned} \tag{32}$$

where $\lambda_1, \lambda_2, t_1, t_2, \mathbf{z}_1, \mathbf{z}_2 \geq 0$. Since the relationship $\|\mathbf{v}\| = \sup \|\mathbf{u}\| \leq 1 \mathbf{u}^\top \cdot \mathbf{v}$ holds for any $\mathbf{v} \in \Re^n$, we can modify the Lagrangian as follows:

$$\begin{aligned} &L_1(\mathbf{w},b,r_i,t_i,\lambda_i,\mathbf{z}_i,\mathbf{u}_i) = -r_1 - Cr_2 - t_1 r_1 - t_2 r_2 \\ &-\lambda_1(\mathbf{w}^\top \cdot \boldsymbol{\mu}_1 - b - r_1 - \kappa_1 \mathbf{u}_1^\top \cdot S_1^\top \mathbf{w}) \\ &-\lambda_2(b - \mathbf{w}^\top \cdot \boldsymbol{\mu}_2 - r_2 - \kappa_2 \mathbf{u}_2^\top \cdot S_2^\top \mathbf{w}) \\ &-\langle \mathbf{e} - \mathbf{w}), \mathbf{z}_1 \rangle - \langle \mathbf{e} + \mathbf{w}), \mathbf{z}_2 \rangle, \end{aligned} \tag{33}$$

where $\lambda_1, \lambda_2, t_1, t_2, \mathbf{z}_1, \mathbf{z}_2 \geq 0$ and $\|\mathbf{u}_i\| \leq 1$, $i = 1,2$. Then

$$L(\mathbf{w},b,r_i,t_i,\lambda_i,\mathbf{z}_i) = \max_{\mathbf{u}_i} \{ L_1(\mathbf{w},b,r_i,t_i,\lambda_i,\mathbf{z}_i,\mathbf{u}_i) : \|\mathbf{u}_i\| \leq 1, \;\; i = 1,2 \}. \tag{34}$$

Thus, Problem (14) can be equivalently written as

$$\min_{\mathbf{w},b,r_i} \max_{t_i,\lambda_i,\mathbf{z}_i,\mathbf{u}_i} \{ L_1(\mathbf{w},b,r_i,t_i,\lambda_i,\mathbf{z}_i,\mathbf{u}_i) : \|\mathbf{u}_i\| \leq 1, \lambda_i, t_i, \mathbf{z}_i \geq 0, \;\; i = 1,2 \}. \tag{35}$$

Hence, the dual problem of (14) is given by

$$\max_{t_i,\lambda_i,\mathbf{z}_i,\mathbf{u}_i\mathbf{w},b,r_i} \min\{L_1(\mathbf{w},b,r_i,t_i,\lambda_i,\mathbf{z}_i,\mathbf{u}_i) : \|\mathbf{u}_i\| \le 1, \lambda_i, t_i, \mathbf{z}_i \ge 0,\ i=1,2\}. \tag{36}$$

The expression (36) now enables us to eliminate the primal variables to give the dual formulation. Taking partial derivatives of L_1 with respect to $\mathbf{w}$, b, r_1 and r_2 yields

$$\nabla_{\mathbf{w}} L_1 = -\lambda_1\boldsymbol{\mu}_1 + \lambda_1\kappa_1 S_1\mathbf{u}_1 + \lambda_2\boldsymbol{\mu}_2 + \lambda_2\kappa_2 S_2\mathbf{u}_2 + \mathbf{z}_1 - \mathbf{z}_2, \tag{37}$$

$$\frac{\partial L_1}{\partial b} = \lambda_1 - \lambda_2, \tag{38}$$

$$\frac{\partial L_1}{\partial r_1} = -1 - t_1 + \lambda_1, \tag{39}$$

$$\frac{\partial L_1}{\partial r_2} = -C - t_2 + \lambda_2. \tag{40}$$

The Karush–Kuhn–Tucker (KKT) conditions with Eqs. (38)–(40) imply that

$$\lambda_1 = \lambda_2 = \lambda, \quad t_1 = \lambda - 1, \quad t_2 = \lambda - C. \tag{41}$$

and with Eq. (37) imply that

$$\mathbf{z}_1 - \mathbf{z}_2 = \lambda(\boldsymbol{\mu}_1 - \kappa_1 S_1\mathbf{u}_1 - \boldsymbol{\mu}_2 - \kappa_2 S_2\mathbf{u}_2). \tag{42}$$

From (41), (42) and (33) the dual problem can be stated as follows:

$$\begin{aligned} \max_{\mathbf{z}_1,\mathbf{z}_2,\mathbf{u}_1,\mathbf{u}_2,\lambda} \quad & -\mathbf{e}^\top \cdot (\mathbf{z}_1 + \mathbf{z}_2) \\ \text{s.t.} \quad & \mathbf{z}_1 - \mathbf{z}_2 = \lambda(\tilde{\mathbf{z}}_1 - \tilde{\mathbf{z}}_2), \\ & \tilde{\mathbf{z}}_i = \boldsymbol{\mu}_i + (-1)^i \kappa_i S_i \mathbf{u}_i, \quad \|\mathbf{u}_i\| \le 1, \ i = 1,2, \\ & \lambda \ge \max\{1, C\}, \ \mathbf{z}_1 \ge 0, \ \mathbf{z}_2 \ge 0. \end{aligned} \tag{43}$$

Remark 2. It follows from the relation (41) that the Lagrange multipliers associated with the conic constraints of the linear SOC problem (14) are always different from zero.

Again, since the relationship (29) holds, the dual problem (43) can be written as

$$\begin{aligned} \max_{\tilde{\mathbf{z}}_1,\tilde{\mathbf{z}}_2,\lambda} \quad & -\lambda \|\tilde{\mathbf{z}}_1 - \tilde{\mathbf{z}}_2\|_1 \\ \text{s.t.} \quad & \tilde{\mathbf{z}}_i \in \mathbf{B}_i(\boldsymbol{\mu}_i, S_i, \kappa_i), \quad i = 1,2, \\ & \lambda \ge \max\{1, C\}, \end{aligned} \tag{44}$$

Table C1
Mean accuracy and AUC, in percentages, for all data sets varying parameter η. SOCP-SVM Model.

η value	Ecoli		Abalone_7		Balance		Car_34		Solar_M		Yeast_ME2	
	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC
η=(0.2,0.2)	82.7	88.2	72.4	75.1	52.6	55.1	90.1	94.6	74.2	70.3	85.9	**85.0**
η=(0.2,0.4)	86.3	89.1	76.9	76.1	52.7	**71.6**	93.7	96.6	79.4	69.5	88.1	82.2
η=(0.2,0.6)	89.3	87.8	75.4	73.9	51.1	35.1	97.6	97.6	84.7	66.0	90.0	81.3
η=(0.2,0.8)	92.5	85.5	80.6	67.4	7.8	50.0	96.8	79.9	84.1	65.8	74.7	57.3
η=(0.4,0.2)	81.8	87.7	70.2	76.2	72.0	68.3	90.3	94.8	68.4	70.1	82.7	83.3
η=(0.4,0.4)	85.1	88.4	75.0	76.9	55.7	33.9	93.3	96.4	74.2	70.1	85.5	81.9
η=(0.4,0.6)	89.3	87.8	77.9	75.5	42.7	36.1	96.8	**98.3**	78.0	70.2	87.9	81.2
η=(0.4,0.8)	91.7	82.4	79.9	68.2	7.8	50.0	97.3	84.2	90.2	65.4	79.3	71.9
η=(0.6,0.2)	81.2	87.3	67.3	**78.4**	82.4	44.7	90.7	95.0	63.0	71.0	77.2	80.5
η=(0.6,0.4)	84.5	**89.2**	71.3	77.6	75.2	42.6	92.1	95.7	70.1	69.5	81.0	81.5
η=(0.6,0.6)	88.7	87.4	73.4	76.5	72.9	46.2	94.9	97.2	66.5	**73.4**	69.2	62.5
η=(0.6,0.8)	89.6	82.4	79.5	72.2	9.9	39.7	98.0	89.4	88.9	62.0	90.2	83.3
η=(0.8,0.2)	78.6	85.8	62.3	77.0	90.6	49.1	91.1	95.2	62.3	60.8	74.6	83.0
η=(0.8,0.4)	81.2	87.3	67.1	78.3	89.0	48.3	91.1	95.2	56.4	71.0	51.4	64.2
η=(0.8,0.6)	85.1	88.4	75.1	76.4	85.0	49.8	91.1	95.2	68.0	66.2	72.7	82.0
η=(0.8,0.8)	80.9	87.2	75.9	74.4	58.6	67.5	98.2	91.2	55.5	61.4	88.4	81.5

Table C2
Mean accuracy and AUC, in percentages, for all data sets varying parameter η. r-SOCP-SVM Model.

η value	Ecoli		Abalone_7		Balance		Car_34		Solar_M		Yeast_ME2	
	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC
η=(0.2,0.2)	82.1	87.8	71.3	72.8	53.6	67.5	87.4	92.5	74.2	71.8	83.6	82.8
η=(0.2,0.4)	84.8	87.8	76.7	76.1	64.8	71.8	92.1	94.4	79.7	69.0	87.3	80.9
η=(0.2,0.6)	86.9	85.3	76.9	73.0	68.1	67.2	96.9	95.6	84.9	67.4	89.8	81.2
η=(0.2,0.8)	91.7	85.4	82.0	67.3	72.8	41.3	96.8	80.2	87.7	66.9	82.8	68.9
η=(0.4,0.2)	82.4	88.0	71.1	75.9	48.2	60.9	87.8	93.4	69.5	**72.0**	81.7	80.9
η=(0.4,0.4)	83.3	87.0	74.8	75.8	56.5	72.7	94.6	97.1	74.8	69.2	85.5	81.9
η=(0.4,0.6)	87.5	86.8	80.2	74.7	61.6	**74.6**	97.3	**98.5**	79.1	68.7	84.2	79.3
η=(0.4,0.8)	90.5	80.3	82.6	67.3	58.6	34.5	97.0	83.0	88.0	67.0	69.1	62.0
η=(0.6,0.2)	81.2	87.3	66.7	**77.9**	50.1	50.3	91.1	95.2	63.3	68.7	77.6	80.7
η=(0.6,0.4)	82.4	86.5	72.0	76.3	55.8	72.4	91.7	95.5	67.5	71.9	69.6	71.7
η=(0.6,0.6)	87.5	86.8	78.2	75.9	54.4	57.4	94.8	97.2	74.5	67.6	85.4	**84.7**
η=(0.6,0.8)	89.0	82.0	82.4	66.4	50.6	31.1	96.8	91.4	86.7	64.2	59.0	48.6
η=(0.8,0.2)	79.5	86.3	61.9	76.7	52.3	30.2	91.1	95.2	63.2	64.7	67.8	80.4
η=(0.8,0.4)	81.5	86.0	69.0	77.1	50.1	31.7	91.1	95.2	82.5	61.9	78.6	80.2
η=(0.8,0.6)	86.0	**90.0**	76.2	76.1	43.7	30.1	91.1	95.2	69.0	69.2	51.9	66.4
η=(0.8,0.8)	88.1	88.2	76.1	69.4	46.4	29.7	98.5	97.5	83.4	64.6	54.8	67.9

Table C3
Mean accuracy and AUC, in percentages, for all data sets varying parameter η. r_1r_2-SOCP-SVM Model.

η value	Ecoli		Abalone_7		Balance		Car_34		Solar_M		Yeast_ME2	
	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC
$\eta=(0.2,0.2)$	87.8	88.4	73.6	**78.7**	60.5	51.1	90.3	94.8	74.9	69.8	85.9	82.6
$\eta=(0.2,0.4)$	89.9	89.6	77.0	76.6	71.0	70.1	94.7	97.1	79.9	67.8	88.4	80.5
$\eta=(0.2,0.6)$	91.1	87.5	82.9	71.5	75.6	68.6	97.9	98.5	85.5	66.7	90.2	81.4
$\eta=(0.2,0.8)$	92.8	85.9	90.5	50.0	92.2	61.9	96.8	80.3	91.1	65.3	91.4	**85.1**
$\eta=(0.4,0.2)$	86.0	88.8	70.4	78.6	60.4	59.5	90.3	94.7	68.8	69.9	82.8	82.7
$\eta=(0.4,0.4)$	89.0	88.6	75.2	78.0	70.1	72.7	93.6	96.5	75.8	70.3	86.0	82.6
$\eta=(0.4,0.6)$	90.5	88.4	90.1	76.5	83.4	**74.1**	97.7	**98.7**	89.9	68.2	88.5	80.6
$\eta=(0.4,0.8)$	92.2	86.8	90.6	54.9	91.7	62.3	97.2	83.9	92.7	67.9	94.8	71.5
$\eta=(0.6,0.2)$	83.0	87.5	67.4	78.3	88.4	65.8	91.1	95.2	65.1	69.6	76.9	81.3
$\eta=(0.6,0.4)$	86.9	89.1	68.8	77.4	90.3	72.6	92.4	95.9	85.8	69.4	83.1	82.7
$\eta=(0.6,0.6)$	89.0	87.6	82.0	76.3	81.6	71.3	96.0	97.8	79.3	69.7	87.0	83.5
$\eta=(0.6,0.8)$	89.3	**90.8**	90.6	56.9	91.2	62.7	97.6	86.4	92.3	64.4	92.7	81.7
$\eta=(0.8,0.2)$	79.2	86.2	62.8	77.0	69.4	54.8	91.1	95.2	68.1	69.1	74.7	80.2
$\eta=(0.8,0.4)$	83.0	87.7	88.3	74.9	73.3	64.2	91.1	95.2	78.5	71.3	94.0	71.3
$\eta=(0.8,0.6)$	87.2	86.6	88.1	75.8	85.4	58.7	92.2	95.8	67.1	**71.6**	84.4	80.7
$\eta=(0.8,0.8)$	89.3	84.6	90.6	64.3	92.0	67.6	97.9	89.4	95.1	69.1	94.5	79.3

where

$$\mathbf{B}_i(\mu_i, S_i, \kappa_i) = \{\tilde{\mathbf{z}}_i : \tilde{\mathbf{z}}_i = \boldsymbol{\mu}_i + (-1)^i S_i \mathbf{u}_i,\ \|\mathbf{u}_i\| \le 1\},\quad i = 1, 2.$$

Note that the objective function is maximized when $\lambda = \hat{\lambda} = \max\{1, C\}$. Then, the dual problem (44) can be stated as follows:

$$\begin{aligned} \max_{\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2} \quad & -\hat{\lambda}\, \|\tilde{\mathbf{z}}_1 - \tilde{\mathbf{z}}_2\|_1 \\ \text{s.t.} \quad & \tilde{\mathbf{z}}_i \in \mathbf{B}_i(\boldsymbol{\mu}_i, S_i, \kappa_i), \quad i = 1, 2. \end{aligned} \tag{45}$$

Appendix C. Model selection performance along different values of η

Tables C1–C3 present the predictive performance in terms of accuracy and AUC for SOCP-SVM, r-SOCP-SVM, and r_1r_2-SOCP-SVM (best performance along the different values of C) respectively, considering all combinations for $\eta = (\eta_1, \eta_2)$.

References

[1] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, Keel: A software tool to assess evolutionary algorithms to data mining problems, Soft Comput. 13 (3) (2009) 307–318.

[2] F. Alizadeh, D. Goldfarb, Second-order cone programming, Math. Progr. 95 (2003) 3–51.

[3] F. Alvarez, J. López, H.C. Ramírez, 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.

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

[5] F. R Bach, D. Heckerman, E. Horvitz, Considering cost asymmetry in learning classifiers, J. Mach. Learn. Res. 7 (2006) 1713–1741.

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

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

[8] N.V. Chawla, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, Smote: Synthetic minority oversampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[9] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explor. 6 (2004) 1–6.

[10] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[11] Z. Ding, Diversified ensemble classifiers for highly imbalanced data learning and their application in bioinformatics (Ph.D. thesis), Computer Science Dissertations, Paper 60, 2011.

[12] T. Fawcett, F. Provost, Adaptive fraud detection, Data Min. Knowl. Discov. 1 (1997) 291–316.

[13] F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutiérrez, A dynamic over-sampling procedure based on sensitivity for multi-class problems, Pattern Recognit. 44 (2011) 1821–1833.

[14] H. He, E. García, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (2009) 1263–1284.

[15] J. Van Hulse, T.M. Khoshgoftaar, A. Napolitano, R. Wald, Feature selection with high-dimensional imbalanced data, in: Proceedings of the IEEE International Conference on Data Mining Workshops, 2009, pp. 507–514.

[16] T.M. Khoshgoftaar, J. Van Hulse, A. Napolitano, An exploration of learning when data is noisy and imbalanced, Intell. Data Anal. 15 (2011) 215–236.

[17] D.A. Kumar, V. Ravi, Predicting credit card customer churn in banks using data mining, Int. J. Data Anal. Tech. Strateg. 1 (2008) 4–28.

[18] G. Lanckriet, L.E. Ghaoui, C. Bhattacharyya, M. Jordan, A robust minimax approach to classification, J. Mach. Learn. Res. 3 (2003) 555–582.

[19] S. Maldonado, R. Weber, A wrapper method for feature selection using support vector machines, Inf. Sci. 179 (2009) 2208–2217.

[20] S. Maldonado, R. Weber, J. Basak, Kernel-penalized SVM for feature selection, Inf. Sci. 181 (1) (2011) 115–128.

[21] O.L. Mangasarian, A finite newton method for classification, Optim. Methods Softw. 17 (5) (2002) 913–929.

[22] 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.

[23] X. Peng, Building sparse twin support vector machine classifiers in primal space, Inf. Sci. 181 (18) (2011) 3967–3980.

[24] K. Pichara, A. Soto, Active learning and subspace clustering for anomaly detection, Intell. Data Anal. 15 (2011) 151–171.

[25] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: Advances in Kernel Methods-Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 185–208.

[26] B. Raskutti, A. Kowalczyk, Extreme rebalancing for SVMS: a case study, ACM SIGKDD Explor. Newslett. 6 (2004) 60–69. (Special Issue on Learning from Imbalanced Datasets).

[27] K.B. Schebesch, R. Stecking, Data analysis, machine learning and applications, in: Using Multiple SVM Models for Unbalanced Credit Scoring Data Sets, Springer, Berlin, Heidelberg, 2008, pp. 515–522.

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

[29] 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.

[30] M. Sokolova, N. Japkowicz, S. Szpakowicz, Beyond accuracy, f-score and ROC: a family of discriminant measures for performance evaluation, in: Advances in Artificial Intelligence, Springer, Berlin, Heidelberg, 2006, pp. 1015–1021.

[31] J.F. Sturm, Using sedumi 10.2, a matlab toolbox for optimization over symmetric cones, Optim. Methods Softw. 11 (12) (1999) 625–653. (Special Issue on Interior Point Methods (CD supplement with software)).

[32] Y.M. Sun, M.S. Kamel, A.K.C. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognit. 40 (2007) 3358–3378.

[33] Y. Tang, S. Krasser, D. Alperovitch, P. Judge, Spam sender detection with classification modeling on highly imbalanced mail server behavior data, in: Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition, 2008, pp. 174–180.

[34] D.M.J. Tax, R. Duin, Support vector data description, Mach. Learn. 54 (2004) 45–66.
[35] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[36] J. Wang, J. You, Q. Li, Y. Xu, Extract minimum positive and maximum negative features for imbalanced binary classification, Pattern Recognit. 45 (2012) 1136–1145.
[37] G.M. Weiss, Mining with rarity: a unifying framework, ACM SIGKDD Explor. Newslett. 6 (2004) 7–19.
[38] J. Weston, A. Elisseeff, G. BakIr, F. Sinz, The Spider Machine Learning Toolbox, 2005.
[39] Z. Zheng, X. Wu, R. Srihari, Feature selection for text categorization on imbalanced data, SIGKDD Explor. 6 (2004) 80–89.
[40] W. Zhou, L. Zhang, L. Jiao, Linear programming support vector machines, Pattern Recognit. 35 (2002) 2927–2936.

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.

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