Knowledge-Based Systems 177 (2019) 127–135

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Regularized minimax probability machine[☆]

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

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

HIGHLIGHTS

- Novel robust approach for classification using nonlinear second-order cone programming.
- The regularized version for Minimax Probability Machine is proposed.
- A geometrically grounded method based on the concept of ellipsoids.
- Superior performance is achieved in experiments on benchmark datasets.

ARTICLE INFO

Article history:
Received 18 October 2018
Received in revised form 2 January 2019
Accepted 19 April 2019
Available online 25 April 2019

Keywords:
Minimax probability machine
Regularization
Second-order cone programming
Support vector machines

ABSTRACT

In this paper, we propose novel second-order cone programming formulations for binary classification, by extending the Minimax Probability Machine (MPM) approach. Inspired by Support Vector Machines, a regularization term is included in the MPM and Minimum Error Minimax Probability Machine (MEMPM) methods. This inclusion reduces the risk of obtaining ill-posed estimators, stabilizing the problem, and, therefore, improving the generalization performance. Our approaches are first derived as linear methods, and subsequently extended as kernel-based strategies for nonlinear classification. Experiments on well-known binary classification datasets demonstrate the virtues of the regularized formulations in terms of predictive performance.

1. Introduction

The Minimax Probability Machine (MPM) [1] is a well-known machine learning method that minimizes the worst-case (maximal) probability of a test example being misclassified. This model assumes that each of the two training patterns is generated by random variables with a known mean and covariance matrix. Therefore, it provides a *robust setting* for machine learning. Robustness is an important virtue since it guarantees that predictive performance does not deteriorate much with changing environments [2–4].

Robust machine learning methods in the area of MPM are usually formulated as Second-Order Cone Programming (SOCP) problems [5]. Robustness is usually conferred via chance constraints which provide bounds for each class accuracy (also referred to as class recall), for even the worst possible data distribution. Using an appropriate application of the Chebyshev inequality, this problem can be written equivalently as a linear SOCP problem [1,3,4]. Although linear SOCP models can be solved efficiently via interior point algorithms [6], research on nonlinear SOCP optimizers is much more recent [3], limiting the opportunities for novel machine learning formulations.

One disadvantage of MPM is that it does not include a regularization term for the construction of the separating hyperplane. Regularization is used for avoiding ill-posed problems, but also for fitting the training samples well, while reducing the risk of overfitting at the same time [7]. For example, the Tikhonov regularization, or l_2-norm, has been used widely in learning machines; a strategy that was popularized by the Support Vector Machine (SVM) method [7]. This strategy is still among the most popular classification methods, and has been widely used in domains such as computer vision [8,9], medical diagnosis [10,11], and business analytics [12,13]. Saketha Nath and Bhattacharyya [4] proposed a regularized alternative for MPM, in which the l_2-norm was minimized. However, this approach uses fixed values for the misclassification rates, rather than optimizing these measures with the optimization problem.

Another disadvantage of MPM is that it assumes a unique class recall for both classes. In other words, it assumes that both classes are equally important. The Minimum Error Minimax Probability Machine (MEMPM) [14] is a relevant MPM extension

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.04.016.

[*] Corresponding author.

E-mail addresses: smaldonado@uandes.cl (S. Maldonado), micarrasco@uandes.cl (M. Carrasco), julio.lopez@udp.cl (J. López).

https://doi.org/10.1016/j.knosys.2019.04.016
0950-7051/© 2019 Elsevier B.V. All rights reserved.

which minimizes a convex combination of the misclassification rates, balancing the two classes based on their prior probabilities.

In this paper, we propose regularized MPM and MEMPM formulations, in which the l_2-norm of the weight vector is minimized jointly with the error rates. Unlike the model proposed by Saketha Nath and Bhattacharyya [4], these rates are a part of the optimization problem, leading to nonlinear Second-Order Cone Programming (NSOCP) formulations.

The remainder of this paper is organized as follows: in Section 2, we present the robust formulations that are relevant for our proposal: MPM, MEMPM, and the maximum margin approach proposed by Saketha Nath and Bhattacharyya [4]. The proposed regularized methods are detailed in Section 3. The results of the numerical experiments on benchmark datasets are reviewed in Section 4. Finally, the main conclusions of this work are provided in Section 5, where future developments are also proposed.

2. Prior work on robust binary classification using SOCP

In this section, we briefly introduce the foundations for differentiable nonlinear second-order cone programming. Subsequently, the models that are relevant for our proposal are formalized, namely the MPM [1], the MEMPM [14], and the robust regularized approach by Saketha Nath and Bhattacharyya [4]. This section concludes with a discussion of current trends and recent studies on robust classification with SOCP.

2.1. Preliminaries on nonlinear second-order cone programming

Let $g\colon \Re^n \to \Re^m$ be a function defined by $g(x) = (g_1(x), \bar{g}(x))$, where $g_1\colon \Re^n \to \Re$, $\bar{g}\colon \Re^n \to \Re^{m-1}$, and $m \geq 2$. A second-order cone (SOC) constraint has the form $g_1(x) \geq \|\bar{g}(x)\|$, where $\|\cdot\|$ denotes the Euclidean norm. In the case in which $m = 1$, the constraint is defined simply by $g(x) \geq 0$. We note that this is equivalent to having $g(x) \in \mathcal{K}^m$, where $\mathcal{K}^m = \{(y_1, \bar{y}) \in \Re\times\Re^{m-1} : y_1 \geq \|\bar{y}\|\}$, for $m \geq 2$, and $\mathcal{K}^1 = \Re_+$.

With this notation, a nonlinear Second-Order Cone Programming (NSOCP) problem is defined as:

$$\min_{x\in\Re^n}\{f(x);\ g^j(x) \in \mathcal{K}^{m_j},\ j = 1,\ldots,J\}, \tag{1}$$

where $f\colon \Re^n \to \Re$ and $g^j\colon \Re^n \to \Re^{m_j}$ $(j = 1,\ldots,J)$ are continuously differentiable functions. In the particular case where the function f is linear and g^j are affine, Eq. (1) becomes a Linear Second-Order Cone Programming (LSOCP) problem.

There are several alternatives for solving an NSOCP problem numerically; see e.g. [15–17]. Recently, we have proposed an interior point algorithm for solving NSOCP problems, which achieves positive results on medium-sized problems [18]. This approach, called FDIPA$_{soc}$, is formalized in Section 3.3.

2.2. The minimax probability machine

Let $\mathbf{X}_1$ an $\mathbf{X}_2$ be n-dimensional random vectors that generate the two classes of a binary classification problem, where their respective mean and covariance matrices are given by $(\boldsymbol{\mu}_i, \Sigma_i)$, with $\boldsymbol{\mu}_i \in \Re^n$ and $\Sigma_i \in \mathcal{S}^n_+$, for $i = 1, 2$, where $\mathcal{S}^n_+$ denotes the set of symmetric positive definite matrices. Let us denote the family of distributions which have a common mean and covariance by $\mathbf{X} \sim (\boldsymbol{\mu}, \Sigma)$.

The main goal of the MPM method is to determine a hyperplane of the form $\mathbf{w}^\top\mathbf{x} + b = 0$, with $\mathbf{w} \in \Re^n \setminus \{\mathbf{0}\}$ and $b \in \Re$, such that it separates the two classes with maximal probability with respect to all distributions [1]. This formulation is given by

$$\begin{aligned}\max_{\mathbf{w},b,\alpha}\quad & \alpha\\ \text{s.t.}\quad & \inf_{\mathbf{X}_1\sim(\boldsymbol{\mu}_1,\Sigma_1)} \Pr\{\mathbf{w}^\top\mathbf{X}_1 + b \geq 0\} \geq \alpha,\\ & \inf_{\mathbf{X}_2\sim(\boldsymbol{\mu}_2,\Sigma_2)} \Pr\{\mathbf{w}^\top\mathbf{X}_2 + b \leq 0\} \geq \alpha,\end{aligned} \tag{2}$$

where $\alpha \in (0, 1)$ represents the lower bound for each class recall, or, in other words, the worst-case accuracy.

A robust formulation can be obtained by using Theorem 2.1 (see [1, Lemma 1] for details), which is presented next:

Theorem 2.1 (*Multivariate Chebyshev Inequality*)**.** *Let* $\mathbf{x}$ *be a* n*-dimensional random variable with mean and covariance* $(\boldsymbol{\mu}, \Sigma)$*, where* Σ *is a positive semidefinite symmetric matrix. Given* $\mathbf{a} \in \Re^n \setminus \{\mathbf{0}\}$, $b \in \Re$, *such that* $\mathbf{a}^\top\boldsymbol{\mu} + b \geq 0$, *and* $\alpha \in (0, 1)$, *the condition*

$$\inf_{\mathbf{x}\sim(\boldsymbol{\mu},\Sigma)} Pr\{\mathbf{a}^\top\mathbf{x} + b \geq 0\} \geq \alpha$$

holds if and only if $\mathbf{a}^\top\boldsymbol{\mu} + b \geq \kappa(\alpha)\sqrt{\mathbf{a}^\top\Sigma\mathbf{a}}$, *where* $\kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$.

Using the Chebyshev inequality presented in Theorem 2.1, the optimization problem (2) is equivalent to

$$\begin{aligned}\max_{\mathbf{w},b,\alpha}\quad & \alpha\\ \text{s.t.}\quad & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq \kappa(\alpha)\sqrt{\mathbf{w}^\top\Sigma_1\mathbf{w}},\\ & -(\mathbf{w}^\top\boldsymbol{\mu}_2 + b) \geq \kappa(\alpha)\sqrt{\mathbf{w}^\top\Sigma_2\mathbf{w}}.\end{aligned} \tag{3}$$

After some algebraic manipulations, we see that Formulation (3) can be written equivalently as (see [1, Theorem 2], for details):

$$\begin{aligned}\min_{\mathbf{w}}\ & \sqrt{\mathbf{w}^\top\Sigma_1\mathbf{w}} + \sqrt{\mathbf{w}^\top\Sigma_2\mathbf{w}}\\ \text{s.t. } & \mathbf{w}^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = 1.\end{aligned} \tag{4}$$

This problem can be reduced to a linear SOCP problem, which can be solved efficiently via interior point algorithms [5,19].

Remark 1. In practice, the mean and the covariance matrix are usually not available. Therefore, their respective empirical estimations are used instead.

Remark 2. A kernel-based version of the MPM model can be obtained. This version can be found in Lanckriet et al. [1, Theorem 6].

2.3. Minimum error minimax probability machine

The Minimum Error Minimax Probability Machine (MEMPM) [14] extends the MPM method by considering two different worst-case accuracies, one for each class, instead of a single variable α. Let $\theta \in (0, 1)$ be the prior probability of class $\mathbf{X_1}$, and, consequently, $1 - \theta$ is the prior probability of class $\mathbf{X_2}$. The MEMPM method is given by the following formulation:

$$\begin{aligned}\max_{\mathbf{w},b,\alpha_1,\alpha_2}\quad & \theta\alpha_1 + (1-\theta)\alpha_2\\ \text{s.t.}\quad & \inf_{\mathbf{X}_1\sim(\boldsymbol{\mu}_1,\Sigma_1)} \Pr\{\mathbf{w}^\top\mathbf{X}_1 + b \geq 0\} \geq \alpha_1,\\ & \inf_{\mathbf{X}_2\sim(\boldsymbol{\mu}_2,\Sigma_2)} \Pr\{\mathbf{w}^\top\mathbf{X}_2 + b \leq 0\} \geq \alpha_2.\end{aligned} \tag{5}$$

Following the reasoning behind MPM, the following robust model can be obtained by using the Chebyshev inequality:

$$\begin{aligned}\max_{\mathbf{w},b,\alpha_1,\alpha_2}\quad & \theta\alpha_1 + (1-\theta)\alpha_2\\ \text{s.t.}\quad & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq \kappa(\alpha_1)\sqrt{\mathbf{w}^\top\Sigma_1\mathbf{w}},\\ & -(\mathbf{w}^\top\boldsymbol{\mu}_2 + b) \geq \kappa(\alpha_2)\sqrt{\mathbf{w}^\top\Sigma_2\mathbf{w}},\end{aligned} \tag{6}$$

where $\kappa(\alpha_k) = \sqrt{\frac{\alpha_k}{1-\alpha_k}}$, for $k = 1, 2$.

In order to solve Problem (6), the authors first set one of the accuracies as fixed, optimizing only one of them. Assuming that

α_1 is fixed, Formulation (6) becomes

$$\begin{aligned} \max_{\alpha_2, \mathbf{w} \neq 0} \quad & \theta \frac{\gamma^2}{\gamma^2+1} + (1-\theta)\alpha_2 \\ \text{s.t.} \quad & \mathbf{w}^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = 1, \end{aligned} \tag{7}$$

where

$$\gamma = \gamma(\mathbf{w}) = \frac{1 - k(\alpha_2)\sqrt{\mathbf{w}^\top \Sigma_2 \mathbf{w}}}{\sqrt{\mathbf{w}^\top \Sigma_1 \mathbf{w}}}. \tag{8}$$

Remark 3. Note that, if α_2 is fixed, Formulation (7) is equivalent to solving the problem

$$\max_{\mathbf{w} \neq 0} \{\gamma(\mathbf{w}) : \mathbf{w}^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = 1\},$$

which has the remarkable property that every local optimum is a global one.

Based on this fact, the authors propose an iterative procedure for solving Formulation (7): fix α_2 and maximize γ, then update α_2 following the Quadratic Interpolation method (see Bertsekas [20] for details), and repeat until convergence.

After obtaining α_2 and $\mathbf{w}$ using the Quadratic Interpolation procedure, the variable α_1 is given by $\alpha_1 = \frac{[\gamma(\mathbf{w})]^2}{[\gamma(\mathbf{w})]^2+1}$, with $\gamma(\mathbf{w})$ obtained by Eq. (8).

Remark 4. In order to obtain a nonlinear classifier, the kernel-based version of the MEMPM problem can be derived. The formulation can be found in [14, Section 4].

Other MPM and MEMPM extensions include the Biased Minimax Probability machine (BMPM) [21], which aims at biasing the model towards one class by fixing one class recall while optimizing the other one; or the Structural Minimax Probability Machine (SMPM), which uses two mixture models to capture the structural information related to each training pattern, instead of simply considering the prior probability distribution for each category [22].

2.4. Maximum-margin classifiers with specified error rates

A Robust Maximum-Margin Classifier (RMMC) was proposed by Saketha Nath and Bhattacharyya [4], which extends the reasoning behind MEMPM and includes a regularization term known as the ℓ_2-norm. This model minimizes the Euclidean norm of $\mathbf{w}$, which is equivalent to maximizing the separation margin between the two training patterns. This strategy is inspired by the Support Vector Machine (SVM) method [7], which maximizes the margin between the two class patterns represented as convex hulls.

The RMMC constructs a classifier in such a way that the probability of correct classification for each class k should be higher than $\alpha_k \in (0, 1)$, $k = 1, 2$, even for the worst possible data distribution. Unlike MPM or MEMPM, α_k are pre-specified parameters that have to be defined arbitrarily, or tuned using a validation strategy. The RMMC formulation follows:

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

where $\Sigma_k = S_k S_k^\top$, and $\kappa_k = \sqrt{\frac{\alpha_k}{1-\alpha_k}}$ for $k = 1, 2$. Formulation (9) has two linear SOC constraints, and it can be also written as a linear SOCP problem with three linear SOC constraints, thereby being solved efficiently by interior point methods for SOCP [5,19].

Notice that the constraints in Formulation (9) have the same structure as those related to the MPM and MEMPM problems.

Saketha Nath and Bhattacharyya demonstrate that RMMC is equivalent to maximizing the separation margin between the two class patterns represented as ellipsoids instead of convex hulls (see [4] for details).

Remark 5. A kernel version can be derived for nonlinear classification; see [4] for a detailed formalization of this formulation.

3. Regularized minimax probability machine for classification

In this section, a regularized formulation is proposed for extending the MPM and MEMPM approaches. Following the reasoning behind SVM classification and RMMC, the ℓ_2-norm is used for margin maximization and to avoid ill-posed problems. Besides margin maximization, our proposal also minimizes the worst-case error rates for future data η_1 and η_2 related to the two classes.

Our approach is first derived as a linear method in Section 3.1, and subsequently extended as a kernel-based model in Section 3.2. Finally, the optimization strategy proposed for solving the NSOCP problems is described in Section 3.3.

3.1. Linear regularized minimax probability machine

Formally, our proposal finds $\mathbf{w} \neq \mathbf{0}$, b, η_1, and η_2 by solving the following model:

$$\begin{aligned} \min_{\mathbf{w},b,\eta_1,\eta_2} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1\eta_1 + C_2\eta_2 \\ \text{s.t.} \quad & \inf_{\mathbf{X}_1 \sim (\mu_1, \Sigma_1)} \Pr\{\mathbf{w}^\top\mathbf{X}_1 + b \geq 0\} \geq 1 - \eta_1, \\ & \inf_{\mathbf{X}_2 \sim (\mu_2, \Sigma_2)} \Pr\{\mathbf{w}^\top\mathbf{X}_2 + b \leq 0\} \geq 1 - \eta_2, \end{aligned} \tag{10}$$

where $\mathbf{X}_k$ are the n-dimensional random vectors that generate the examples from class $k = 1, 2$, and $(\boldsymbol{\mu}_k, \Sigma_k)$ are their corresponding mean and covariance matrices, respectively. Additionally, $C_k > 0$ is a trade-off parameter. Variables $\eta_k \in (0, 1)$ can be interpreted as the upper bounds for the misclassification probability of class k in a worst-case setting.

Thanks to an appropriate application of the multivariate Chebyshev inequality (cf. Theorem 2.1), Formulation (10) can be written as

$$\begin{aligned} \min_{\mathbf{w},b,\eta_1,\eta_2} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1\eta_1 + C_2\eta_2 \\ \text{s.t.} \quad & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq \sqrt{(1-\eta_1)/\eta_1}\sqrt{\mathbf{w}^\top \Sigma_1 \mathbf{w}}, \\ & -\mathbf{w}^\top\boldsymbol{\mu}_2 - b \geq \sqrt{(1-\eta_2)/\eta_2}\sqrt{\mathbf{w}^\top \Sigma_2 \mathbf{w}}, \\ & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq 0, \\ & -\mathbf{w}^\top\boldsymbol{\mu}_2 - b \geq 0, \\ & \eta_1, \eta_2 \geq 0. \end{aligned} \tag{11}$$

Note that the first four constraints of Problem (11) are positively homogeneous in $\mathbf{w}, b$; that is, if $(\eta_1, \eta_2, \mathbf{w}, b)$ is a feasible solution, then $(\eta_1, \eta_2, t\mathbf{w}, tb)$ is also feasible for any $t > 0$. Then, without loss of generality, we can assume that $\mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq 1$, $-\mathbf{w}^\top\boldsymbol{\mu}_2 - b \geq 1$. Therefore, the following formulation can be derived by defining $\kappa_k = \sqrt{(1-\eta_k)/\eta_k}$ for $k = 1, 2$:

$$\begin{aligned} \min_{\mathbf{w},b,\kappa_1,\kappa_2} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1\frac{1}{\kappa_1^2+1} + C_2\frac{1}{\kappa_2^2+1} \\ \text{s.t.} \quad & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq \kappa_1\|S_1^\top\mathbf{w}\|, \\ & -\mathbf{w}^\top\boldsymbol{\mu}_2 - b \geq \kappa_2\|S_2^\top\mathbf{w}\|, \\ & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq 1, \\ & -\mathbf{w}^\top\boldsymbol{\mu}_2 - b \geq 1, \\ & \kappa_1, \kappa_2 \geq 0, \end{aligned} \tag{12}$$

where $\Sigma_k = S_kS_k^\top$, for $k = 1, 2$. Note that Formulation (12) is an NSOCP problem since it contains a nonconvex objective function with two nonlinear SOC constraints and four linear constraints. We refer to this formulation as the ℓ_2-MEMPM model.

Remark 6. An ℓ_2-regularized version of the MPM problem can be derived from Formulation (12) by setting $\kappa_1 = \kappa_2$. In this case, the ℓ_2-MPM model follows:

$$\begin{aligned} \min_{\mathbf{w},b,\kappa} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\frac{1}{\kappa^2+1} \\ \text{s.t.} \quad & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq \kappa\|S_1^\top\mathbf{w}\|, \\ & -\mathbf{w}^\top\boldsymbol{\mu}_2 - b \geq \kappa\|S_2^\top\mathbf{w}\|, \\ & \mathbf{w}^\top\boldsymbol{\mu}_1 + b \geq 1, \\ & -\mathbf{w}^\top\boldsymbol{\mu}_2 - b \geq 1, \\ & \kappa \geq 0, \end{aligned} \tag{13}$$

with $C > 0$. Notice that the ℓ_2-MPM model is also an NSOCP problem.

3.2. Kernel-based regularized minimax probability machine

In this section, the ℓ_2-MEMPM and ℓ_2-MPM models (Formulations (12) and (13), respectively) are extended as kernel methods for deriving nonlinear classifiers. Let us denote by m_1 and m_2 the number of elements of the positive and negative class, respectively. Also, we denote by $A \in \Re^{m_1\times n}$ the data matrix for the positive class (i.e. for $y_i = +1$), and by $B \in \Re^{m_2\times n}$ the data matrix for the negative class (i.e. for $y_i = -1$).

The kernel-based version for the ℓ_2-MEMPM is obtained by rewriting the weight vector $\mathbf{w} \in \Re^n$ as $\mathbf{w} = \mathbb{X}^\top\mathbf{s} + M\mathbf{r}$, where $\mathbb{X} = [A;\ B] \in \Re^{m\times n}$ is the data matrix containing both training patterns, M is a matrix whose columns are orthogonal to the data points, and $\mathbf{s}$ and $\mathbf{r}$ are vectors of combining coefficients with the appropriate dimension. Additionally, the empirical estimates of the mean $\boldsymbol{\mu}_k$ and covariance Σ_k are given by

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

with

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

where $\mathbf{e}_k \in \Re^{m_k}$ are all-ones vectors. Thus, one has

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

where

$$\mathbf{g}_k = \frac{1}{m_k}\begin{bmatrix} \mathbf{K}_{1k}\mathbf{e}_k \\ \mathbf{K}_{2k}\mathbf{e}_k \end{bmatrix} \in \Re^m,$$

$$\Xi_k = \frac{1}{m_k}\begin{bmatrix} \mathbf{K}_{1k} \\ \mathbf{K}_{2k} \end{bmatrix}\left(I_{m_k} - \frac{1}{m_k}\mathbf{e}_k\mathbf{e}_k^\top\right)\left[\mathbf{K}_{1k}^\top\ \mathbf{K}_{2k}^\top\right] \in \Re^{m\times m},$$

with $I_{m_k} \in \Re^{m_k\times m_k}$ denoting the identity matrix, and $\mathbf{K}_{11} = AA^\top$, $\mathbf{K}_{12} = \mathbf{K}_{21}^\top = AB^\top$, and $\mathbf{K}_{22} = BB^\top$ are matrices whose elements are inner products between data points. For instance, the (l, s) entry of matrix $\mathbf{K}_{kk'}$ corresponds to $(\mathbf{K}_{kk'})_{ls} = (\mathbf{x}_l^k)^\top\mathbf{x}_s^{k'}$.

The inner product $(\mathbf{x}_l^k)^\top\mathbf{x}_s^{k'}$ can be replaced by a kernel function $\mathcal{K}(\mathbf{x}_l^k, \mathbf{x}_s^{k'})$, where $\mathbf{x}_l^k$ corresponds to the lth vector of the class $k \in \{1, 2\}$, leading to the following nonlinear formulation:

$$\begin{aligned} \min_{\mathbf{s},b,\kappa_1,\kappa_2} \quad & \frac{1}{2}\mathbf{s}^\top\mathbf{K}\mathbf{s} + C_1\frac{1}{\kappa_1^2+1} + C_2\frac{1}{\kappa_2^2+1} \\ \text{s.t.} \quad & \mathbf{s}^\top\mathbf{g}_1 + b \geq \kappa_1\sqrt{\mathbf{s}^\top\Xi_1\mathbf{s}}, \\ & -\mathbf{s}^\top\mathbf{g}_2 - b \geq \kappa_2\sqrt{\mathbf{s}^\top\Xi_2\mathbf{s}}, \\ & \mathbf{s}^\top\mathbf{g}_1 + b \geq 1, \\ & -\mathbf{s}^\top\mathbf{g}_2 - b \geq 1, \\ & \kappa_1, \kappa_2 \geq 0, \end{aligned} \tag{14}$$

where $\mathbf{K} = [\mathbf{K}_{11}, \mathbf{K}_{12}; \mathbf{K}_{21}, \mathbf{K}_{22}]$.

Similarly, the kernel-based version for the ℓ_2-MPM model can be obtained directly by using Formulation (14), leading to the following NSOCP problem:

$$\begin{aligned} \min_{\mathbf{s},b,\kappa} \quad & \frac{1}{2}\mathbf{s}^\top\mathbf{K}\mathbf{s} + C\frac{1}{\kappa^2+1} \\ \text{s.t.} \quad & \mathbf{s}^\top\mathbf{g}_1 + b \geq \kappa\sqrt{\mathbf{s}^\top\Xi_1\mathbf{s}}, \\ & -\mathbf{s}^\top\mathbf{g}_2 - b \geq \kappa\sqrt{\mathbf{s}^\top\Xi_2\mathbf{s}}, \\ & \mathbf{s}^\top\mathbf{g}_1 + b \geq 1, \\ & -\mathbf{s}^\top\mathbf{g}_2 - b \geq 1, \\ & \kappa \geq 0. \end{aligned} \tag{15}$$

3.3. The FDIPA$_{soc}$ strategy for solving NSOCP problems

In order to solve the ℓ_2-MPM and the ℓ_2-MEMPM models in both their linear and kernel versions, we use the interior-point algorithm called FDIPA$_{soc}$ [18], which is designed for solving NSOCP problem (1). Notice that formulations (12), (13), (14), and (15) have the structure of an NSOCP problem. For instance, Formulation (12) can be rewritten by defining $\mathbf{u} = (\mathbf{w}, b, \kappa_1, \kappa_2) \in \Re^{n+3}$, leading to the following objective function:

$$f(\mathbf{u}) = \frac{1}{2}\|\mathbf{w}\|^2 + C_1\frac{1}{\kappa_1^2+1} + C_2\frac{1}{\kappa_2^2+1},$$

and the following constraints:

$$g^1(\mathbf{u}) = (\mathbf{w}^\top\boldsymbol{\mu}_1 + b, k_1S_1^\top\mathbf{w}), \quad g^2(\mathbf{u}) = (-\mathbf{w}^\top\boldsymbol{\mu}_2 - b, k_2S_2^\top\mathbf{w}),$$

$$g^3(\mathbf{u}) = \mathbf{w}^\top\boldsymbol{\mu}_1 + b - 1,\ g^4(\mathbf{u}) = -\mathbf{w}^\top\boldsymbol{\mu}_2 - b - 1,\ g^5(\mathbf{u}) = \kappa_1,$$
$$g^6(\mathbf{u}) = \kappa_2,$$

where $\mathcal{K}^{m_1} = \mathcal{K}^{m_2} = \mathcal{K}^{n+1}$, $\mathcal{K}^{m_j} = \Re_+$ for $j = 3, \ldots, 6$.

Next, we introduce some notations, and the idea of the algorithm proposed by Canelas et al. [18]. Let $\mathcal{K}$ be the Cartesian product of second-order cones, namely, $\mathcal{K} = \mathcal{K}^{m_1} \times \cdots \times \mathcal{K}^{m_J}$. Let us define $\mathbf{g}(\mathbf{u}) := (g^1(\mathbf{u}), \ldots, g^J(\mathbf{u})) \in \Re^m$, where $m = \sum_{j=1}^J m_j$. We denote by $\Omega := \{\mathbf{u} :\ \mathbf{g}(\mathbf{u}) \in \mathcal{K}\}$ the feasible set, by $\mathcal{S}_{++}^n$ the set of symmetric positive definite matrices, and by $\text{Arw}(\mathbf{u}) := \begin{pmatrix} u_1 & \bar{\mathbf{u}}^\top \\ \bar{\mathbf{u}} & u_1I_{m-1} \end{pmatrix}$ the arrow matrix conformed by $\mathbf{u} = (u_1, \bar{\mathbf{u}}) \in \Re \times \Re^{m-1}$. We say that two vectors $\mathbf{u}$ and $\mathbf{v}$ operator commutes if $\text{Arw}(\mathbf{u})\,\text{Arw}(\mathbf{v}) = \text{Arw}(\mathbf{v})\,\text{Arw}(\mathbf{u})$.

Let $L : \Re^n \times \Re^m \to \Re$ be the Lagrangian function associated with the NSOCP problem:

$$L(\mathbf{u}, \mathbf{z}) = f(\mathbf{u}) - \langle\mathbf{g}(\mathbf{u}), \mathbf{z}\rangle,$$

where $\mathbf{z} = (\mathbf{z}^1, \ldots, \mathbf{z}^J) \in \Re^m$ is the Lagrange multiplier vector. Then, the Karush–Kuhn–Tucker (KKT) conditions for Problem (1) are given by

$$\nabla_{\mathbf{u}} L(\mathbf{u},\mathbf{z}) = \nabla f(\mathbf{u}) - \mathcal{J}\mathbf{g}(\mathbf{u})^\top \mathbf{z} = \mathbf{0}, \tag{16}$$

$$\langle g^j(\mathbf{u}), \mathbf{z}^j \rangle = 0, \quad j = 1,\dots,J, \tag{17}$$

$$g^j(\mathbf{u}),\ \mathbf{z}^j \in \mathcal{K}^{m_j}, \quad j = 1,\dots,J. \tag{18}$$

Note that the relation (17) can be replaced by $\mathrm{Arw}(g^j(\mathbf{u}))\mathbf{z}^j = \mathrm{Arw}(\mathbf{z}^j)g^j(\mathbf{u}) = 0$, for $j = 1,\dots,J$ (see [5, Lemma 15]).

The proposed algorithm is based on a Newton-like iterative process for solving the nonlinear system of equations (16)–(17), which can be stated as follows:

$$\begin{pmatrix} B & -\mathcal{J}\mathbf{g}(\mathbf{u})^\top \\ \mathrm{Arw}(\mathbf{z})\mathcal{J}\mathbf{g}(\mathbf{u}) & \mathrm{Arw}(\mathbf{g}(\mathbf{u})) \end{pmatrix} \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{y}_1 \end{pmatrix} = -\begin{pmatrix} \nabla f(\mathbf{u}) \\ \mathbf{0} \end{pmatrix}, \tag{19}$$

where $\mathcal{J}\mathbf{g}(\mathbf{u})$ denotes the Jacobian matrix of $\mathbf{g}$ at $\mathbf{u}$, $\mathrm{Arw}(\mathbf{z}) = \mathrm{diag}(\mathrm{Arw}(\mathbf{z}^j))$ is a block diagonal matrix with $\mathrm{Arw}(\mathbf{z}^j)$ as its entries, $(\mathbf{u},\mathbf{z})$ is the starting (interior) point of the iteration, and $B \in \mathcal{S}^n_{++}$. Typically, B is chosen as a quasi-Newton estimate of $\nabla^2_{\mathbf{u}} L(\mathbf{u},\mathbf{z})$. In particular, when $B = \nabla^2_{\mathbf{u}} L(\mathbf{u},\mathbf{z})$, we get the well-known Newton iteration of (16)–(17).

It can be shown that $\mathbf{d}_1$ is a descent direction of the objective function [18, Lemma 3.2], but it cannot be taken as a search direction since it is not always a feasible direction when $\mathbf{u}$ is at the boundary of the feasible set. In order to obtain a feasible direction, we add a positive vector on the right side of the second equality of Eq. (19) [18, Lemma 3.3]; that is, we consider the following linear system for $\rho > 0$:

$$\begin{pmatrix} B & -\mathcal{J}\mathbf{g}(\mathbf{u})^\top \\ \mathrm{Arw}(\mathbf{z})\mathcal{J}\mathbf{g}(\mathbf{u}) & \mathrm{Arw}(\mathbf{g}(\mathbf{u})) \end{pmatrix} \begin{pmatrix} \mathbf{d} \\ \hat{\mathbf{y}} \end{pmatrix} = -\begin{pmatrix} \nabla f(\mathbf{u}) \\ \rho\mathbf{z} \end{pmatrix}. \tag{20}$$

Since the feasible direction $\mathbf{d}$ obtained by the system (20) is not necessarily a descent direction for all $\rho > 0$ [18,23], we need to impose a convenient upper bound on ρ. This bound is obtained by imposing the following condition (see [23]):

$$\langle \mathbf{d}, \nabla f(\mathbf{u}) \rangle \le \xi \langle \mathbf{d}_1, \nabla f(\mathbf{u}) \rangle, \quad \xi \in (0,1). \tag{21}$$

Under this assumption, the feasible direction $\mathbf{d}$ will also be a descent direction. The algorithm FDIPA$_{soc}$ for solving the NSOCP problem is presented in the next column (see Algorithm 1).

Algorithm 1 FDIPA$_{soc}$ algorithm

```
Input: $\xi, \eta, \nu \in (0,1)$, $\varphi > 0$, and $\lambda_m > 0$.
Output: Solution $\mathbf{u}^k$.
1: Start with $\mathbf{u}^0 \in \mathrm{int}(\Omega_a) = \{\mathbf{u} \in \mathrm{int}(\Omega) : f(\mathbf{u}) < a\}$, for some
   $a \in \Re$; $\mathbf{z}^0 \in \mathrm{int}(\mathcal{K})$ such that it operator commutes with $\mathbf{g}(\mathbf{u}^0)$,
   and $B^0 \in \mathcal{S}^n_{++}$. Set $k = 0$.
2: Compute $\mathbf{d}_1^k$ and $\mathbf{y}_1^k$ by solving the linear system:
   $$\begin{pmatrix} B^k & -\mathcal{J}\mathbf{g}(\mathbf{u}^k)^\top \\ \mathrm{Arw}(\mathbf{z}^k)\mathcal{J}\mathbf{g}(\mathbf{u}^k) & \mathrm{Arw}(\mathbf{g}(\mathbf{u}^k)) \end{pmatrix} \begin{pmatrix} \mathbf{d}_1^k \\ \mathbf{y}_1^k \end{pmatrix} = -\begin{pmatrix} \nabla f(\mathbf{u}^k) \\ 0 \end{pmatrix}. \tag{22}$$
3: if $\mathbf{d}_1^k = 0$ then
4:     return $\mathbf{u}^k$ and stop.
5: end if
6: Compute $\mathbf{d}_2^k$ and $\mathbf{y}_2^k$ by solving the linear system:
   $$\begin{pmatrix} B^k & -\mathcal{J}\mathbf{g}(\mathbf{u}^k)^\top \\ \mathrm{Arw}(\mathbf{z}^k)\mathcal{J}\mathbf{g}(\mathbf{u}^k) & \mathrm{Arw}(\mathbf{g}(\mathbf{u}^k)) \end{pmatrix} \begin{pmatrix} \mathbf{d}_2^k \\ \mathbf{y}_2^k \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{z}^k \end{pmatrix}. \tag{23}$$
7: if $\langle \mathbf{d}_2^k, \nabla f(\mathbf{u}^k) \rangle > 0$ then
8:     set $\rho^k = \min\left\{\varphi\|\mathbf{d}_1^k\|^2, (\xi - 1)\frac{\langle \mathbf{d}_1^k, \nabla f(\mathbf{u}^k)\rangle}{\langle \mathbf{d}_2^k, \nabla f(\mathbf{u}^k)\rangle}\right\}$.
9: else
10:    set $\rho^k = \varphi\|\mathbf{d}_1^k\|^2$.
11: end if
12: Compute $\mathbf{d}^k = \mathbf{d}_1^k + \rho^k \mathbf{d}_2^k$ and $\hat{\mathbf{y}}^k = \mathbf{y}_1^k + \rho^k \mathbf{y}_2^k$.
13: (Armijo line search): Let $g^j(\mathbf{u}^k) = \lambda_1(g^j(\mathbf{u}^k))\mathbf{v}_1^{jk} + \lambda_2(g^j(\mathbf{u}^k))\mathbf{v}_2^{jk}$
    its spectral decomposition. Compute $t^k$ as the first number of
    the sequence $\{1, \nu, \nu^2, \dots\}$ satisfying
    $f(\mathbf{u}^k + t^k\mathbf{d}^k) \le f(\mathbf{u}^k) + t^k\eta\nabla f(\mathbf{u}^k)^\top \mathbf{d}^k$,
    $\mathbf{g}(\mathbf{u}^k + t^k\mathbf{d}^k) \in \mathrm{int}(\mathcal{K})$, and
    $\lambda_i(g^j(\mathbf{u}^k + t^k\mathbf{d}^k)) \ge \lambda_i(g^j(\mathbf{u}^k))$, if $\langle \mathbf{v}_i^{jk}, \hat{\mathbf{y}}^{jk} \rangle < 0$
    and $\lambda_i(g^j(\mathbf{u}^k)) < \lambda_m$,
    where $\hat{\mathbf{y}}^k = (\hat{\mathbf{y}}^{1k}, \dots, \hat{\mathbf{y}}^{Jk})$, with $\hat{\mathbf{y}}^{jk} \in \Re^{m_j}$.
14: Set $\mathbf{u}^{k+1} = \mathbf{u}^k + t^k\mathbf{d}^k$. Define $\mathbf{z}^{k+1} \in \mathrm{int}(\mathcal{K})$ such that operator
    commutes with $\mathbf{g}(\mathbf{u}^{k+1})$, and $B^{k+1} \in \mathcal{S}^n_{++}$.
15: Replace $k$ by $k + 1$ and repeat from Step 2.
```

The global convergence for the FDIPA$_{soc}$ algorithm can be reached under some assumptions on the parameters. This holds since any accumulation point $\mathbf{u}^*$ of the sequence $\{\mathbf{u}^k\}$ generated by Algorithm 1 is a KKT point of the NSOCP problem. For a proof of this statement, we refer the reader to [18, Theorem 3.8].

4. Experimental results

The proposed regularized methods were applied to an illustrative toy dataset, and to sixteen benchmark datasets from the UCI Repository [24]. First, the geometrical interpretation for our proposal is illustrated in Section 4.1 using the toy dataset. Then, a description of the dataset is provided in Section 4.2, including relevant aspects of the experimental setting, such as model validation and implementation. Finally, the main results are summarized in Section 4.3, including a statistical performance analysis and discussions.

4.1. An illustrative example

The purpose of this analysis is to provide the geometrical interpretation for the proposed ℓ_2-MPM and ℓ_2-MEMPM models, and compare them with the original methods (MPM and MEMPM, respectively). Our reasoning follows from the RMMC method [4]. Given $\boldsymbol{\mu}$, S, and κ, the constraint $\mathbf{w}^\top\boldsymbol{\mu} + b \ge \kappa\|S^\top\mathbf{w}\|$ related to the RMMC method is satisfied if and only if $\mathbf{w}^\top\mathbf{x} + b \ge 0$ for all $\mathbf{x}$ belonging to the ellipsoid.[1] $B(\boldsymbol{\mu}, S, \kappa)$, where $\boldsymbol{\mu}$ denotes its center, S determines its shape, and κ its size. Taking this into account, the MPM and MEMPM methods look for the largest ellipsoids that separate two the two training patterns, while our proposal aim at finding a good balance between maximizing the margin and the size of the ellipsoids.

The toy dataset consists of 30 points of the class +1 generated from a two-dimensional Gaussian distribution with mean $\hat{\boldsymbol{\mu}}_1 = [0; 3]$ and covariance matrix $\hat{\Sigma}_1 = [1, 0; 0, 3]$, and 30 points of the class -1 generated from a two-dimensional Gaussian distribution with mean $\hat{\boldsymbol{\mu}}_2 = [0; -2]$ and covariance matrix $\hat{\Sigma}_2 = [2, 0.5; 0.5, 3]$.

For these experiments, 10-fold cross-validation was used to set the various hyperparameters. The linear versions of the four methods were studied. The average performances in terms of Area Under the Curve (AUCx100) achieved by MPM, ℓ_2-MPM, MEMPM, and ℓ_2-MEMPM were 90.00, 91.67, 90.00, and 93.33, respectively. These results show that our proposal is able to improve predictive performance slightly for the two variants.

[1] Formally an ellipsoid is defined by $B(\boldsymbol{\mu}, S, \kappa) = \{\mathbf{z} \in \Re^n : \mathbf{z} = \boldsymbol{\mu} + \kappa S\mathbf{u}, \|\mathbf{u}\| \le 1\}$.

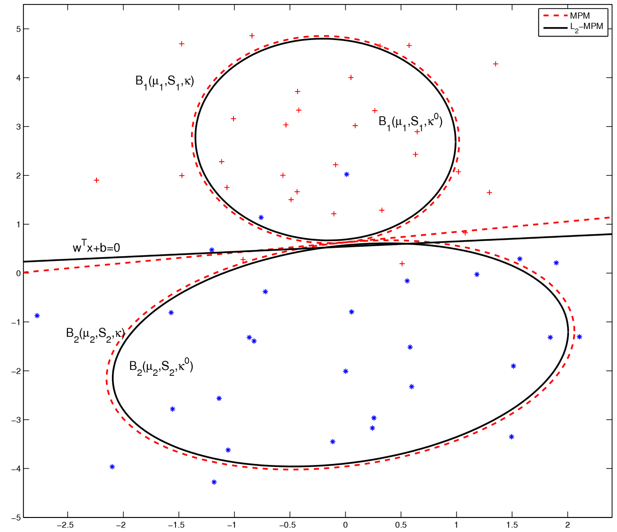

Fig. 1. Geometrical interpretation for the MPM (dashed red lines) and ℓ_2-MPM (solid black lines) approaches. $B_k(\mu_k, S_k, \kappa)$ (with $\kappa = 1.37$) and $B_k(\mu_k, S_k, \kappa^0)$ (with $\kappa^0 = 1.34$) denote the ellipsoids for each method and each class.

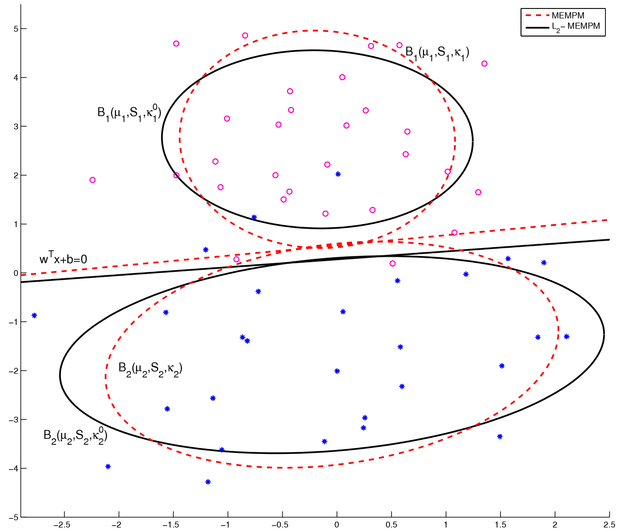

Fig. 2. Geometrical interpretation for the MEMPM (dashed red lines) and ℓ_2-MEMPM (solid black lines) approaches. $B_k(\boldsymbol{\mu}_k, S_k, \kappa_k)$ (with $\kappa_1 = 1.44$ and $\kappa_2 = 1.35$) and $B_k(\boldsymbol{\mu}_k, S_k, \kappa_k^0)$ (with $\kappa_1^0 = 1.62$ and $\kappa_2^0 = 1.18$) denote the ellipsoids for each method and each class.

The solutions obtained by MPM and ℓ_2-MPM are presented in Fig. 1, while the MEMPM and ℓ_2-MEMPM methods are illustrated in Fig. 2. In both figures, the optimal hyperplanes are highlighted together with the ellipsoids that represent the two training patterns. The solid lines are used for our proposals, while the dashed lines represent the original approaches that do not consider regularization. For the MPM method, the ellipsoids are tangent to each other, and hence, the optimal hyperplane is the common tangent to the ellipsoids [1]. In ℓ_2-MPM, in contrast, the optimal hyperplane is tangent to one of the ellipsoids.

It can be observed in Figs. 1 and 2 that the final hyperplanes are relatively similar, however, those constructed with our proposals seem to be flatter in the sense that they tend to ignore the least relevant attribute (the x-axis), leading to a better generalization.

Table 1
Number of variables, sample size, percentage of observations in each class, and imbalance ratio (IR) for all datasets.

Dataset	#features	#examples	%class(min.,maj.)	IR
Class-balanced datasets				
TITA	3	2201	(32.3,67.7)	2.1
DIA	8	768	(34.9,65.1)	1.9
HEART	13	270	(44.4,55.6)	1.25
AUS	14	690	(44.5,55.5)	1.2
IMAGE5	19	2310	(38.1,61.9)	1.6
PHONE	19	5404	(29.3,70.7)	2.4
RING	20	7400	(49.5,50.5)	1.0
WAVE	21	5000	(33.1,66.9)	2.0
GER	24	1000	(30.0,70.0)	2.3
WBC	30	569	(37.3,62.7)	1.7
IONO	34	351	(35.9,64.1)	1.8
SPLICE	60	1000	(48.3,51.7)	1.1
Class-imbalanced datasets				
YEAST3	8	1484	(11.0,89.0)	8.1
YEAST4	8	1484	(3.4,96.6)	28.1
FLARE	10	1389	(4.9,95.1)	19.4
IMAGE1	19	2310	(14.3,85.7)	6.0

4.2. Experimental setting and datasets

We compared the predictive performance of our proposals and alternative approaches on the following binary classification datasets from the UCI [24] and KEEL [25] repositories: Titanic (TITA), Pima Indians Diabetes (DIA), Heart/Statlog (HEART), Australian Credit (AUS), Phoneme (PHONE), Ring, German Credit (GERMAN), Wisconsin Breast Cancer (WBC), Ionosphere (IONO), and Splice.

One goal of the experimental section is to assess the influence of skewness in the class distribution, also known as the class-imbalance issue. This is of particular interest for assessing the gain in considering two different class recall variables α_k, with $k \in \{1, 2\}$ (MEMPM and ℓ_2-MEMPM approaches), instead of a single one α (MPM and ℓ_2-MPM approaches). Therefore, we study datasets with a wide range of Imbalanced Ratios (i.e. the quotient between the number of examples of the majority class and the minority class).

The following multiclass classification datasets were cast into class-imbalanced binary classification problems and used for benchmarking: Solar Flares, in which two classes were constructed from the occurrence of zero M-class flares in 24 h versus one or more in the same time period (FLARE); Image Segmentation (IMAGE), in which the positive class is image 1 and image 5 (IMAGE1 and IMAGE5, respectively), while the remaining classes were used as the majority class; Waveform (WAVE), in which the positive class is wave class 1; and Yeast, in which class and ME3 and ME2 were studied as the minority class, while the negative examples belong to the rest (YEAST3 and YEAST4, respectively). The detailed information on these datasets can be found in the KEEL dataset repository [25]. The relevant meta-data is summarized in Table 1.

We performed a model comparison using 10-fold cross-validation with the Area Under the ROC Curve (AUC) as the performance measure. The following binary classification approaches were studied using linear and Gaussian kernels:

- The well-known soft-margin SVM proposed by Cortes and Vapnik [26].
- The MPM method by Lanckriet et al. [1] (Formulation (2)).
- The MEMPM method by Huang et al. [14] (Formulation (5)).
- The maximum-margin SOCP framework by Saketha Nath and Bhattacharyya [4], which we refer to as the RMMC method (Formulation (9)).

- The twin SVM method by Shao et al. [27] (TWSVM). This popular SVM strategy constructs two nonparallel hyperplanes in such a way that each function is closer to one of the classes, and as far from the other at the same time. This is done by solving two quadratic programming problems of smaller size when compared with the standard soft-margin SVM formulation. See [27,28] for the detailed derivation of this strategy.
- The Nonparallel SVM method by Tian et al. [29] (NPSVM). This technique is a generalization of the TWSVM method, constructing two twin hyperplanes similar to TWSVM. Additionally, NPSVM represents each class using ϵ-insensitive tubes for providing a better alignment between each classification function and the class that it represents. In contrast to TWSVM, the NPSVM method is able to derive a kernel-based model directly from the dual form of the twin problems. We refer the reader to Tian et al. [29] for the detailed derivation of this method.
- The proposed ℓ_2-MPM and ℓ_2-MEMPM approaches (Formulation (13) and Formulation (12), respectively, for the linear versions, and Formulation (14) and Formulation (15), respectively, for the kernel-based versions).

The following values for the hyperparameters were explored: $C, C_1, C_2, C_3, \quad C_4, \sigma \quad \in \quad \{2^{-7}, 2^{-6}, 2^{-5}, \ldots, 2^5, 2^6, 2^7\}$, $\theta \in \{2^{-7}, 2^{-6}, \ldots, 2^{-1}, 1 - 2^{-1}, \ldots, 1 - 2^{-6}, 1 - 2^{-7}\}$, $\epsilon \in \{0.1, 0.2, 0.3, 0.4\}$, and $\alpha_1, \alpha_2 \in \{0.2, 0.4, 0.6, 0.8\}$. Notice that parameter C is included in the soft-margin SVM and ℓ_2-MPM methods; parameters $C_1, \ C_2$ in the ℓ_2-MEMPM method; C_1, C_2, C_3, C_4 in the TWSVM method; C_1, C_2, C_3, C_4, and ϵ in the NPSVM model; θ in the MEMPM method; and α_1 and α_2 in the RMMC model. Gaussian kernel parameter σ is included in all kernel-based approaches.

The following tools were used for implementing the alternative approaches: LIBSVM [30] for soft-margin SVM, the codes developed by Yuan-Hai Shao, author of Twin-Bounded SVM [27], for TWSVM,[2] the QUADPROG Matlab solver for NPSVM, the SeDuMi Matlab Toolbox [6] for RMMC and MPM, and the codes provided by Huang, author of the MEMPM method.[3] We note that our proposal was implemented using our self-developed optimization strategy, called FDIPA$_{soc}$, which is described in Section 3.3.

4.3. Summary of results

Next, a summary of the results is presented for the sixteen datasets. Table 2 shows the best performance for each method in terms of average AUCx100. The best strategy for each dataset is highlighted in bold type. For each method and dataset, the maximum predictive performance between the linear and the Gaussian kernels is presented.

It can be observed in Table 2 that the proposals achieve very positive predictive performances, having the largest AUC in seven of the sixteen datasets. Furthermore, they both perform better than their respective unregularized counterparts (the MPM and MEMPM methods), demonstrating the virtues of including a regularization term for these approaches. The RMMC and TWSVM methods also achieve a very positive predictive performance.

In order to confirm the previous results, the Friedman test and Holm's test are used to assess statistical significance. This approach was recommended in [31] for comparing classification performance among various machine learning methods. First, the average rank for each method is computed based on the AUC value on all datasets. Next, the Friedman test with Iman–Davenport correction is used to assess whether or not all ranks are equal statistically [31]. In case the null hypothesis of equal ranks is rejected, the Holm's post-hoc test is used for pairwise comparisons between the method with the highest rank and those remaining [31].

Table 2
Performance summary for the various binary classification approaches.

Dataset	SVM	RMMC	TWSVM	NPSVM	MPM	ℓ_2-MPM	MEMPM	ℓ_2-MEMPM
TITA	71.1	71.1	71.1	71.1	71.1	70.9	70.0	**72.1**
DIA	72.1	76.3	75.6	76.3	75.2	75.2	72.5	**76.6**
HEART	79.4	84.7	85.0	**85.8**	83.9	84.3	83.4	85.3
AUS	86.2	86.9	**87.6**	87.1	86.4	87.2	86.3	87.2
IMAGE5	67.9	70.7	67.6	69.7	66.3	70.4	57.3	**71.8**
PHONE	88.0	**88.5**	88.4	87.4	86.7	84.9	72.1	86.9
RING	98.0	98.1	**98.1**	96.8	97.5	98.0	75.3	**98.1**
WAVE	88.3	89.0	**89.1**	87.0	88.0	88.0	86.9	88.3
GER	69.4	72.2	72.4	73.0	72.0	73.1	70.7	**73.2**
WBC	97.3	97.4	97.0	**98.4**	96.5	97.4	96.6	97.6
IONO	94.1	95.2	**95.4**	95.2	90.5	95.2	86.0	94.6
SPLICE	88.1	88.7	**88.9**	88.6	81.2	87.6	80.8	88.4
YEAST3	87.0	**93.0**	92.0	92.4	91.7	92.6	91.0	92.6
YEAST4	64.4	85.0	82.3	74.7	81.9	84.1	80.8	**85.2**
FLARE	53.8	73.4	70.3	60.9	69.5	73.2	65.1	**73.3**
IMAGE1	99.7	99.3	**99.8**	**99.8**	98.5	99.3	98.6	99.3

Table 3
Holm's post-hoc test for pairwise comparisons.

Method	Mean rank	Avg. AUC × 100	p value	$\alpha/(j-1)$	Action
ℓ_2-MEMPM	2.53	85.66	–	–	Not reject
RMMC	2.81	85.59	0.75	0.05	Not reject
TWSVM	3.09	85.04	0.52	0.02	Not reject
NPSVM	3.94	84.01	0.10	0.02	Not reject
ℓ_2-MPM	4.25	85.09	0.05	0.01	Not reject
SVM	5.94	81.55	0.00	0.01	Reject
MPM	6.12	83.56	0.00	0.01	Reject
MEMPM	7.31	79.59	0.00	0.01	Reject

The F statistic obtained with the Friedman test and Iman–Davenport correction is $F = 17.1$, with a p value below 0.001, rejecting the null hypothesis of equal ranks. The results for the Holm's test are presented in Table 3 for the various binary classification methods. For each technique we present the average rank, the average AUC × 100, the p value for the Holm's test, the significance threshold, and the outcome of the test. The outcome is 'reject' when the p value is below the significance threshold, implying that the corresponding approach is outperformed by the one with the best ranking. We used $\alpha = 5\%$ as the significance level, with $j = 1, \ldots, 5$ being the overall ranking for a given method.

In Table 3, it can be seen that ℓ_2-MEMPM outperforms MPM, MEMPM, and SVM statistically. According to this analysis, there are no significant differences among the three regularized classifiers that use robust classification, and the twin classification approaches TWSVM and NPSVM. However, the proposed ℓ_2-MEMPM achieves the best overall performance among the eight methods. Interestingly, the MPM and MEMPM achieve the worst predictive performance, and therefore the regularized versions of these methods make them competitive in comparison with state-of-the-art SVM strategies.

The final set of experiments considers the training times for all methods and datasets. This analysis was performed on an HP Envy dv6 with 16 GB RAM (750 GB SSD), and an i7-2620M processor with 2.70 GHz. All methods were implemented on Matlab R2014a and Microsoft Windows 8.1 Operating System (64-bits). The results are reported in Table 4.

In Table 4, it can be seen that all training times are tractable and under seventeen minutes for all datasets. Our proposals

[2] These codes are publicly available on http://www.optimal-group.org/.

[3] These codes are publicly available on http://www.cse.cuhk.edu.hk/irwin.king/software/mempm.

Table 4
Running times, in seconds, for all datasets and methods.

Dataset	SVM	RMMC	TWSVM	NPSVM	MPM	ℓ_2-MPM	MEMPM	ℓ_2-MEMPM
TITA	$0''.203$	$0''.391$	$3''.330$	$2''.558$	$0''.844$	$21''.72$	$0''.168$	$24''.68$
DIA	$0''.061$	$0''.631$	$0''.928$	$1''.039$	$1''.108$	$3''.845$	$0''.766$	$3''.634$
HEART	$0''.008$	$0''.363$	$0''.111$	$1''.419$	$0''.994$	$0''.247$	$0''.434$	$0''.330$
AUS	$0''.050$	$0''.364$	$0''.864$	$1''.605$	$1''.023$	$2''.617$	$1''.419$	$2''.058$
IMAGE5	$0''.766$	$1''.406$	$13''.43$	$10''.62$	$3''.133$	$340''.5$	$0''.031$	$444''.4$
PHONE	$0''.047$	$1''.090$	$476''.4$	$6''.886$	$1''.863$	$26''.06$	$4''.500$	$33''.20$
RING	$1''.609$	$1''.254$	$121''.7$	$24''.85$	$3''.008$	$991''.2$	$0''.801$	$1004''.4$
WAVE	$1''.059$	$1''.113$	$42''.69$	$14''.26$	$2''.168$	$128''.7$	$0''.555$	$174''.8$
GER	$0''.001$	$0''.497$	$1''.591$	$3''.519$	$1''.178$	$6''.805$	$0''.725$	$7''.483$
WBC	$0''.039$	$0''.314$	$0''.418$	$1''.984$	$1''.055$	$12''.33$	$1''.566$	$12''.97$
IONO	$0''.016$	$0''.370$	$0''.216$	$1''.989$	$1''.120$	$1''.908$	$3''.175$	$3''.423$
SPLICE	$0''.266$	$0''.584$	$1''.303$	$11''.88$	$1''.184$	$6''.998$	$1''.563$	$8''.636$
YEAST3	$0''.075$	$0''.394$	$3''.113$	$4''.681$	$0''.986$	$13''.29$	$0''.547$	$15''.52$
YEAST4	$0''.128$	$0''.867$	$2''.300$	$4''.642$	$1''.444$	$13''.07$	$0''.477$	$15''.43$
FLARE	$0''.050$	$0''.539$	$1''.377$	$6''.073$	$1''.125$	$17''.98$	$1''.014$	$24''.47$
IMAGE1	$0''.189$	$0''.764$	$6''.463$	$11''.03$	$1''.384$	$101''.0$	$0''.583$	$120''.1$

Table 5
Best hyperparameter configuration for all datasets and methods.

	SVM	TWSVM		NPSVM			RMMC	ℓ_2-MPM	MEMPM	ℓ_2-MEMPM	
	C	$C_1 = C_2$	$C_3 = C_4$	$C_1 = C_2$	$C_3 = C_4$	ϵ	(α_1, α_2)	C	θ	C_1	C_2
TITA	2^{-2}	2^{0}	2^{-1}	2^{0}	2^{0}	0.1	(0.2,0.6)	2^{3}	2^{-1}	2^{-6}	2^{2}
DIA	2^{2}	2^{-7}	2^{0}	2^{-4}	2^{-5}	0.2	(0.2,0.4)	2^{7}	2^{-2}	2^{-6}	2^{7}
HEART	2^{7}	2^{-2}	2^{-1}	2^{-6}	2^{-7}	0.4	(0.6,0.4)	2^{6}	2^{-1}	2^{1}	2^{-1}
AUS	2^{1}	2^{-4}	2^{-1}	2^{-6}	2^{-3}	0.1	(0.2,0.6)	2^{-2}	2^{-1}	2^{-2}	2^{3}
IMAGE5	2^{4}	2^{-2}	2^{7}	2^{1}	2^{1}	0.2	(0.4,0.4)	2^{7}	2^{-1}	2^{7}	2^{0}
PHONE	2^{7}	2^{-7}	2^{-4}	2^{7}	2^{7}	0.2	(0.2,0.4)	2^{3}	2^{-1}	2^{-7}	2^{4}
RING	2^{7}	2^{0}	2^{1}	2^{-4}	2^{-5}	0.3	(0.6,0.2)	2^{4}	2^{-3}	2^{7}	2^{6}
WAVE	2^{0}	2^{2}	2^{6}	2^{-3}	2^{-1}	0.2	(0.6,0.4)	2^{0}	2^{-1}	2^{-1}	2^{1}
GER	2^{2}	2^{-3}	2^{-1}	2^{6}	2^{6}	0.3	(0.4,0.4)	2^{6}	$1 - 2^{-2}$	2^{5}	2^{6}
WBC	2^{6}	2^{-5}	2^{-2}	2^{4}	2^{1}	0.2	(0.6,0.8)	2^{7}	2^{-1}	2^{6}	2^{5}
IONO	2^{4}	2^{5}	2^{4}	2^{-1}	2^{-4}	0.5	(0.4,0.8)	2^{7}	2^{-1}	2^{7}	2^{7}
SPLICE	2^{-5}	2^{-2}	2^{5}	2^{5}	2^{6}	0.5	(0.2,0.6)	2^{7}	2^{-1}	2^{-6}	2^{2}
YEAST3	2^{7}	2^{-7}	2^{-5}	2^{0}	2^{-1}	0.4	(0.6,0.4)	2^{7}	2^{-1}	2^{4}	2^{7}
YEAST4	2^{1}	2^{-7}	2^{-2}	2^{4}	2^{4}	0.1	(0.2,0.2)	2^{7}	2^{-1}	2^{4}	2^{7}
FLARE	2^{1}	2^{-7}	2^{2}	2^{1}	2^{1}	0.1	(0.8,0.4)	2^{7}	$1 - 2^{-2}$	2^{7}	2^{4}
IMAGE1	2^{7}	2^{-5}	2^{-5}	2^{4}	2^{4}	0.3	(0.4,0.4)	2^{7}	2^{-1}	2^{6}	2^{6}

show longer running times when compared with the alternative approaches, however, this is to be expected since we are proposing the first strategy, to the best of our knowledge, for solving nonlinear SOCP problems in a machine learning context, while the remaining methods consider well-known, highly optimized solvers. We conclude that the additional computational effort is worth doing given the positive predictive results achieved by the regularized ℓ_2-MPM and ℓ_2-MEMPM methods.

Finally, the best hyperparameter configuration for all datasets and classification approaches in their linear versions are reported in Table 5.

5. Concluding remarks

We have proposed a novel classification approach based on the MPM [1] and MEMPM [14] methods, in which we introduce a regularization term that leads to the ℓ_2-MPM and ℓ_2-MEMPM formulations. Our proposals are non-convex SOCP problems, and we propose an efficient strategy, called FDIPA$_{soc}$, for solving differentiable nonlinear SOCP problems [18]. Our methods also share similarities with the RMMC approach proposed by Saketha Nath and Bhattacharyya [4], which also considers a robust framework and includes a regularization term. However, this method assumes that each class recall η is a fixed parameter, rather than aiming at maximizing them in the optimization process. Therefore, our proposal reduces the number of hyperparameters that need to be tuned with a validation process.

From our experiments, we conclude that the use of a regularization strategy leads to significant improvements in performance, and may be worth losing the convexity of the problem. We propose a suitable algorithm for solving non-convex second-order cone programming formulations, such as ℓ_2-MPM and ℓ_2-MEMPM, which achieves the best results in tractable running times.

There are many opportunities for future research. Several applications can benefit from robust approaches for binary classification. Robustness in artificial intelligence refers to the effectiveness of a method when being tested on new data that has a distribution that is slightly different from the training set [2–4]. In this sense, most business applications face changing environments, such as evolving granting policies in credit scoring, or dynamic fraud patterns in fraud prediction. Robust methods such as those proposed in this study can be useful for enhancing predictive performance. Furthermore, since these methods optimize the two class recalls independently, they are suitable for dealing with the class-imbalance problem. Notice that the previously mentioned business analytics tasks usually face this issue. Finally, the profit that leads a classifier can be computed for a given task and incorporated in the modeling process. Recent work on profit metrics includes the adaptation of decision trees [32] and logistic regression [33] for maximizing profit within the model training, and these ideas can be adapted to our proposals.

Acknowledgments

The first author was supported by FONDECYT, Chile project 1160738, while the second and third authors were supported by FONDECYT, Chile project 1160894. This research was partially funded by the Complex Engineering Systems Institute, ISCI, Chile

(ICM-FIC: P05-004-F, CONICYT: FB0816). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

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

[2] J. López, S. Maldonado, Robust twin support vector regression via second-order cone programming, Knowl.-Based Syst. 152 (2018) 83–93.

[3] J. López, S. Maldonado, M. Carrasco, Double regularization methods for robust feature selection and svm classification via dc programming, Inf. Sci. 429 (2018) 377–389.

[4] J. Saketha 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.

[5] F. Alizadeh, D. Goldfarb, Second-order cone programming, Math. Program. 95 (1, Ser. B) (2003) 3–51.

[6] J.F. Sturm, Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optim. Methods Softw. 11/12 (1–4) (1999) 625–653.

[7] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.

[8] W. Liu, L. Zhang, D. Tao, J. Cheng, Support vector machine active learning by hessian regularization, J. Vis. Commun. Image Represent. 49 (2017) 47–56.

[9] D. Tao, L. Jin, W. Liu, X. Li, Hessian regularized support vector machines for mobile image annotation on the cloud, IEEE Trans. Multimed. 15 (4) (2013) 833–844.

[10] B.D. Barkana, I. Saricicek, B. Yildirim, Performance analysis of descriptive statistical features in retinal vessel segmentation via fuzzy logic, ann, svm, and classifier fusion, Knowl.-Based Syst. 118 (2017) 165–176.

[11] A. Wang, N. An, G. Chen, L. Liu, G. Alterovitz, Subtype dependent biomarker identification and tumor classification from gene expression profiles, Knowl.-Based Syst. 146 (2018) 104–117.

[12] Z. -Y. Chen, Z. -P. Fan, Distributed customer behavior prediction using multiplex data: A collaborative MK-SVM approach, Knowl.-Based Syst. 35 (2012) 111–119.

[13] S. Maldonado, C. Bravo, J. Pérez, J. López, Integrated framework for profit-based feature selection and svm classification in credit scoring, Decis. Support Syst. 104 (113–121) (2017).

[14] K. Huang, H. Yang, I. King, M. Lyu, L. Chan, The minimum error minimax probability machine, in: J. Mach. Learn. Res., J. Mach. Learn. Res. 5 (2004) 1253–1286.

[15] E.H. Fukuda, P.J.S. Silva, M. Fukushima, Differentiable exact penalty functions for nonlinear second-order cone programs, SIAM J. Optim. 22 (4) (2012) 1607–1633.

[16] H. Kato, M. Fukushima, An sqp-type algorithm for nonlinear second-order cone programs, Optim. Lett. 1 (2) (2007) 129–144.

[17] H. Yamashita, H. Yabe, A primal-dual interior point method for nonlinear optimization over second-order cones, Optim. Methods Softw. 24 (3) (2009) 407–426.

[18] A. Canelas, M. Carrasco, J. López, A feasible direction algorithm for nonlinear second-order cone programs, Optim. Methods Softw. 0 (0) (2018) 1–20, http://dx.doi.org/10.1080/10556788.2018.1506452.

[19] F. Alvarez, J. López, H. Ramírez C., Interior proximal algorithm with variable metric for second-order cone programming: applications to structural optimization and support vector machines, Optim. Methods Softw. 25 (6) (2010) 859–881.

[20] Dimitri P. Bertsekas, Nonlinear Programming, second ed., Athena Scientific, 1999.

[21] K. Huang, H. Yang, I. King, M.R. Lyu, Maximizing sensitivity in medical diagnosis using biased minimax probability machine, IEEE Trans. Biomed. Eng. 53 (5) (2006) 821–831.

[22] B. Gu, X. Sun, V.S. Sheng, Structural minimax probability machine, IEEE Trans. Neural Netw. Learn. Syst. 28 (7) (2017) 1646–1656.

[23] J. Herskovits, Feasible direction interior-point technique for nonlinear optimization, J. Optim. Theory Appl. 99 (1) (1998) 121–146.

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

[25] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL Data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, J. Mult.-Valued Logic Soft Comput. 17 (2–3) (2011) 255–287.

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

[27] Y.H. Shao, C.H. Zhang, X.B. Wang, N.Y. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Netw. 22 (6) (2011) 962–968.

[28] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.

[29] Y. Tian, Z. Qi, X. Ju, Y. Shi, X. Liu, Nonparallel support vector machines for pattern classification, IEEE Trans. Cybern. 44 (7) (2014) 1067–1079.

[30] 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.

[31] J. Demšar, Statistical comparisons of classifiers over multiple data set, J. Mach. Learn. Res. (2006) 1–30.

[32] S. Höppner, E. Stripling, B. Baesens, S. vanden Broucke, T. Verdonck, Profit Driven Decision Trees for Churn Prediction, arXiv:1712.08101 [stat.ML], 2017.

[33] E. Stripling, S. vanden Broucke, K. Antonio, B. Baesens, M. Snoeck, Profit maximizing logistic model for customer churn prediction using genetic algorithms, Swarm Evol. Comput. (2017).