Expert Systems With Applications 54 (2016) 95–104

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A second-order cone programming formulation for nonparallel hyperplane support vector machine

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

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

ARTICLE INFO

Keywords:
Support vector classification
Nonparallel hyperplane SVM
Second-order cone programming

ABSTRACT

Expert systems often rely heavily on the performance of binary classification methods. The need for accurate predictions in artificial intelligence has led to a plethora of novel approaches that aim at correctly predicting new instances based on nonlinear classifiers. In this context, Support Vector Machine (SVM) formulations via two nonparallel hyperplanes have received increasing attention due to their superior performance. In this work, we propose a novel formulation for the method, Nonparallel Hyperplane SVM. Its main contribution is the use of robust optimization techniques in order to construct nonlinear models with superior performance and appealing geometrical properties. Experiments on benchmark datasets demonstrate the virtues in terms of predictive performance compared with various other SVM formulations. Managerial insights and the relevance for intelligent systems are discussed based on the experimental outcomes.

1. Introduction

Support Vector Machine is one of the most popular tools used for prediction in intelligent systems. Its superior performance and flexibility are appealing virtues that lead to numerous extensions. SVM has proved to be very effective in various expert systems applications, such as medical diagnosis (Ríos & Erazo, 2016), churn prediction (Ali & Aritürk, 2014), and human resources analytics (Saradhi & Palshikar, 2011).

Recently, second-order cone programming (SOCP) has been used not only as an alternative optimization scheme for SVM (Debnath, Muramatsu, & Takahashi, 2005), but also to derive robust formulations that follow the SVM principle of maximum margin (Maldonado & López, 2014a; Nath & Bhattacharyya, 2007). The goal of such models is to construct one that correctly classifies most instances of each training pattern even for the worst distribution of the class-conditional densities with a given mean and covariance matrix. Such methods have proved to be very effective in terms of classification performance (Maldonado & López, 2014b).

On the other hand, there is a promising new stream of research that extends SVM to constructing two nonparallel hyperplanes in such a way that each one is close to one of the classes, and as far as possible from the other. The most popular approach is Twin SVM (Jayadeva, Khemchandani, & Chandra, 2007; Shao, Zhang, Wang, & Deng, 2011), while some other extensions, such as Nonparallel Hyperplane SVM (NH-SVM) (Shao, Chen, & Deng, 2014), have also been proposed in the literature, claiming successful results. Twin SVM splits the original problem into two smaller subproblems, and the two hyperplanes are constructed independently. In contrast, NH-SVM solves a single problem to obtain both classifiers simultaneously.

In this work, we propose a novel SVM-based method that extends the ideas of NH-SVM to second-order cones. The approach constructs two nonparallel classifiers, and represents each training pattern by an ellipsoid characterized by the mean and covariance of each class, instead of the reduced convex hulls used in NH-SVM. Originally developed for linear classifiers, the method is also adapted to construct nonlinear classification functions via the kernel trick. The use of ellipsoids for SVM modeling has been applied successfully in the context of expert systems (Czarnecki & Tabor, 2014).

This paper is organized as follows: in Section 2 we present the relevant SVM formulations for this work: Twin SVM, NH-SVM, and SOCP-SVM. The proposed method based on SOCP for Nonparallel Hyperplane SVM is described in Section 3. Experimental results using seven benchmark data sets are given in Section 4. Finally, Section 5 provides the main conclusions of this work, discussing managerial insights and addressing future developments in the context of expert and intelligent systems.

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

http://dx.doi.org/10.1016/j.eswa.2016.01.044
0957-4174/

2. Prior work in SVM classification

In this section, we discuss the relevant SVM formulations in this work: standard soft-margin SVM (Cortes & Vapnik, 1995), Twin SVM (Jayadeva et al., 2007; Shao et al., 2011), Nonparallel Hyperplane SVM (Shao et al., 2014), and SVM based on second-order cone programming (Nath & Bhattacharyya, 2007).

2.1. Soft-margin support vector machine

Given a set of training examples and their respective labels ($\mathbf{x}_i$, y_i), where $\mathbf{x}_i \in \Re^n$, $i = 1, \ldots, m$ and $y_i \in \{-1, +1\}$, the soft-margin SVM formulation aims at finding a classification function of the form $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ by solving the following quadratic programming problem (QPP):

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

where $\boldsymbol{\xi} \in \Re^m$ is a set of slack variables and $C > 0$ is a regularization parameter.

A non-linear classification function can be obtained via the Kernel Trick on the dual of Formulation (1) (Schölkopf & Smola, 2002). This kernel-based SVM formulation follows:

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

where $\boldsymbol{\alpha} \in \Re^m$ is the set of dual variables corresponding to the constraints in (1). In this work we use the *Gaussian kernel*, which usually lead to best empirical results (see e.g. Maldonado, Weber, and Basak (2011); Schölkopf and Smola. (2002)), and has the following form:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_s) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_s||^2}{2\sigma^2}\right), \tag{3}$$

where σ is a parameter that controls the width of the kernel (Schölkopf and Smola. (2002)).

2.2. Twin support vector machine

The twin SVM performs classification by using two nonparallel hyperplanes obtained by solving two smaller-sized QPPs (Jayadeva et al., 2007). Let us denote the cardinality of the positive (negative) class by m_1 (m_2), and by $A \in \Re^{m_1 \times n}$ ($B \in \Re^{m_2 \times n}$) the data matrix related to the positive (negative) class. The linear Twin SVM formulation follows:

$$\begin{aligned} \min_{\mathbf{w}_1,b_1,\boldsymbol{\xi}_2} \quad & \frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{c_1}{2}(\|\mathbf{w}_1\|^2 + b_1^2) + c_3\mathbf{e}_2^\top\boldsymbol{\xi}_2 \\ \text{s.t.} \quad & -(B\mathbf{w}_1 + \mathbf{e}_2 b_1) \geq \mathbf{e}_2 - \boldsymbol{\xi}_2, \\ & \boldsymbol{\xi}_2 \geq 0, \end{aligned} \tag{4}$$

and

$$\begin{aligned} \min_{\mathbf{w}_2,b_2,\boldsymbol{\xi}_1} \quad & \frac{1}{2}\|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 + \frac{c_2}{2}\left(\|\mathbf{w}_2\|^2 + b_2^2\right) + c_4\mathbf{e}_1^\top\boldsymbol{\xi}_1 \\ \text{s.t.} \quad & (A\mathbf{w}_2 + \mathbf{e}_1 b_2) \geq \mathbf{e}_1 - \boldsymbol{\xi}_1, \\ & \boldsymbol{\xi}_1 \geq 0. \end{aligned} \tag{5}$$

Formulation (4)–(5) constructs two hyperplanes $\mathbf{w}_k^\top\mathbf{x} + b_k = 0$, $k = 1, 2$, such that each one is closer to instances of one of the two classes and is as far as possible from those of the other class. A new data point $\mathbf{x}$ is assigned to k^* according to its proximity to the hyperplanes based on the following rule:

$$k^* = \underset{k=1,2}{\operatorname{argmin}}\left\{d_k(\mathbf{x}) := \frac{|\mathbf{w}_k^\top\mathbf{x} + b_k|}{\|\mathbf{w}_k\|}\right\}, \tag{6}$$

where d_k corresponds to the perpendicular distance of the data sample $\mathbf{x}$ from hyperplane $\mathbf{w}_k^\top\mathbf{x} + b_k = 0$, $k = 1, 2$. The scalars c_1, c_2, c_3, and c_4 are positive parameters, and $\mathbf{e}_1$ and $\mathbf{e}_2$ are vectors of ones of appropriate dimensions. We refer to Formulation (4)–(5) as Twin-Bounded SVM (TB-SVM) (Shao et al., 2011), which extends the original Twin SVM (TW-SVM) formulation (Jayadeva et al., 2007). Both problems are equivalent if $c_1 = c_2 = \epsilon$, with $\epsilon > 0$ a fixed small parameter. The dual formulation of Twin-Bounded SVM can be found by Shao et al. (2011).

The linear Twin SVM can be extended to non-linear classification surfaces of the form $\mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{u}_k + b_k = 0$ ($k = 1, 2$) via kernel functions by solving the following quadratic problems (kernel-based Twin SVM):

$$\begin{aligned} \min_{\mathbf{u}_1,b_1,\boldsymbol{\xi}_2} \quad & \frac{1}{2}\left\|\mathcal{K}(A^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_1 b_1\right\|^2 + \frac{c_1}{2}(\|\mathbf{u}_1\|^2 + b_1^2) + c_3\mathbf{e}_2^\top\boldsymbol{\xi}_2 \\ \text{s.t.} \quad & -(\mathcal{K}(B^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_2 b_1) \geq \mathbf{e}_2 - \boldsymbol{\xi}_2, \\ & \boldsymbol{\xi}_2 \geq 0, \end{aligned} \tag{7}$$

and

$$\begin{aligned} \min_{\mathbf{u}_2,b_2,\boldsymbol{\xi}_1} \quad & \frac{1}{2}\left\|\mathcal{K}(B^\top, \mathbb{X})\mathbf{u}_2 + \mathbf{e}_2 b_2\right\|^2 + \frac{c_2}{2}(\|\mathbf{u}_2\|^2 + b_2^2) + c_4\mathbf{e}_1^\top\boldsymbol{\xi}_1 \\ \text{s.t.} \quad & (\mathcal{K}(A^\top, \mathbb{X})\mathbf{u}_2 + \mathbf{e}_1 b_2) \geq \mathbf{e}_1 - \boldsymbol{\xi}_1, \\ & \boldsymbol{\xi}_1 \geq 0, \end{aligned} \tag{8}$$

where $\mathbb{X} = [A^\top\ B^\top] \in \Re^{n \times m}$ is the matrix that combines both training patterns sorted by class, and $\mathcal{K} : \Re^n \times \Re^n \to \Re$ is a kernel function (Schölkopf and Smola. (2002)).

2.3. Nonparallel hyperplane SVM (NH-SVM)

The NH-SVM approach constructs two nonparallel hyperplanes simultaneously by solving a single QPP. Similarly to Twin SVM, the linear NH-SVM formulation finds two hyperplanes in $\Re^n$ such that each classifier is close to one of the training patterns and is as far as possible from the other. The main difference compared to Twin SVM is that, since one single QPP is constructed, both hyperplanes are simultaneously optimized in the same formulation. The linear NH-SVM formulation follows:

$$\begin{aligned} \min_{\substack{\mathbf{w}_k,b_k,\boldsymbol{\xi}_k \\ k=1,2}} \quad & \frac{1}{2}\left(\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2\right) \\ & + \frac{c_1}{2}\left(\|\mathbf{w}_1\|^2 + b_1^2 + \|\mathbf{w}_2\|^2 + b_2^2\right) + \frac{c_2}{2}\left(\mathbf{e}_1^\top\boldsymbol{\xi}_1 + \mathbf{e}_2^\top\boldsymbol{\xi}_2\right) \\ \text{s.t.} \quad & A\mathbf{w}_1 + \mathbf{e}_1 b_1 - A\mathbf{w}_2 - \mathbf{e}_1 b_2 \geq \mathbf{e}_1 - \boldsymbol{\xi}_1, \\ & B\mathbf{w}_2 + \mathbf{e}_2 b_2 - B\mathbf{w}_1 - \mathbf{e}_2 b_1 \geq \mathbf{e}_2 - \boldsymbol{\xi}_2, \\ & \boldsymbol{\xi}_1 \geq 0, \boldsymbol{\xi}_2 \geq 0, \end{aligned} \tag{9}$$

where c_1, $c_2 > 0$ are regularization parameters (Shao et al., 2014). A point $\mathbf{x}$ in $\Re^n$ is assigned to class k^* by identifying the nearest hyperplane according to Eq. (6).

The computation of the Lagrangian and the Karush–Kuhn–Tucker (KKT) conditions leads to the following dual formulation for Problem (9):

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathbf{e}^\top\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top\bar{A}^\top\left[(H^\top H + c_1 I)^{-1} + (G^\top G + c_1 I)^{-1}\right]\bar{A}\boldsymbol{\alpha}, \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\alpha} \leq c_2\mathbf{e}, \end{aligned}$$

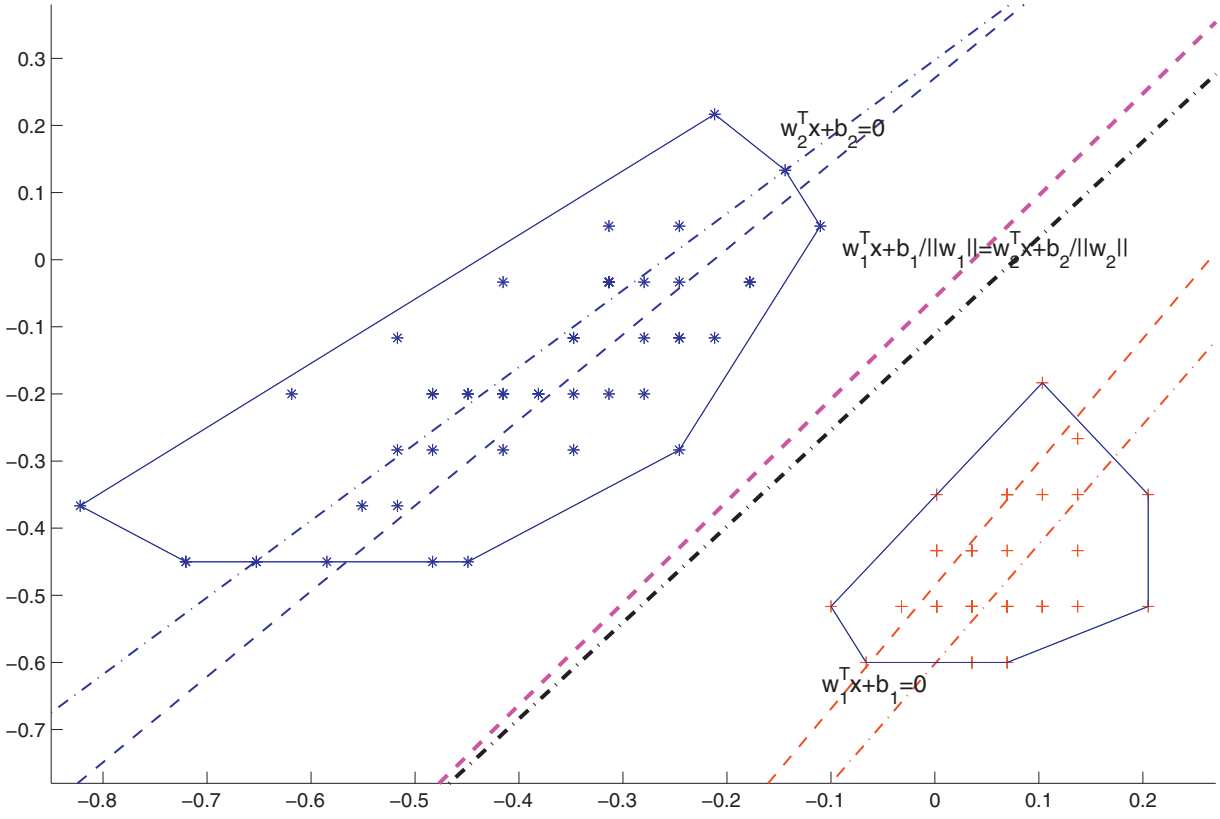

Fig. 1. Geometric interpretation for Twin SVM and NH-SVM.

where $H = [A, \mathbf{e}_1] \in \Re^{m_1 \times (n+1)}$, $G = [B, \mathbf{e}_2] \in \Re^{m_2 \times (n+1)}$, $\bar{A} = [H^\top, -G^\top] \in \Re^{(n+1)\times m}$, and $\mathbf{e} = [\mathbf{e}_1^\top, \mathbf{e}_2^\top]^\top \in \Re^m$. Using this formulation, the values of $\mathbf{w}_k$ and b_k $(k = 1, 2)$ are computed as

$$\mathbf{v}_1 = (H^\top H + c_1 I)^{-1} \bar{A}\boldsymbol{\alpha} \quad \text{and} \quad \mathbf{v}_2 = (G^\top G + c_1 I)^{-1} \bar{A}\boldsymbol{\alpha},$$

where $\mathbf{v}_k = [\mathbf{w}_k^\top, b_k]^\top \in \Re^{n+1}$ for $k = 1, 2$. Similarly to Twin SVM, a kernel-based formulation NH-SVM can be obtained via the *kernel trick*. This formulation is given by

$$\begin{aligned} \min_{\substack{\mathbf{u}_k, b_k, \boldsymbol{\xi}_k \\ k=1,2}} \; & \frac{1}{2}\left(\|\mathcal{K}(A^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_1 b_1\|^2 + \|\mathcal{K}(B^\top, \mathbb{X})\mathbf{u}_2 + \mathbf{e}_2 b_2\|^2\right) + \\ & \frac{c_1}{2}\left(\|\mathbf{u}_1\|^2 + b_1^2 + \|\mathbf{u}_2\|^2 + b_2^2\right) + \frac{c_2}{2}\left(\mathbf{e}_1^\top \xi_1 + \mathbf{e}_2^\top \xi_2\right) \qquad (10) \\ \text{s.t. } & \mathcal{K}(A^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_1 b_1 - \mathcal{K}(A^\top, \mathbb{X})\mathbf{u}_2 - \mathbf{e}_1 b_2 \geq \mathbf{e}_1 - \boldsymbol{\xi}_1, \\ & \mathcal{K}(B^\top, \mathbb{X})\mathbf{u}_2 + \mathbf{e}_2 b_2 - \mathcal{K}(B^\top, \mathbb{X})\mathbf{u}_1 - \mathbf{e}_2 b_1 \geq \mathbf{e}_2 - \boldsymbol{\xi}_2, \\ & \boldsymbol{\xi}_1 \geq 0, \boldsymbol{\xi}_2 \geq 0, \end{aligned}$$

where c_1, and c_2 are positive parameters.

Fig. 1 presents the geometrical interpretation of Twin SVM and NH-SVM in a two-dimensional toy data set. The dashed lines represent the three hyperplanes constructed with Twin SVM: the two nonparallel classifiers over the training patterns and the one that defines the decision rule between both twin hyperplanes. Similarly, The dot-dash lines correspond to the hyperplanes defined by NH-SVM.

In Fig. 1 we observe the small differences between Twin SVM and NH-SVM in terms of the construction of the twin hyperplanes. Both methods construct a decision rule adequately that classifies all training points correctly for this toy data set, although the decision rules are slightly different. The NH-SVM method has the theoretical advantage that it optimizes both twin hyperplanes in the same optimization problem, leading to potentially better predictive performance. On the other hand, Twin SVM splits the formulation into two subproblems, providing more efficient training according to the *divide and conquer* paradigm.

Besides NH-SVM, several extensions for Twin SVM have also been proposed in the literature. Some of the tasks explored with Twin SVM are feature selection (Bai, Wang, Shao, & Deng, 2014), least squares classification (Kumar & Gopal, 2009; Peng, 2010), and weighted regression (Xu & Wang, 2012).

2.4. Second-order cone programming SVM

In this section we introduce the robust SVM version based on second-order cones presented by Nath and Bhattacharyya (2007). Let $\mathbf{X}_k$ be a random variable that generates the training samples from class $k = 1, 2$, with mean vectors and covariance matrices given by $(\boldsymbol{\mu}_k, \Sigma_k)$, where $\Sigma_k \in \Re^{n \times n}$ are symmetric positive semidefinite matrices. The method constructs a maximum-margin classifier such that the probability of false-negative (resp. false-positive) errors does not exceed $1 - \eta_1$ (resp. $1 - \eta_2$), with η_1, $\eta_2 \in (0, 1)$. This problem can be formulated as the following quadratic chance-constrained programming model:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \Pr\{\mathbf{w}^\top \mathbf{X}_1 + b \geq 1\} \geq \eta_1, \qquad (11) \\ & \Pr\{\mathbf{w}^\top \mathbf{X}_2 + b \leq -1\} \geq \eta_2. \end{aligned}$$

A robust setting can be defined from the previous formulation by requiring that each training pattern k has to be correctly classified, up to the rate η_k, even for the worst data distribution. To achieve this goal, the probability constraints in (11) are replaced with their robust counterparts:

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

where $\mathbf{X} \sim (\boldsymbol{\mu}, \Sigma)$ refers a family of distributions which a common mean $\boldsymbol{\mu}$, and covariance Σ.

Thanks to an appropriate application of the multivariate Chebyshev inequality (Lanckriet, Ghaoui, Bhattacharyya, & Jordan, 2003, Lemma 1), these constraints are equivalent to

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

where $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$, for $k = 1, 2$. Replacing the constraints in (11) leads to the following deterministic problem:

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

where $\Sigma_k = S_k S_k^\top$, for $k = 1, 2$. Formulation (12) is a quadratic SOCP with two blocks (Alizadeh & Goldfarb, 2003). This problem can be formulated as a linear SOCP with three blocks by introducing a new variable t and a constraint $\|\mathbf{w}\| \leq t$. The solutions for both models are similar, but linear SOCP formulations are required by some SOCP solvers, such as the one used in this work. These linear SOCP formulations can be solved efficiently by interior point methods (Alizadeh & Goldfarb, 2003; Alvarez, López, & Ramírez C., 2010).

A kernel-based version can be derived from Formulation (12) by rewriting weight vector $\mathbf{w} \in \Re^n$ as $\mathbf{w} = \mathbb{X}\mathbf{s} + M\mathbf{r}$, where M is a matrix whose columns (as vectors) are orthogonal to training data points; **s, r** are vectors of combining coefficients with the appropriate dimension; and $\mathbb{X} = [A^\top \; B^\top] \in \Re^{n \times m}$ is the data matrix containing both training patterns. On the other hand, the empirical estimates of the mean $\boldsymbol{\mu}_k$ and covariance Σ_k are given by

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

where

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

Thus, for each class k, we have

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

where

$$\mathbf{g}_k = \frac{1}{m_k}\begin{bmatrix} \mathbf{K_{1k}}\mathbf{e_k} \\ \mathbf{K}_{2k}\mathbf{e}_k \end{bmatrix}, \quad \Xi_k = \frac{1}{m_k}\begin{bmatrix} \mathbf{K_{1k}} \\ \mathbf{K}_{2k} \end{bmatrix}\left(I_{m_k} - \frac{1}{m_k}\mathbf{e}_k\mathbf{e}_k^\top\right)\left[\mathbf{K}_{1k}^\top \; \mathbf{K}_{2k}^\top\right],$$

where $\mathbf{K}_{11} = AA^\top$, $\mathbf{K}_{12} = \mathbf{K}_{21}^\top = BA^\top$, $\mathbf{K}_{22} = BB^\top$ are matrices whose elements are inner products of data points. For instance, the entry (l, s) for the matrix $\mathbf{K}_{kk'}$ is the following $(\mathbf{K}_{kk'})_{ls} = (\mathbf{x}_l^k)^\top \mathbf{x}_s^{k'}$. Using a kernel function, this quantity becomes:

$$(\mathbf{K}_{kk'})_{ls} = \mathcal{K}(\mathbf{x}_l^k, \mathbf{x}_s^{k'}).$$

Therefore, the non-linear formulation is given by:

$$\begin{aligned} \min_{\mathbf{s},b} \quad & \frac{1}{2}\mathbf{s}^\top \mathbf{K}\mathbf{s} \\ \text{s.t.} \quad & \mathbf{s}^\top \mathbf{g}_1 + b \geq 1 + \kappa_1 \sqrt{\mathbf{s}^\top \Xi_1 \mathbf{s}} \\ & -\mathbf{s}^\top \mathbf{g}_2 - b \geq 1 + \kappa_2 \sqrt{\mathbf{s}^\top \Xi_2 \mathbf{s}}, \end{aligned} \tag{13}$$

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

3. Robust nonparallel hyperplane SVM (RNH-SVM)

A novel approach for binary classification using second-order cones and nonparallel hyperplanes is presented in this section. This formulation extends the ideas of the NH-SVM approach (Shao et al., 2014) to second-order cones. The reasoning behind this approach is to construct two nonparallel classifiers simultaneously, in such a way that each hyperplane is close to one class and far away from the other class, while each training pattern is represented by ellipsoids instead of reduced convex hulls.

The linear formulation of RNH-SVM is presented in Section 3.1. The dual form of RNH-SVM is derived in Section 3.2, providing the geometrical interpretation of the method. The kernel-based version of RNH-SVM is described in Section 3.3. Finally, the relationship between our approach and other SVM-based methods is discussed in Section 3.4.

3.1. Linear RNH-SVM

In order to obtain two linear nonparallel hyperplanes, we consider the following quadratic chance-constrained programming problem:

$$\begin{aligned} \min_{\substack{\mathbf{w}_k, b_k \\ k=1,2}} \; & \frac{1}{2}\left(\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2\right) \\ & + \frac{\theta}{2}\left(\|\mathbf{w}_1\|^2 + b_1^2 + \|\mathbf{w}_2\|^2 + b_2^2\right) \\ \text{s.t.} \; & \inf_{\mathbf{x}_1 \sim (\boldsymbol{\mu}_1, \Sigma_1)} \Pr\{\mathbf{X}_1 \in H^+(\mathbf{w}_1 - \mathbf{w}_2, b_1 - b_2)\} \geq \eta_1, \\ & \inf_{\mathbf{x}_1 \sim (\boldsymbol{\mu}_1, \Sigma_1)} \Pr\{\mathbf{X}_2 \in H^-(\mathbf{w}_1 - \mathbf{w}_2, b_1 - b_2)\} \geq \eta_2, \end{aligned}$$

where $\theta > 0$, and

$$H^+(\mathbf{w}, b) := \{\mathbf{x} : \mathbf{x}^\top \mathbf{w} + b \geq 1\}, \quad H^-(\mathbf{w}, b) := \{\mathbf{x} : \mathbf{x}^\top \mathbf{w} + b \leq -1\}.$$

Notice that the previous formulation has a similar objective function compared to the NH-SVM formulation when setting $c_2 = 0$ for the latter (hard-margin NH-SVM). The constraints are used to assure that the two hyperplanes, H^- and H^+, classify correctly the instances from both classes up to the rate η_k $(k = 1, 2)$ under a probabilistic scheme. Denoting $\Sigma_1 = S_1 S_1^\top$, $\Sigma_2 = S_2 S_2^\top$, and following the arguments provided in Section 2.4, we obtain the following deterministic problem (RNH-SVM formulation):

$$\begin{aligned} \min_{\substack{\mathbf{w}_k, b_k \\ k=1,2}} \; & \frac{1}{2}\left(\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2\right) \\ & + \frac{\theta}{2}\left(\|\mathbf{w}_1\|^2 + b_1^2 + \|\mathbf{w}_2\|^2 + b_2^2\right) \\ \text{s.t.} \; & (\mathbf{w}_1 - \mathbf{w}_2)^\top \boldsymbol{\mu}_1 + (b_1 - b_2) \geq 1 + \kappa_1 \|S_1^\top (\mathbf{w}_1 - \mathbf{w}_2)\|, \\ & -\left((\mathbf{w}_1 - \mathbf{w}_2)^\top \boldsymbol{\mu}_2 + (b_1 - b_2)\right) \geq 1 + \kappa_2 \|S_2^\top (\mathbf{w}_1 - \mathbf{w}_2)\|, \end{aligned} \tag{14}$$

where $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$ for $k = 1, 2$.

Note that the objective function of Problem (14) can be written compactly as

$$\frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta}{2}(\|\mathbf{w}_1\|^2 + b_1^2) = \frac{1}{2}\mathbf{v}_1^\top (H^\top H + \theta I)\mathbf{v}_1, \tag{15}$$

and

$$\frac{1}{2}\|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 + \frac{\theta}{2}(\|\mathbf{w}_2\|^2 + b_2^2) = \frac{1}{2}\mathbf{v}_2^\top (G^\top G + \theta I)\mathbf{v}_2, \tag{16}$$

where

$$\begin{aligned} & \mathbf{v}_k = [\mathbf{w}_k^\top, b_k]^\top \in \Re^{n+1}, \; H = [A, \mathbf{e}_1] \in \Re^{m_1 \times (n+1)}, \\ & G = [B, \mathbf{e}_2] \in \Re^{m_2 \times (n+1)}. \end{aligned} \tag{17}$$

Then, by introducing new variables t_1, t_2 and the constraints

$$\|(H^\top H + \theta I)^{1/2}\mathbf{v}_1\| \leq t_1, \quad \|(G^\top G + \theta I)^{1/2}\mathbf{v}_2\| \leq t_2,$$

Problem (14) can be cast as the following linear SOCP problem:

$$\begin{aligned}
&\min_{\substack{\mathbf{w}_k,b_k,t_k\\k=1,2}} \; t_1+t_2\\
&\text{s.t. } \left\|(H^\top H+\theta I)^{1/2}\mathbf{v}_1\right\| \le t_1,\\
&\quad \left\|(G^\top G+\theta I)^{1/2}\mathbf{v}_2\right\| \le t_2,\\
&\quad (\mathbf{w}_1-\mathbf{w}_2)^\top\boldsymbol{\mu}_1+(b_1-b_2) \ge 1+\kappa_1\|S_1^\top(\mathbf{w}_1-\mathbf{w}_2)\|,\\
&\quad -\big((\mathbf{w}_1-\mathbf{w}_2)^\top\boldsymbol{\mu}_2+(b_1-b_2)\big) \ge 1+\kappa_2\|S_2^\top(\mathbf{w}_1-\mathbf{w}_2)\|.
\end{aligned} \tag{18}$$

The decision function is similar to the one used for the NH-SVM method, that is, a new sample $\mathbf{x}$ belongs to the class k^* iff $k^* = \text{argmin}_{k=1,2}\{\frac{|\mathbf{w}_k^\top\mathbf{x}+b_k|}{\|\mathbf{w}_k\|}\}$.

3.2. Dual formulation of RNH-SVM and geometric interpretation

In this section we derive the dual formulation of RNH-SVM in its linear version (Formulation (14)), and provide geometrical insights into the method. The Lagrangian function associated with Problem (14) is given by

$$\begin{aligned}
&L(\mathbf{w}_k,b_k,\lambda_k)\\
&=\frac{1}{2}\Big(\|A\mathbf{w}_1+\mathbf{e}_1b_1\|^2+\|B\mathbf{w}_2+\mathbf{e}_2b_2\|^2\Big)+\frac{\theta}{2}\Big(\|\mathbf{w}_1\|^2+b_1^2\Big)\\
&\quad+\frac{\theta}{2}\Big(\|\mathbf{w}_2\|^2+b_2^2\Big)-\lambda_1(\mathbf{w}_1-\mathbf{w}_2)^\top\boldsymbol{\mu}_1+\lambda_2(\mathbf{w}_1-\mathbf{w}_2)^\top\boldsymbol{\mu}_2\\
&\quad+\lambda_1\Big(-(b_1-b_2)+1+\kappa_1\|S_1^\top(\mathbf{w}_1-\mathbf{w}_2)\|\Big)\\
&\quad+\lambda_2\Big((b_1-b_2)+1+\kappa_2\|S_2^\top(\mathbf{w}_1-\mathbf{w}_2)\|\Big),
\end{aligned} \tag{19}$$

where $\lambda_k \ge 0$, for $k=1,2$. Since $\|\mathbf{v}\| = \max_{\|\mathbf{u}\|\le 1}\mathbf{u}^\top\mathbf{v}$ holds for any $\mathbf{v}\in\Re^n$, we can rewrite the Lagrangian as follows:

$$L(\mathbf{w}_k,b_k,\lambda_k) = \max_{\mathbf{u}_1,\mathbf{u}_2}\{\hat{L}(\mathbf{w}_k,b_k,\lambda_k,\mathbf{u}_k) : \|\mathbf{u}_k\|\le 1\},$$

where $\hat{L}$ is given by

$$\begin{aligned}
&\hat{L}(\mathbf{w}_k,b_k,\lambda_k,\mathbf{u}_k)\\
&=\frac{1}{2}\Big(\|A\mathbf{w}_1+\mathbf{e}_1b_1\|^2+\|B\mathbf{w}_2+\mathbf{e}_2b_2\|^2\Big)+\frac{\theta}{2}\Big(\|\mathbf{w}_1\|^2+b_1^2\Big)\\
&\quad+\frac{\theta}{2}\Big(\|\mathbf{w}_2\|^2+b_2^2\Big)-\lambda_1(\mathbf{w}_1-\mathbf{w}_2)^\top\boldsymbol{\mu}_1+\lambda_2(\mathbf{w}_1-\mathbf{w}_2)^\top\boldsymbol{\mu}_2\\
&\quad+\lambda_1\Big(-(b_1-b_2)+1+\kappa_1(\mathbf{w}_1-\mathbf{w}_2)^\top S_1\mathbf{u}_1\Big)\\
&\quad+\lambda_2\Big((b_1-b_2)+1+\kappa_2(\mathbf{w}_1-\mathbf{w}_2)^\top S_2\mathbf{u}_2\Big).
\end{aligned} \tag{20}$$

Thus, Problem (14) can be written equivalently as

$$\min_{\mathbf{w}_k,b_k}\max_{\mathbf{u}_k,\lambda_k}\{\hat{L}(\mathbf{w}_k,b_k,\lambda_k,\mathbf{u}_k) : \|\mathbf{u}_k\|\le 1,\lambda_k\ge 0\},$$

and therefore the Wolfe-dual of Formulation (14) (see, e.g. Mangasarian (1994)) corresponds to

$$\max_{\mathbf{u}_k,\lambda_k}\{\hat{L} : \nabla_{\mathbf{w}_k}\hat{L}=0,\nabla_{b_k}\hat{L}=0,\|\mathbf{u}_k\|\le 1,\lambda_k\ge 0,k=1,2\}. \tag{21}$$

Computing the gradient of $\hat{L}$ with respect to $\mathbf{w}_k$ and b_k $(k=1,2)$ leads to the following linear system

$$(A^\top A+\theta I)\mathbf{w}_1+b_1A^\top\mathbf{e}_1=\lambda_1\mathbf{z}_1-\lambda_2\mathbf{z}_2, \tag{22}$$

$$(B^\top B+\theta I)\mathbf{w}_2+b_2B^\top\mathbf{e}_2=-\lambda_1\mathbf{z}_1+\lambda_2\mathbf{z}_2, \tag{23}$$

$$\theta b_1+(\mathbf{w}_1^\top A^\top\mathbf{e}_1+\mathbf{e}_1^\top\mathbf{e}_1b_1)=\lambda_1-\lambda_2, \tag{24}$$

$$\theta b_2+(\mathbf{w}_2^\top B^\top\mathbf{e}_2+\mathbf{e}_2^\top\mathbf{e}_2b_2)=-\lambda_1+\lambda_2. \tag{25}$$

where $\mathbf{z}_1=\mu_1-\kappa_1S_1\mathbf{u}_1$ and $\mathbf{z}_2=\mu_2+\kappa_2S_2\mathbf{u}_2$.

The Relations (22)–(25) can be written compactly as

$$(H^\top H+\theta I)\mathbf{v}_1=Z\boldsymbol{\lambda},\quad (G^\top G+\theta I)\mathbf{v}_2=-Z\boldsymbol{\lambda}, \tag{26}$$

where $\boldsymbol{\lambda}=[\lambda_1,\lambda_2]\in\Re^2$, $Z=[\mathbf{z}_1,-\mathbf{z}_2;1,-1]\in\Re^{n+1\times 2}$. The operator ', ' in $[a,b]$ concatenates matrices a and b horizontally, while the operator '; ' in $[a;b]$ concatenates both matrices vertically. Since the symmetric matrices $(H^\top H+\theta I)$ and $(G^\top G+\theta I)$ are positive definite, for any $\theta>0$, one has that

$$\mathbf{v}_1=(H^\top H+\theta I)^{-1}Z\boldsymbol{\lambda},\quad \mathbf{v}_2=-(G^\top G+\theta I)^{-1}Z\boldsymbol{\lambda}. \tag{27}$$

Subsequently, the objective function $\hat{L}$ in (21) can be rewritten using (15), (16), and (26), as follows:

$$\hat{L}=-\frac{1}{2}\Big[\mathbf{v}_1^\top(H^\top H+\theta I)\mathbf{v}_1+\mathbf{v}_2^\top(G^\top G+\theta I)\mathbf{v}_2\Big]+\mathbf{e}^\top\boldsymbol{\lambda},$$

where $\mathbf{e}^\top=[1,1]\in\Re^2$.

Finally, the dual problem can be rewritten using (27), as follows:

$$\begin{aligned}
&\min_{\substack{\mathbf{z}_k,\mathbf{u}_k,\\ \lambda_k,k=1,2}} \frac{1}{2}\boldsymbol{\lambda}^\top Z^\top[(H^\top H+\theta I)^{-1}+(G^\top G+\theta I)^{-1}]Z\boldsymbol{\lambda}-\mathbf{e}^\top\boldsymbol{\lambda}\\
&\text{s.t. } \mathbf{z}_1=\mu_1-\kappa_1S_1\mathbf{u}_1,\ \|\mathbf{u}_1\|\le 1,\\
&\quad \mathbf{z}_2=\mu_2+\kappa_2S_2\mathbf{u}_2,\ \|\mathbf{u}_2\|\le 1,\\
&\quad \lambda_1\ge 0,\lambda_2\ge 0.
\end{aligned} \tag{28}$$

The optimal value for $\boldsymbol{\lambda}$ can be obtained by fixing variables $\mathbf{z}_k$ and $\mathbf{u}_k$ $(k=1,2)$, and solving the following linear system:

$$Z^\top[(H^\top H+\theta I)^{-1}+(G^\top G+\theta I)^{-1}]Z\boldsymbol{\lambda}=\mathbf{e}.$$

The solution of this system of linear equations allows us to rewrite the dual problem (28) as

$$\begin{aligned}
&\min_{\substack{\mathbf{z}_k,\mathbf{u}_k\\k=1,2}} -\frac{1}{2}\mathbf{e}^\top\Big(Z^\top[(H^\top H+\theta I)^{-1}+(G^\top G+\theta I)^{-1}]Z\Big)^{-1}\mathbf{e}\\
&\text{s.t. } \mathbf{z}_1\in\mathbf{B}(\boldsymbol{\mu}_1,S_1,-\kappa_1),\\
&\quad \mathbf{z}_2\in\mathbf{B}(\boldsymbol{\mu}_2,S_2,\kappa_2),
\end{aligned} \tag{29}$$

where

$$\mathbf{B}(\boldsymbol{\mu},S,\kappa)=\{\mathbf{z}:\mathbf{z}=\boldsymbol{\mu}+\kappa S\mathbf{u},\|\mathbf{u}\|\le 1\}. \tag{30}$$

The set $\mathbf{B}(\boldsymbol{\mu},S,\kappa)$ denotes an ellipsoid centered at $\boldsymbol{\mu}$ whose shape is determined by S. This result is important since we can link the proposed formulation to the geometrical interpretation: the ellipsoids $\mathbf{B}(\boldsymbol{\mu},S,\kappa)$ define the two hyperplanes, and subsequently the classification rule. Fig. 2 illustrates the geometrical interpretation of the proposed approach in its linear version.

The parameter $\kappa_k=\sqrt{\frac{\eta_k}{1-\eta_k}}$ $(k=1,2)$ governs the size of the ellipsoids (Lanckriet et al., 2003). Fig. 3 presents the influence of parameter η in the solution of RNH-SVM for a toy example.

In Fig. 3 we observe that η controls the size of the ellipsoid of the respective class: higher values of η imply bigger ellipsoids. Uneven η values allow the method to manage the classification performance for each class, favoring the one with higher error costs, for example. The SOCP formulations provide an adequate framework for dealing with the class-imbalance problem (Maldonado & López, 2014b).

Finally, the following remark relates the primal and dual variables of the RNH-SVM formulation, and is relevant since we can solve the dual formulations and then obtain both nonparallel hyperplanes. The weights $\mathbf{w}_k$ provide interesting insight into the solution found, since we can assess the importance of each attribute in the final solution (Bai et al., 2014; Maldonado, Famili, & Weber, 2014).

Remark 3.1. Note that if $\mathbf{z}_k^*\in\Re^n$, for $k=1,2$, are the optimal solutions of Problem (29), then by using (27), the solution

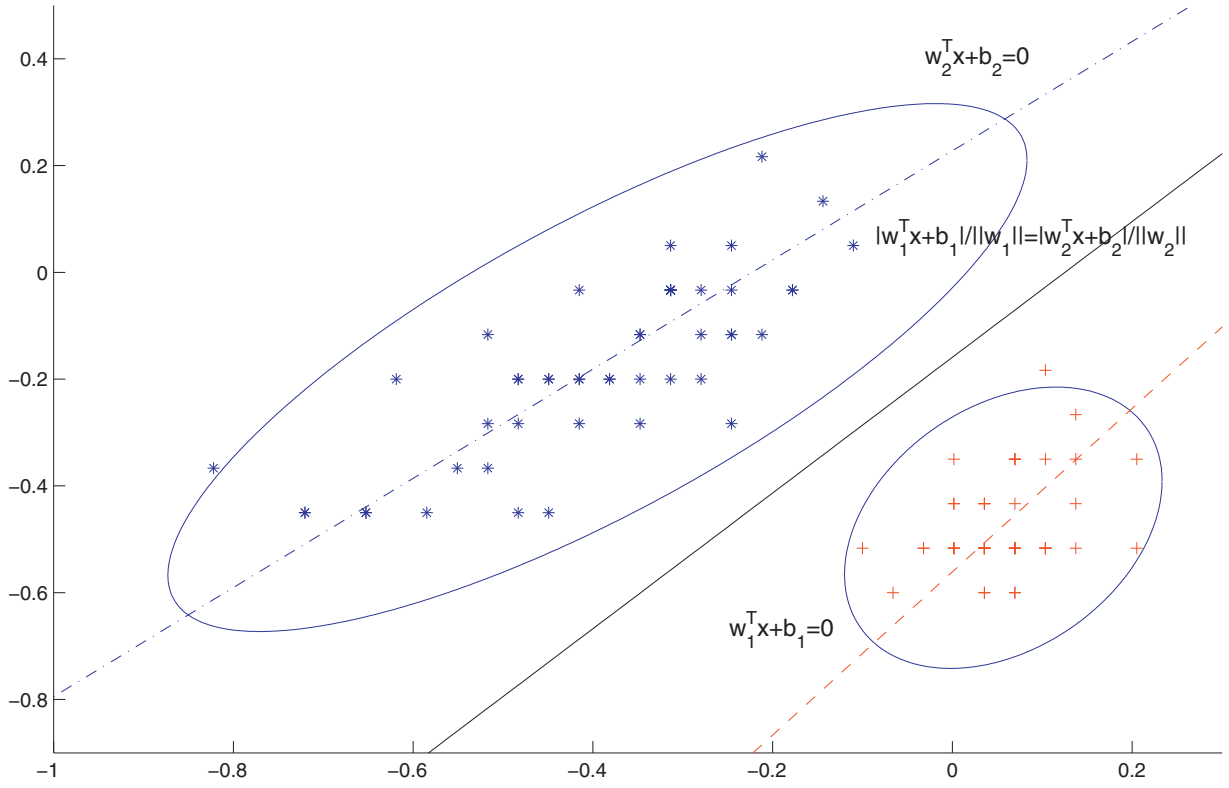

Fig. 2. Geometric interpretation for Robust Nonparallel Hyperplane SVM.

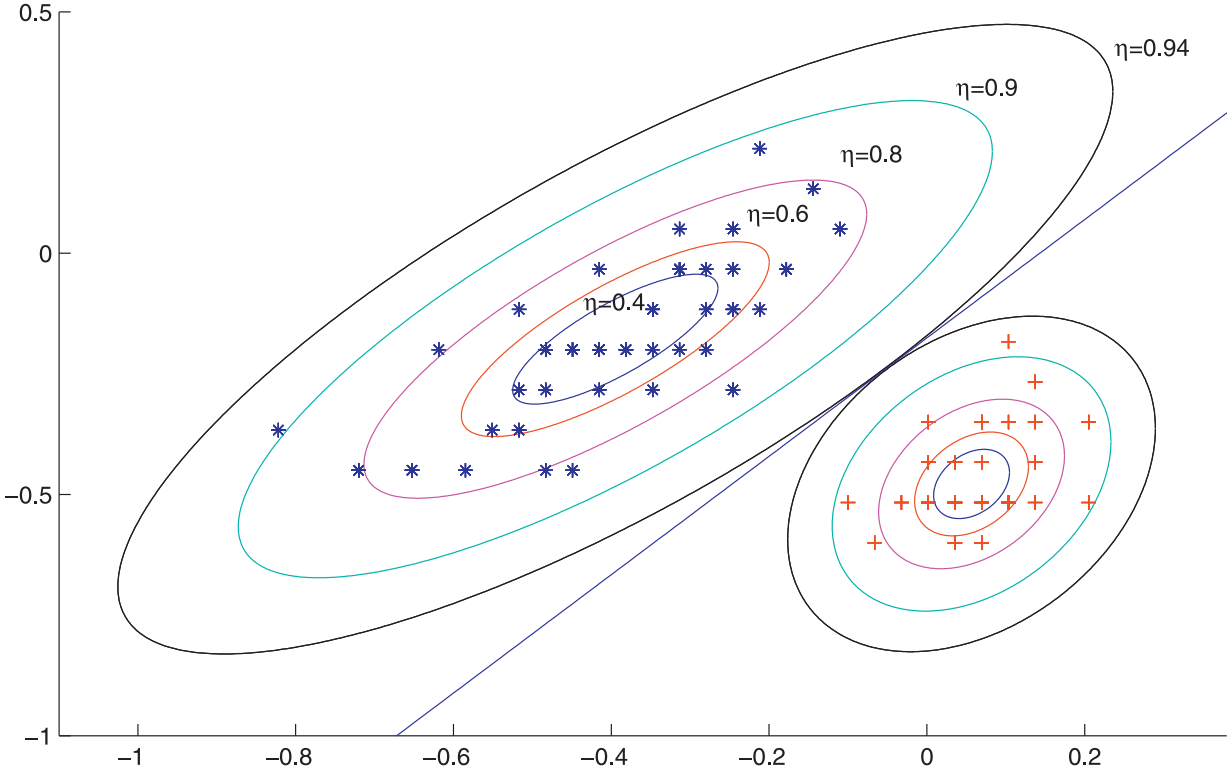

Fig. 3. Geometric interpretation for RNH-SVM, influence of parameter η.

$\mathbf{v}_k^* = [\mathbf{w}_k^{*\top}, b_k^*]^\top$ $(k = 1, 2)$ of Problem (14) can be compute by

$$\mathbf{v}_1^* = (H^\top H + \theta I)^{-1} Z (Z^\top [(H^\top H + \theta I)^{-1} + (G^\top G + \theta I)^{-1}] Z)^{-1} \mathbf{e}, \tag{31}$$

and

$$\mathbf{v}_2^* = (G^\top G + \theta I)^{-1} Z (Z^\top [(H^\top H + \theta I)^{-1} + (G^\top G + \theta I)^{-1}] Z)^{-1} \mathbf{e}, \tag{32}$$

with $Z = [\mathbf{z}_1^*, -\mathbf{z}_2^*; 1, -1]$.

3.3. Kernel-based RNH-SVM formulation

In this section we extend RNH-SVM to kernel functions to obtain non-linear classifiers. Following the notation introduced in Section 2.4, the weight vectors for each one of the nonparallel hyperplanes can be written as $\mathbf{w}_k = \mathbb{X}\mathbf{s}_k + M\mathbf{r}_k$, where M and $\mathbb{X}$ are equivalent to the matrices described in Section 2.4, and $\mathbf{s}_k$ and $\mathbf{r}_k$ are combining coefficients with the appropriate dimension. For each problem we have

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

and

$$A\mathbf{w}_1 = [\mathbf{K}_{11}\ \mathbf{K}_{12}]\mathbf{s}_1 = \mathbf{K}_{1\bullet}\mathbf{s}_1, \quad B\mathbf{w}_2 = [\mathbf{K}_{21}\ \mathbf{K}_{22}]\mathbf{s}_2 = \mathbf{K}_{2\bullet}\mathbf{s}_2,$$

where $\mathbf{g}_k$, Ξ_k, and $\mathbf{K}_{kk'}$ have a similar form compared to the notation presented in Section 2.4. Hence, in order to obtain a kernel formulation for Problem (14), we replace the inner product that appears in the expressions $\mathbf{K}_{kk'}$ by any kernel function $\mathcal{K}: \Re^n \times \Re^n \to \Re$ satisfying Mercer's condition (see Mercer (1909)), obtaining the following model (kernel-based RNH-SVM):

$$\begin{aligned}
\min_{\substack{\mathbf{s}_k, b_k, \\ k=1,2}} \ & \frac{1}{2}(\|\mathbf{K}_{1\bullet}\mathbf{s}_1 + \mathbf{e}_1 b_1\|^2 + \|\mathbf{K}_{2\bullet}\mathbf{s}_2 + \mathbf{e}_2 b_2\|^2) \\
& + \frac{\theta}{2}(\|\mathbf{s}_1\|^2 + b_1^2 + \|\mathbf{s}_2\|^2 + b_2^2) \\
\text{s.t. } & (\mathbf{s}_1 - \mathbf{s}_2)^\top \mathbf{g}_1 + (b_1 - b_2) \geq 1 + \kappa_1 \|\Lambda_1^\top (\mathbf{s}_1 - \mathbf{s}_2)\|, \\
& -(\mathbf{s}_1 - \mathbf{s}_2)^\top \mathbf{g}_1 - (b_1 - b_2) \geq 1 + \kappa_2 \|\Lambda_2^\top (\mathbf{s}_1 - \mathbf{s}_2)\|,
\end{aligned} \tag{33}$$

where $\Xi_k = \Lambda_k \Lambda_k^\top$, for $k = 1, 2$. Then, the solution of Problem (33) generates the following kernel-based surfaces:

$$\mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{s}_1 + b_1 = 0, \quad \mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{s}_2 + b_2 = 0, \tag{34}$$

where, for a given $\mathbf{x} \in \mathbb{R}^n$, the row vector $\mathcal{K}(\mathbf{x}, \mathbb{X})$ is defined by

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

with $\mathbb{X}_{\bullet j} \in \Re^n$ denoting the jth column of the matrix $\mathbb{X}$. According to this, a new point $\mathbf{x} \in \Re^n$ belongs to the class k^* iff

$$k^* = \underset{k=1,2}{\operatorname{argmin}} \frac{|\mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{s}_k + b_k|}{\sqrt{\mathbf{s}_k^\top \mathbf{K} \mathbf{s}_k}}, \tag{35}$$

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

3.4. Relation to other SVM methods

Our proposal extends the ideas of NH-SVM (Shao et al., 2014) to second-order cones, following the methodology suggested by Nath and Bhattacharyya (2007). These authors proposed a maximum-margin separating hyperplane to split the training patterns, characterized by ellipsoids. On the other hand, NH-SVM extends the ideas of Twin SVM (Jayadeva et al., 2007), in which two nonparallel hyperplanes are constructed to perform binary classification. In contrast to Twin SVM, where the hyperplanes are constructed independently via two different optimization problems, NH-SVM estimates both classification functions jointly, taking all available information into account in a single formulation.

Afew approaches are closely related to our proposal. The only approach, to the best of our knowledge, that relates Twin SVM and second-order cone programming is the work proposed by Qi, Tian, and Shi (2013). In that work, the SOCP-based SVM formulation by Goldfarb and Iyengar (2003) is extended to Twin SVM. The main difference between this method and the SOCP-SVM approach developed by Nath and Bhattacharyya (2007) is that the former uses robust constraints to deal with noisy data (instances with measurement errors for example), while the latter provides a probabilistic framework for the class-conditional densities. The Goldfarb and Iyengar formulation results in m linear constraints and one second-order cone constraint, while the Nath and Bhattacharyya method considers one second-order cone constraint for each training pattern k, which is computationally more tractable. Similar to Goldfarb and Iyengar, Zhong and Fukushima (Zhong & Fukushima, 2007) proposed a multi-class approach based on SOCP to deal with noisy observations.

Several extensions have been proposed for both the Twin SVM and the SOCP-SVM approach by Nath & Bhattacharyya (2007). An efficient optimization scheme together with other modifications to the original version was proposed by Shao et al. (2011) for Twin SVM. On the other hand, the Nath and Bhattacharyya formulation has been extended further for dealing with class-imbalance (Maldonado & López, 2014b) and high dimensionality (Maldonado & López, 2015).

4. Experimental results

We applied the proposed approach in its linear and kernel-based forms to seven well-known data sets from the UCI Repository (Bache & Lichman, 2013): Australian Credit (AUS), Wisconsin Breast Cancer (WBC), BUPA Liver (LIVER), German Credit (GER), Pima Indians Diabetes (DIA), Heart/Statlog (HEART), and Ionosphere (IONO). All variables in the data sets were scaled between −1 and 1. Table 1 summarizes the relevant information for each benchmark data set, including the number of variables, the sample size, the percentage of observations in each class, and the imbalance ratio (IR).

Together with our proposals, namely the Robust Nonparallel Hyperplane SVM method in its linear (RNH-SVM$_l$, Formulation (14)), and the kernel-based version (RNH-SVM$_K$, Formulation (33)), the following alternative approaches have been studied and reported for benchmarking purposes:

- Standard SVM, linear (SVM$_l$, Formulation (1)) and kernel-based version (SVM$_K$, Formulation (2)).
- Twin-Bounded SVM, linear (TB-SVM$_l$, Formulation (4)–(5)) and kernel-based version (TB-SVM$_K$, Formulation (7)–(8)).
- Nonparallel hyperplane SVM, linear (NH-SVM$_l$, Formulation (9)) and kernel-based version (NH-SVM$_K$, Formulation (10)).
- SOCP-SVM, linear (SOCP-SVM$_l$, Formulation (12)) and kernel-based version (SOCP-SVM$_K$, Formulation (13)).

The validation and model selection procedure consisted of a grid search for SVM parameters C and σ; Twin SVM parameter c_i, $i = \{1, 2, 3, 4\}$; SOCP parameters η_k; and parameter θ used in the proposed approach. We studied the following values of $\eta_k \in \{0.2, 0.4, 0.6, 0.8\}$. We used the following set of values: C, c_i, θ, $\sigma \in \{2^{-7}, 2^{-6}, \ldots, 2^6, 2^7\}$. Training and test sets were obtained using 10-fold cross-validation, while the metric Area Under the Curve (AUC) was used as the main performance measure. We used LIBSVM for Matlab (Chang & Lin, 2011) for standard SVM approaches, the SeDuMi Matlab Toolbox for SOCP-based classifiers (Sturm, 1999), and the codes provided by Yuan-Hai Shao, author of NH-SVM and Twin-Bounded SVM, which are publicly available in http://www.optimal-group.org/.

Table 1
The metadata for all data sets.

Data set	#features	#examples	%class(min.,maj.)	IR
AUS	14	690	(55.5,44.5)	1.2
WBC	30	569	(62.7,37.3)	1.7
LIVER	6	345	(58.0,42.0)	1.4
GER	24	1000	(70.0,30.0)	2.3
DIA	8	768	(65.1,34.9)	1.9
HEART	13	270	(55.6,44.4)	1.25
IONO	34	351	(64.1,35.9)	1.8

Table 2
Predictive performance summary for all linear approaches and for all data sets.

	AUS	WBC	LIVER	GER	DIA	HEART	IONO
SVM_l	86.2	97.3	51.5	69.4	72.1	50.8	**93.2**
TB-SVM_l	86.7	96.8	65.9	72.2	73.4	85.0	85.2
NH-SVM_l	**87.0**	95.8	65.4	68.0	72.2	**85.0**	80.5
SOCP-SVM_l	86.8	96.5	63.9	72.2	74.9	84.7	86.1
RNH-SVM_l	86.8	**97.9**	**67.6**	**73.2**	**75.9**	84.8	83.5

Table 3
Predictive performance summary for all kernel-based approaches and for all data sets.

	AUS	WBC	LIVER	GER	DIA	HEART	IONO
SVM_K	86.2	97.1	**73.3**	68.8	72.1	79.4	94.1
TB-SVM_K	87.6	97.0	65.0	72.4	75.6	62.3	**95.4**
NH-SVM_K	87.1	97.1	67.1	68.9	73.7	64.4	**95.4**
SOCP-SVM_K	86.9	97.4	72.9	72.2	76.3	79.5	95.2
RNH-SVM_K	**87.9**	**98.1**	73.2	**72.6**	**76.4**	**80.9**	**95.4**

Tables 2 and 3 present the best performance of all methods obtained by the validation procedure described above. Table 2 provides the results for the linear approaches, while Table 3 presents the results of the kernel-based methods. For each family of methods (linear or non-linear) and each data set, the best technique in terms of AUC is highlighted in bold type.

In Table 2 we observe that the best predictive results were achieved using the proposed method RNH-SVM_l in four out of seven data sets, while NH-SVM had better AUC in two data sets (Australian Credit and Heart/Statlog) and standard SVM in one data set (Ionosphere). The TB-SVM_l, NH-SVM_l, and SOCP-SVM_l approaches, and the proposed RNH-SVM_l have relatively similar performance among all data sets, outperforming SVM_l in BUPA Liver and Heart/Statlog. Similar results can be observed in Table 3, where the proposed method performs better in six out of seven data sets.

Although no method outperformed others in all the experiments, and the differences in terms of AUC are not conclusive in most cases, our proposal achieved the best overall performance for all methods. We used the robustness analysis procedure proposed by Geng, Zhan, and Zhou (2005) to assess this claim quantitatively. We first computed the *relative performance* of a given method M on a data set i as the ratio between its AUC and the highest among all the compared strategies:

$$AUCRatio_i(M) = \frac{AUC(M)}{max_j\ AUC(j)}, \tag{36}$$

where $AUC(j)$ is the AUC for method j when trained over data set i. The larger the value of $AUCRatio_i(M)$, the better the performance of M in data set i. The best approach M^* will have $AUCRatio_i(M^*)$ equal to 1 for data set i. The value of $\Sigma_i AUCRatio_i(M)$ represents a measure of robustness and overall performance for a method M, and the larger its value, the better its overall performance and robustness (Geng et al., 2005).

Figs. 4 and 5 present the distribution of the relative performance for the five methods and all data sets. Fig. 4 shows all linear methods, while Fig. 5 presents all kernel-based approaches. Each technique is represented by a stacked bar that aggregates the relative performances for all data sets.

In Figs. 4 and 5 we observe that RNH-SVM has the best overall performance for both linear and kernel-based approaches, being close to the optimal performance measure of 7 (6.89) for the former, and achieving optimal performance for the latter. For linear methods, standard SVM has the lowest overall performance, while TB-SVM and NH-SVM have the lowest relative performance for kernel methods.

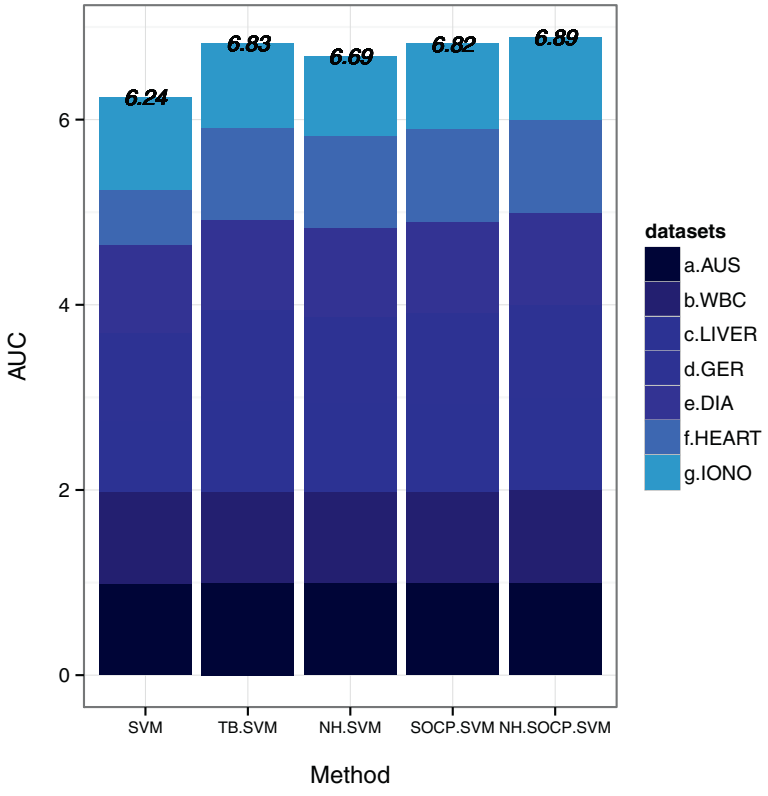

Fig. 4. Sum of AUC ratios for all linear methods.

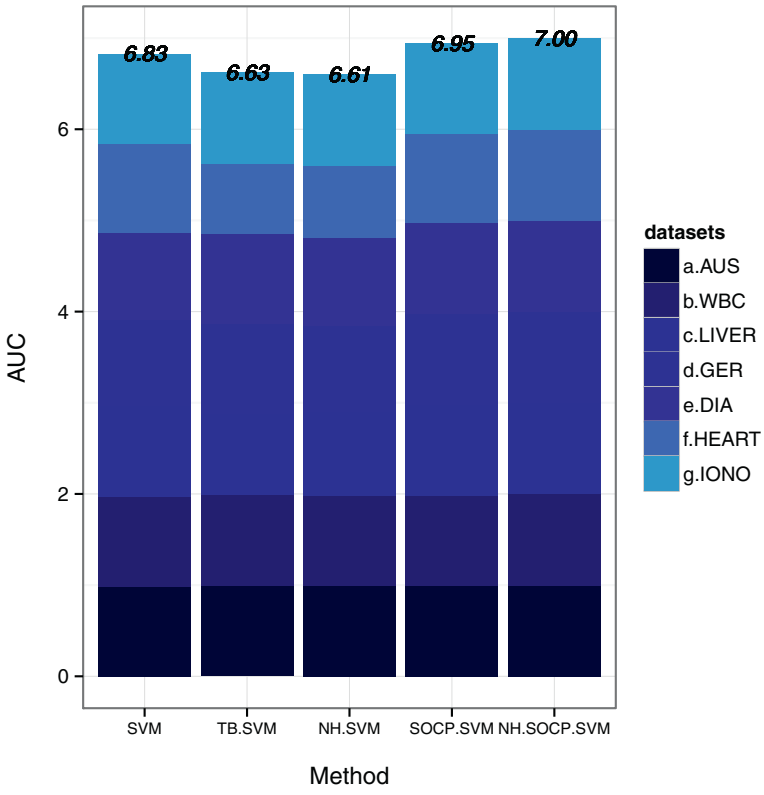

Fig. 5. Sum of AUC ratios for all kernel-based methods.

We conclude that RNH-SVM is an excellent alternative for binary classification, since it is based on a robust framework that constructs two nonparallel hyperplanes in a single optimization problem. Both robust optimization via SOCP and the Twin SVM formulation have been proved to be effective at enhancing SVM's predictive performance, and our results demonstrate that the combination of both strategies leads to a formulation that achieves the best overall performance among all the methods studied.

5. Discussion and conclusions

In this work, a novel SVM-based classification approach is presented. Its main contribution is the use of second-order cones to model each training pattern, conferring robustness to the NH-SVM formulation (Shao et al., 2014). The method constructs two nonparallel hyperplanes using kernel functions and solving a single optimization problem. Empirically, we observed the best average performance on seven benchmark data sets with our proposal. The gain is particularly important compared to standard SVM for two data sets, BUPA Liver and Heart/Statlog, since the latter method failed at constructing a classifier that correctly split the two classes, leading to an AUC close to 0.5.

An important managerial advantage of the proposed method is its superior performance. Our proposal achieved important gains, in some cases up to a 34% increase in terms of AUC compared to standard SVM, and the proposed kernel-based RNH-SVM was either the best method or only 0.1% below the best technique on one occasion. This method was also 1.93%, 4.17%, and 4.4% better on average with respect to kernel-based SVM, Twin SVM, and NH-SVM, respectively.

In expert systems applications, a 1% increase in predictive performance could be enough to achieve significant monetary gains. For example, it has been suggested that an increase of only 1% in forecast error for the electricity demand in United Kingdom caused an increase of 10 million £in the operating cost per year (Gross & Galiana, 1987). In expert systems like those used for cancer diagnosis, the benefits of a good classifier can be measured in terms of human lives since early cancer detection is the main form of fighting it successfully (Borges, Corrêa, Cardoso, & Gattass, 2015).

Another important managerial insight can be linked to the balanced structure of the proposed method. In contrast to SVM, our proposal assures the correct classification of each training pattern by constraining the misclassification errors. Since the parameter η manages the Type I and Type II errors, the RNH-SVM method has an enormous potential when facing highly imbalanced data sets, a common issue in intelligent systems such as the one used in churn prediction (Ali & Aritürk, 2014; Saradhi & Palshikar, 2011) or medical diagnosis (Borges et al., 2015; Ríos & Erazo, 2016). A differentiated value for η should suffice for constructing classification functions that include the asymmetric misclassification costs.

The main limitation of the proposed framework is the higher computational effort it requires compared to standard SVM. Several optimization strategies have been tailored for SVM in order to make the classification method more efficient. Some examples of incremental optimization methods are Sequential Minimal Optimization (SMO) for standard SVM (Platt, 1999) and QPSOR for Twin SVM (Shao et al., 2011). However, no optimization technique has been customized for SOCP-based SVM, to the best of our knowledge. We therefore rely exclusively on a general purpose solver for convex SOCP, such as SeDuMI. Future developments could include the design of more efficient optimization strategies for SOCP-SVM.

Future work can be performed in several directions. First, it would interesting to apply the proposed method in class-imbalance problems related to intelligent systems. As mentioned above, this issue occurs in several domains related to expert systems, and we believe that the proposal has strong potential for compensating for the undesirable effects caused by unbalanced data sets. Secondly, it would be interesting to extend the proposed

method to multi-class classification, which also has wide applicability in expert systems. Some relevant applications are credit scoring, in which defaulters can be labeled differently according to their willingness to pay (Bravo, Thomas, & Weber, 2014), and the prediction of the occurrence of different types of cancer (Yang, Cai, Li, & Lin, 2006). Thirdly, the development of more efficient implementations than SeDuMI Matlab toolbox for SOCP models is an interesting future research suggestion, since the fast training of an SVM algorithm is an important virtue for expert systems (Czarnecki & Tabor, 2014). Finally, there are several margin maximization strategies for SVM, besides SOCP, that can be explored in intelligent systems in more applied contexts, such as flexible and affine convex hulls (Zeng, Yang, Zheng, & Cheng, 2015), and other strategies based on ellipsoids (Czarnecki & Tabor, 2014).

Acknowledgments

The first author was supported by FONDECYT project 1130905, the second one was funded by FONDECYT project 1160894, and third author was supported by FONDECYT project 1140831. The work reported in this paper has been partially funded by Millennium Scientific Institute on Complex Engineering Systems Institute (ICM: P-05-004-F, CONICYT: FB016).

References

Ali, O. G., & Aritürk, U. (2014). Dynamic churn prediction framework with more effective use of rare event data: the case of private banking. *Expert Systems with Applications, 41*(17), 7889–7903.

Alizadeh, F., & Goldfarb, D. (2003). Second-order cone programming. *Mathematical Programming, 95*, 3–51.

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

Bache, K., & Lichman, M. (2013). UCI machine learning repository. Url: http://archive.ics.uci.edu/ml.

Bai, L., Wang, Z., Shao, Y.-H., & Deng, N.-Y. (2014). A novel feature selection method for twin support vector machine. *Knowledge-Based Systems, 59*(0), 1–8.

Borges, W., Corrêa, A., Cardoso, A., & Gattass, M. (2015). Detection of masses in mammograms with adaption to breast density using genetic algorithm, phylogenetic trees, lbp and svm. *Expert Systems with Applications, 42*(22), 8911–8928.

Bravo, C., Thomas, L., & Weber, R. (2014). Improving credit scoring by differentiating defaulter behaviour. *Journal of the Operational Research Society, 66*, 771–781.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*, 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

Czarnecki, W., & Tabor, J. (2014). Two ellipsoid support vector machines. *Expert Systems with Applications, 41*, 8211–8224.

Debnath, R., Muramatsu, M., & Takahashi, H. (2005). An efficient support vector machine learning method with second-order cone programming for large-scale problems. *Applied Intelligence, 23*, 219–239.

Geng, X., Zhan, D.-C., & Zhou, Z.-H. (2005). Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35*(6), 1098–1107.

Goldfarb, D., & Iyengar, G. (2003). Robust convex quadratically constrained programs. *Mathematical Programming, 97*(3), 495–515.

Gross, G., & Galiana, F. D. (1987). Short term load forecasting. *Proceedings of the IEEE, 75*, 1558–1573.

Jayadeva, Khemchandani, R., & Chandra, S. (2007). Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(5), 905–910.

Kumar, M., & Gopal, M. (2009). Least squares twin support vector machines for pattern classification. *Expert Systems with Applications, 36*, 7535–7543.

Lanckriet, G., Ghaoui, L., Bhattacharyya, C., & Jordan, M. (2003). A robust minimax approach to classification. *Journal of Machine Learning Research, 3*, 555–582.

Maldonado, S., Famili, F., & Weber, R. (2014). Feature selection for high-dimensional class-imbalanced data sets using support vector machines. *Information Sciences, 286*, 228–246.

Maldonado, S., & López, J. (2014a). Alternative second-order cone programming formulations for support vector classification. *Information Sciences, 268*, 328–341.

Maldonado, S., & López, J. (2014b). Imbalanced data classification using second-order cone programming support vector machines. *Pattern Recognition, 47*, 2070–2079.

Maldonado, S., & López, J. (2015). An embedded feature selection approach for support vector classification via second-order cone programming. *Intelligent Data Analysis, 19*(6), 1233–1257.

Maldonado, S., Weber, R., & Basak, J. (2011). Kernel-penalized SVM for feature selection. *Information Sciences, 181*(1), 115–128.

Mangasarian, O. L. (1994). Nonlinear Programming Olvi L. Mangasarian, *Classics in Applied Mathematics* p. 236. Society for Industrial and Applied Mathematics. doi:10.1137/1013044.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London, 209*, 415–446.

Nath, S., & Bhattacharyya, C. (2007). Maximum margin classifiers with specified false positive and false negative error rates. In *Proceedings of the siam international conference on data mining*.

Peng, X. (2010). Least squares twin support vector hypersphere (LS-TSVH) for pattern recognition. *Expert Systems with Applications, 37*(12), 8371–8378.

Platt, J. (1999). *Advances in kernel methods-support vector learning* (pp. 185–208). MIT Press, Cambridge, MA.

Qi, Z., Tian, Y., & Shi, Y. (2013). Robust twin support vector machine for pattern classification. *Pattern Recognition, 46*(1), 305–316.

Ríos, S., & Erazo, L. (2016). An automatic apnea screening algorithm for children. *Expert Systems with Applications, 48*, 42–54.

Saradhi, V., & Palshikar, G. (2011). Employee churn prediction. *Expert Systems with Applications, 38*(3), 1999–2006.

Schölkopf, B., & Smola. , A. J. (2002). *Learning with Kernels*. MIT Press.

Shao, Y., Chen, W., & Deng, N. (2014). Nonparallel hyperplane support vector machine for binary classification problems. *Information Sciences, 263*(0), 22–35.

Shao, Y., Zhang, C., Wang, X., & Deng, N. (2011). Improvements on twin support vector machines. *IEEE Transactions on Neural Networks, 22*(6), 962–968.

Sturm, J. (1999). Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software, 11*(12), 625–653. Special issue on Interior Point Methods (CD supplement with software).

Xu, Y., & Wang, L. (2012). A weighted twin support vector regression. *Knowledge-Based Systems, 33*, 92–101.

Yang, K., Cai, Z., Li, J., & Lin, G. (2006). A stable gene selection in microarray data analysis. *BMC Bioinformatics, 7*, 228.

Zeng, M., Yang, Y., Zheng, J., & Cheng, J. (2015). Maximum margin classification based on flexible convex hulls. *Neurocomputing, 149*(B), 957–965.

Zhong, P., & Fukushima, M. (2007). Second-order cone programming formulations for robust multiclass classification. *Neural Computation, 19*, 258–282.