Knowledge-Based Systems 152 (2018) 83–93

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Robust twin support vector regression via second-order cone programming

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

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

ARTICLE INFO

Article history:
Received 9 September 2017
Revised 1 April 2018
Accepted 2 April 2018
Available online 3 April 2018

Keywords:
Support vector regression
Twin support vector machines
Second-order cone programming
Robust optimization

ABSTRACT

Twin Support Vector Regression is an effective machine learning strategy, which splits the predictive task into two small problems, gaining in both efficiency and predictive performance. In this paper, a novel extension for twin Support Vector Regression is presented. The proposal is based on robust optimization, conferring robustness to the predictive task by dealing effectively with uncertainty. The method is first developed as a linear one, and then, subsequently extended to a kernel-based formulation. Our approach accomplishes the best performance on benchmark datasets compared to alternative methods, such as linear regression, support vector regression, and twin support vector regression. This gain in performance demonstrates the virtues of robust optimization on reducing the risk of overfitting, and generalizing the training patterns well with reduced complexity.

1. Introduction

Twin Support Vector Machine (SVM) is a powerful tool in pattern analysis, given its appealing properties [20,30]. It splits the SVM problem into two smaller sub-problems, constructing two hyperplanes independently. This strategy allows efficient training in terms of running times, thanks to the option of solving the two problems in parallel [26,41]. The twin SVM method has also proved to be more effective in terms of predictive performance than the traditional SVM formulation [20,26,41].

Originally developed by Jayadeva et al. [20] for binary classification, twin SVM has been extended to regression by Peng [30]. The main idea is to construct two non-parallel hyperplanes in order to define an ε-insensitive tube around the data points, where errors are discarded inside the tube and penalized outside of it. Several variants of twin SVR have also been proposed in the literature (see [5,10,21,29,31,36,37,39]), resulting in a fruitful field of research.

Recently, second-order cone programming (SOCP) has been used as a robust optimization scheme for SVM classification. The reasoning behind this approach is to classify all instances correctly for specified class recalls, even for the worst possible class distribution for given means and covariance matrices [34]. This framework was further extended to twin classification in [26].

1.1. Problem Statement and Contribution

In artificial intelligence, robustness characterizes how effective a predictive method is when being applied on new data. The performance of a robust algorithm does not deteriorate much when it is constructed and tested with slightly different data samples, reducing the risk of overfitting [26].

In our proposal, we provide robustness to Support Vector Regression, seizing the virtues of the twin SVR formulation. We propose two twin chance-constrained problems to construct an ε-insensitive tube, for which the up-bound and down-bound functions are obtained in a robust setting. This strategy corresponds to replace the chance constraints of the two twin problems with their robust counterparts, assuming a worst-case setting for the data distribution. This strategy leads to two SOCP problems that can be solved efficiently via interior point algorithms [1,2,28].

The proposed approach is developed first as a linear method, and then a kernel-based formulation is derived in order to confer flexibility to the approach. Our main contribution, therefore, is presenting two novel formulations for addressing the regression task via robust optimization. This approach has not been previously reported in the SVM literature, to the best of our knowledge. Our results confirm the virtues of the proposed robust approach, since it achieved the best predictive performance compared with other SVR formulations.

This paper is organized as follows: in Section 2, the methods ε-SVR and twin SVR are briefly introduced. Section 3 presents the proposed robust regression method based on SVM. Experimental

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

https://doi.org/10.1016/j.knosys.2018.04.005
0950-7051/

results on benchmark data sets are given in Section 4. The main findings obtained from our results and their implications are discussed in Section 5. Finally, the main conclusions of this work are summarized in Section 6, in which future developments are also addressed.

2. Prior work on support vector regression and twin SVR

Support Vector Machine is a well-known learning approach one, which has been widely used in several application domains, such as genomics [25], marine sciences [8], computer vision [45], and business analytics [27].

Among its virtues, SVM enables the construction of flexible nonlinear models thanks to the use of Kernel functions. Additionally, it reduces the risk of overfitting by applying the Structural Risk Minimization (SRM) principle, improving predictive performance [42]. Finally, it provides a flexible optimization problem that can be adapted for dealing with complexities related to the nature of the data, such as high dimensionality [25], class-imbalance [40], or the presence of noise [32,44]. Originally developed by V. Vapnik for binary classification [42], this approach has been extended to various machine learning tasks, such as regression [14], multiclass classification [43], and outlier detection [40], among others.

A brief overview of the SVR and twin SVR methods is given in this section. Specifically, we describe the ε-SVR method [14], and its extension based on the concept of reduced convex hulls developed by Bi and Bennett [7], the original TSVR method proposed by Peng [30], and the ε-TSVR method proposed by Shao et al. [36]. Additionally, recent developments on twin SVR are discussed.

2.1. ε-Support vector regression

For training examples $\mathbf{A} = [\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_m]^\top \in \Re^{m\times n}$, the linear ε-SVR approach [14] finds an optimal regression function of the form $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, which should be as close as possible to the corresponding responses $\mathbf{y} = (y_1, y_2, \ldots, y_m) \in \Re^m$, where the weights $\mathbf{w} \in \Re^n$ and the bias term $b \in \Re$ are decision variables. The ε-insensitive loss function is used to model the empirical risk, which can be interpreted as a tube where only the samples that lie outside of it are (linearly) penalized. The structural risk is minimized by including the Tikhonov regularization, with the goal of making the regression function $f(\mathbf{x})$ as flat as possible. The following quadratic programming problem (QPP) is solved:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\mathbf{e}^\top(\boldsymbol{\xi} + \boldsymbol{\xi}^*) \\ \text{s.t. } & \mathbf{y} - (A\mathbf{w} + b\mathbf{e}) \le \varepsilon\mathbf{e} + \boldsymbol{\xi},\ \ \boldsymbol{\xi} \ge \mathbf{0}, \\ & (A\mathbf{w} + b\mathbf{e}) - \mathbf{y} \le \varepsilon\mathbf{e} + \boldsymbol{\xi}^*,\ \ \boldsymbol{\xi}^* \ge \mathbf{0}, \end{aligned} \tag{1}$$

where $\boldsymbol{\xi}, \boldsymbol{\xi}^* \in \Re^m$ are slack variables that activate when objects are outside the tube, $C > 0$ controls the trade-off between structural and empirical risk, and $\mathbf{e} \in \Re^m$ is a vector of ones.

The ε-SVR method can be extended to kernel functions thanks to the kernel trick. The dual problem of Formulation (1) can be rewritten including of kernel functions as follows:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \quad & \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^\top K(\mathbf{A}, \mathbf{A}^\top)(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \mathbf{y}^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \varepsilon\,\mathbf{e}^\top(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \\ \text{s.t. } & \mathbf{e}^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0, \\ & \mathbf{0} \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le C\mathbf{e}, \end{aligned} \tag{2}$$

where $K(\mathbf{A}, \mathbf{A}^\top) \in \Re^{m\times m}$ is the matrix of kernel functions of the form $k_{is} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_s)$, and $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ are the dual variables related with the constraints of Problem (1). The Gaussian kernel, which has the following expression for two samples $\mathbf{x}_i$ and $\mathbf{x}_s \in \Re^n$, is used in this work:

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

where the kernel width parameter $\sigma > 0$ controls the shape of the kernel [35].

Alternatively, Bi and Bennett [7] developed a variation of the ε-SVR method that has a different geometrical interpretation; instead of using the ε-insensitive loss function, this formulation aims to maximize the margin between two training patterns. These patterns $\mathcal{D}^+ = \{(\mathbf{x}_i, y_i + \varepsilon) : i = 1, \ldots, m\}$ and $\mathcal{D}^- = \{(\mathbf{x}_i, y_i - \varepsilon) : i = 1, \ldots, m\}$ are constructed by shifting the target variable up and down by ε and adding it to the data points, resulting in the following data matrices:

$$A_1 = \begin{bmatrix} \mathbf{A}^\top \\ (\mathbf{y} + \varepsilon\mathbf{e})^\top \end{bmatrix},\ A_2 = \begin{bmatrix} \mathbf{A}^\top \\ (\mathbf{y} - \varepsilon\mathbf{e})^\top \end{bmatrix} \in \Re^{n+1\times m}. \tag{4}$$

The ε-SVR formulation by Bi and Bennett [7] then maximizes the margin between the closest points in the reduced convex hulls of $\mathcal{D}^+$ and $\mathcal{D}^-$, resulting in the following QPP:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{v}} \quad & \frac{1}{2}||A_1\mathbf{u} - A_2\mathbf{v}||^2 \\ \text{s.t. } & \mathbf{e}^\top\mathbf{u} = 1,\ \ \mathbf{e}^\top\mathbf{v} = 1, \\ & \mathbf{0} \le \mathbf{u}, \mathbf{v} \le D\mathbf{e}, \end{aligned} \tag{5}$$

where $D > 0$ is a parameter used to limit the influence of outliers [7]. The authors demonstrate that the dual formulation of (5) is equivalent to the original ε-SVR method (cf. Formulation (1)) under certain conditions for the parameters C, ε, and D [7].

2.2. Twin support vector regression

There are several extensions of ε-SVR that follow the original idea of constructing two non-parallel hyperplanes instead of a single function, as suggested originally in [20] for binary classification. The first twin SVR method was developed by Peng [30]. It constructs two regressors $f_1(\mathbf{x}) = \mathbf{w}_1^\top\mathbf{x} + b_1$ and $f_2(\mathbf{x}) = \mathbf{w}_2^\top\mathbf{x} + b_2$ that define the down- and up-bounds for the ε-insensitive tube by solving the following pair of QPPs:

$$\begin{aligned} \min_{\mathbf{w}_1,b_1,\boldsymbol{\xi}_1} \quad & \frac{1}{2}\|\mathbf{y} - \varepsilon_1\mathbf{e} - (\mathbf{A}\mathbf{w}_1 + b_1\mathbf{e})\|^2 + c_1\mathbf{e}^\top\boldsymbol{\xi}_1 \\ \text{s.t. } & \mathbf{y} - (\mathbf{A}\mathbf{w}_1 + b_1\mathbf{e}) \ge \varepsilon_1\mathbf{e} - \boldsymbol{\xi}_1,\ \ \boldsymbol{\xi}_1 \ge \mathbf{0}, \end{aligned} \tag{6}$$

and

$$\begin{aligned} \min_{\mathbf{w}_2,b_2,\boldsymbol{\xi}_2} \quad & \frac{1}{2}\|\mathbf{y} + \varepsilon_2\mathbf{e} - (\mathbf{A}\mathbf{w}_2 + b_2\mathbf{e})\|^2 + c_2\mathbf{e}^\top\boldsymbol{\xi}_2 \\ \text{s.t. } & (\mathbf{A}\mathbf{w}_2 + b_2\mathbf{e}) - \mathbf{y} \ge \varepsilon_2\mathbf{e} - \boldsymbol{\xi}_2,\ \ \boldsymbol{\xi}_2 \ge \mathbf{0}, \end{aligned} \tag{7}$$

where ε_1, $\varepsilon_2 > 0$ and c_1, $c_2 > 0$ are the input parameters, while $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2 \in \Re^m$ are the slack variables used to minimize the empirical risk.

The original TSVR approach has the disadvantage of not being strongly convex, causing suboptimal solutions (local minima) [36]. Therefore, Shao et al. [36] included extra regularization terms for the weight vectors related to both hyperplanes, in order to minimize the structural risk properly. Specifically, the authors consider the following pair of QPPs:

$$\begin{aligned} \min_{\mathbf{w}_1,b_1,\boldsymbol{\xi}_1} \quad & \frac{1}{2}\|\mathbf{y} - (\mathbf{A}\mathbf{w}_1 + b_1\mathbf{e})\|^2 + \frac{\hat{c}_1}{2}(\|\mathbf{w}_1\|^2 + b_1^2) + c_1\mathbf{e}^\top\boldsymbol{\xi}_1 \\ \text{s.t. } & \mathbf{y} - (\mathbf{A}\mathbf{w}_1 + b_1\mathbf{e}) \ge -\varepsilon_1\mathbf{e} - \boldsymbol{\xi}_1,\ \ \boldsymbol{\xi}_1 \ge \mathbf{0}, \end{aligned} \tag{8}$$

and

$$\begin{aligned} \min_{\mathbf{w}_2,b_2,\boldsymbol{\xi}_2} \quad & \frac{1}{2}\|\mathbf{y} - (\mathbf{A}\mathbf{w}_2 + b_2\mathbf{e})\|^2 + \frac{\hat{c}_2}{2}(\|\mathbf{w}_2\|^2 + b_2^2) + c_2\mathbf{e}^\top\boldsymbol{\xi}_2 \\ \text{s.t. } & (\mathbf{A}\mathbf{w}_2 + b_2\mathbf{e}) - \mathbf{y} \ge -\varepsilon_2\mathbf{e} - \boldsymbol{\xi}_2,\ \ \boldsymbol{\xi}_2 \ge \mathbf{0}, \end{aligned} \tag{9}$$

where $c_1, c_2, \hat{c}_1, \hat{c}_2, \varepsilon_1, \varepsilon_2$ are positive parameters.

The TSVR and ε-TSVR approaches were also adapted as kernel methods in [30] and [36], respectively. The two twin regression functions are defined as nonlinear surfaces of the form $f_1(\mathbf{x}) = K(\mathbf{x}^\top, \mathbf{A}^\top)\mathbf{w}_1 + b_1$ and $f_2(\mathbf{x}) = K(\mathbf{x}^\top, \mathbf{A}^\top)\mathbf{w}_2 + b_2$. The following QPPs are solved by the TSVR method:

$$\begin{aligned} \min_{\mathbf{w}_1, b_1, \boldsymbol{\xi}_1} \quad & \frac{1}{2}\|\mathbf{y} - \varepsilon_1\mathbf{e} - (K(\mathbf{A}, \mathbf{A}^\top)\mathbf{w}_1 + b_1\mathbf{e})\|^2 + c_1\mathbf{e}^\top\boldsymbol{\xi}_1 \\ \text{s.t. } & \mathbf{y} - (K(\mathbf{A}, \mathbf{A}^\top)\mathbf{w}_1 + b_1\mathbf{e}) \geq \varepsilon_1\mathbf{e} - \boldsymbol{\xi}_1, \ \ \boldsymbol{\xi}_1 \geq \mathbf{0}, \end{aligned} \tag{10}$$

and

$$\begin{aligned} \min_{\mathbf{w}_2, b_2, \boldsymbol{\xi}_2} \quad & \frac{1}{2}\|\mathbf{y} + \varepsilon_2\mathbf{e} - (K(\mathbf{A}, \mathbf{A}^\top)\mathbf{w}_2 + b_2\mathbf{e})\|^2 + c_2\mathbf{e}^\top\boldsymbol{\xi}_2 \\ \text{s.t. } & (K(\mathbf{A}, \mathbf{A}^\top)\mathbf{w}_2 + b_2\mathbf{e}) - \mathbf{y} \geq \varepsilon_2\mathbf{e} - \boldsymbol{\xi}_2, \ \ \boldsymbol{\xi}_2 \geq \mathbf{0}. \end{aligned} \tag{11}$$

For both linear and kernel-based twin SVR methods, the final regressor is obtained by simply averaging the two twin functions: $f(\mathbf{x}) = \frac{1}{2}(f_1(\mathbf{x}) + f_2(\mathbf{x}))$.

Several extensions of the TSVR method by Peng have been proposed. Chen et al. [10] reformulated TSVR as a pair of unconstrained, strongly convex optimization problems by using a smoothing technique, solving them with the Newton-Armijo algorithm. An approach similar to ε-TSVR was proposed in [39], in which a quadratic loss function is used, resulting in a strongly convex formulation.

Efficiency is another relevant topic when designing twin SVR approaches. Peng [29] proposed the Primal TSVR (PTSVR), which casts the original QPPs into a series of equations by replacing the hinge loss by a quadratic loss function. Balasundaram and Meena [5] also solved TSVR in the primal space, converting the quadratic problems into unconstrained optimization ones, which can be solved efficiently via gradient-based iterative methods. Singh et al. [37] extended TSVR by using rectangular kernels, allowing an efficient matrix inversion operation and thus reducing computational times. Peng [31] proposed the twin parametric insensitive support vector regression (TPISVR) approach, which introduces a parameter ν to control the number of support vectors.

An extension relevant for our work is the twin SVR approach proposed by Khemchandaniet et al. [21]. The authors extended the ε-SVR formulation by Bi and Bennett [7] to twin regression, leading to an alternative TSVR model that shows better predictive performance compared with the one proposed by Peng. In our work, we also follow the strategy proposed by Bi and Bennett [7] of maximizing the margin between the two augmented sets; but our robust framework uses ellipsoids to represent the training patterns, instead of the reduced convex hulls.

3. Robust twin support vector regression

In this section, a novel SVR method is presented. The main idea is to extend the robust framework of Saketha Nath and Bhattacharyya [34] for binary classification to twin SVR. This framework constructs maximum margin predictors by proposing a chance-constrained optimization problem, which is subsequently cast into an SOCP model by assuming a pessimistic data distribution. This approach was extended to twin SVM classification in [26], providing a good starting point for this research.

In order to adapt the robust twin SVM classification framework properly, we need to cast the SVR problem into a margin maximization one between two training patterns. In this context, the SVR method proposed by Bi and Bennett [7] provides a more suitable approach than the traditional ε-SVR by Drucker et al. [14].

Our proposal is introduced into two steps. First, the linear robust twin SVR formulation is presented in Section 3.1, in which the geometrical interpretation and other properties are derived. The kernel-based version of our proposal is subsequently described in Section 3.2.

3.1. Robust twin SVR - linear version

Following the ideas of Bi and Bennett [7], the proposed method constructs two nonparallel hyperplanes in such a way that each one of them is closest to one of the augmented sets, $\mathcal{D}^+ = \{(\mathbf{x}_i, y_i + \varepsilon) : i = 1, \ldots, m\}$ or $\mathcal{D}^- = \{(\mathbf{x}_i, y_i - \varepsilon) : i = 1, \ldots, m\}$, and as far as possible from the other. Instead of using reduced convex hulls, these sets are represented in our approach by the means and covariance matrices of the respective training samples.

Formally, let $\mathbf{X}_1$ and $\mathbf{X}_2$ be two random vectors that generate the samples A_1 and A_2 of the augmented sets $\mathcal{D}^+$ and $\mathcal{D}^-$, respectively (see Eq. (4)). Each hyperplane $f_k(\mathbf{x})$ is constructed to agree, in probabilistic terms, with the data from each augmented set at least to rates $\eta_k \in (0, 1)$, for $k = 1, 2$. In other words, the samples generated by $\mathbf{X}_1$ should be above the down bound of the ε-insensitive tube $f_2(\mathbf{x}) = \mathbf{w}_2^\top\mathbf{x} + b_2$, and the samples generated by $\mathbf{X}_2$ should be below the up bound of the ε-insensitive tube $f_1(\mathbf{x}) = \mathbf{w}_1^\top\mathbf{x} + b_1$; and these two conditions should not exceed error rates of $1 - \eta_2$ and $1 - \eta_1$, respectively. The following quadratic chance-constrained programming problems are proposed:

$$\begin{aligned} \min_{\mathbf{w}_1^*, b_1} \quad & \frac{1}{2}\left\|A_1^\top\mathbf{w}_1^* + b_1\mathbf{e}_1\right\|^2 + \frac{\theta_1}{2}(\|\mathbf{w}_1^*\|^2 + b_1^2) \\ \text{s.t. } & \Pr\{\mathbf{w}_1^{*\top}\mathbf{X}_2 + b_1 \leq -1\} \geq \eta_2, \end{aligned} \tag{12}$$

and

$$\begin{aligned} \min_{\mathbf{w}_2^*, b_2} \quad & \frac{1}{2}\left\|A_2^\top\mathbf{w}_2^* + b_2\mathbf{e}_2\right\|^2 + \frac{\theta_2}{2}(\|\mathbf{w}_2^*\|^2 + b_2^2) \\ \text{s.t. } & \Pr\{\mathbf{w}_2^{*\top}\mathbf{X}_1 + b_2 \geq 1\} \geq \eta_1, \end{aligned} \tag{13}$$

where $(\mathbf{w}_k^*, b_k) \in \Re^{n+1} \times \Re$ are the two solutions that define the twin hyperplanes, $\mathbf{e}_k$ are vectors of ones of appropriate dimensions, and $\theta_k > 0$ are trade-off parameters, for $k = 1, 2$. In contrast with Peng [30], our proposal includes the Tikhonov regularization, as suggested in Shao et al. [36] for twin SVR.

The proposed robust framework aims to predict these two patterns $\mathcal{D}^+$ and $\mathcal{D}^-$ accurately, even for the worst possible data distribution [26]. Therefore, the chance constraints in formulations (12)–(13) are replaced with their robust counterparts:

$$\begin{aligned} \inf_{\mathbf{X}_1 \sim (\boldsymbol{\mu}_1, \Sigma_1)} \Pr\{\mathbf{w}_2^{*\top}\mathbf{X}_1 + b_2 \geq 1\} \geq \eta_1, \\ \inf_{\mathbf{X}_2 \sim (\boldsymbol{\mu}_2, \Sigma_2)} \Pr\{\mathbf{w}_1^{*\top}\mathbf{X}_2 + b_1 \leq -1\} \geq \eta_2, \end{aligned}$$

where $\mathbf{X}_k \sim (\boldsymbol{\mu}_k, \Sigma_k)$ refers to the family of distributions with common mean $\boldsymbol{\mu}_k \in \Re^{n+1}$ and covariance $\Sigma_k \in \Re^{n+1 \times n+1}$, for $k = 1, 2$. The empirical estimates for these first two moments of the distribution are given by:

$$\boldsymbol{\mu}_1 = \begin{bmatrix} \boldsymbol{\mu}_\mathbf{x} \\ \mu_\mathbf{y} + \varepsilon \end{bmatrix}, \ \ \boldsymbol{\mu}_2 = \begin{bmatrix} \boldsymbol{\mu}_\mathbf{x} \\ \mu_\mathbf{y} - \varepsilon \end{bmatrix} \in \Re^{n+1}, \tag{14}$$

where $\boldsymbol{\mu}_\mathbf{x} = \frac{1}{m}\mathbf{A}^\top\mathbf{e} \in \Re^n$ corresponds to the means of the variables, and $\mu_\mathbf{y} = \frac{1}{m}\mathbf{y}^\top\mathbf{e} \in \Re$ the mean for the output vector. The covariance matrices are computed as follows:

$$\Sigma_1 = \Sigma_2 = \Sigma = \begin{bmatrix} \Sigma_\mathbf{x} & \Sigma_\mathbf{xy} \\ \Sigma_\mathbf{xy}^\top & \Sigma_\mathbf{y} \end{bmatrix} \in \Re^{n+1 \times n+1}, \tag{15}$$

with

$$\Sigma_\mathbf{x} = \frac{1}{m}\mathbf{A}^\top(I - \frac{1}{m}\mathbf{e}\mathbf{e}^\top)\mathbf{A} \in \Re^{n \times n}, \quad \Sigma_\mathbf{xy} = \frac{1}{m}\mathbf{A}^\top(I - \frac{1}{m}\mathbf{e}\mathbf{e}^\top)\mathbf{y} \in \Re^n,$$

and

$$\Sigma_\mathbf{y} = \frac{1}{m}\mathbf{y}^\top(I - \frac{1}{m}\mathbf{e}\mathbf{e}^\top)\mathbf{y} \in \Re.$$

In order to cast the proposed chance-constrained problem (Eqs. (12)–(13)) into a deterministic model, the multivariate Chebyshev inequality is used [22, Lemma 1], resulting in the following formulations:

$$\begin{aligned} &\min_{\mathbf{w}_1^*, b_1} \frac{1}{2}\left\|A_1^\top \mathbf{w}_1^* + b_1\mathbf{e}_1\right\|^2 + \frac{\theta_1}{2}(\|\mathbf{w}_1^*\|^2 + b_1^2) \\ &\text{s.t. } -\mathbf{w}_1^{*\top}\boldsymbol{\mu}_2 - b_1 \geq 1 + \kappa_2\sqrt{\mathbf{w}_1^{*\top}\Sigma_2\mathbf{w}_1^*}, \end{aligned} \tag{16}$$

and

$$\begin{aligned} &\min_{\mathbf{w}_2^*, b_2} \frac{1}{2}\left\|A_2^\top \mathbf{w}_2^* + b_2\mathbf{e}_2\right\|^2 + \frac{\theta_2}{2}(\|\mathbf{w}_2^*\|^2 + b_2^2) \\ &\text{s.t. } \mathbf{w}_2^{*\top}\boldsymbol{\mu}_1 + b_2 \geq 1 + \kappa_1\sqrt{\mathbf{w}_2^{*\top}\Sigma_1\mathbf{w}_2^*}, \end{aligned} \tag{17}$$

where $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$ for $k = 1, 2$. Formulations (16)–(17) are convex optimization problems; more precisely, they are quadratic ones with one second-order cone (SOC) constraint each.[1] General-purpose solvers such as SeDuMi [38] can handle those problems efficiently. This solver uses interior-point methods for SOCP [1,28], which yields a worst-case complexity of $O(n^3)$.

Let $\mathbf{w}_k^* = [\mathbf{w}_k^\top, \delta_k]^\top$. Using relations (14)–(15), formulations (16)–(17) can be rewritten equivalently as the following problems:

$$\begin{aligned} &\min_{\mathbf{w}_1, \delta_1, b_1} \frac{1}{2}\|A\mathbf{w}_1 + \delta_1(\mathbf{y} + \varepsilon\mathbf{e}) + b_1\mathbf{e}\|^2 + \frac{\theta_1}{2}(\|\mathbf{w}_1\|^2 + \delta_1^2 + b_1^2) \\ &\text{s.t. } -\mathbf{w}_1^\top\boldsymbol{\mu}_\mathbf{x} - \delta_1(\mu_\mathbf{y} - \varepsilon) - b_1 \geq 1 + \kappa_2\sqrt{\mathbf{w}_1^{*\top}\Sigma\mathbf{w}_1^*}, \end{aligned} \tag{18}$$

and

$$\begin{aligned} &\min_{\mathbf{w}_2, \delta_2, b_2} \frac{1}{2}\|A\mathbf{w}_2 + \delta_2(\mathbf{y} - \varepsilon\mathbf{e}) + b_2\mathbf{e}\|^2 + \frac{\theta_2}{2}(\|\mathbf{w}_2\|^2 + \delta_2^2 + b_2^2) \\ &\text{s.t. } \mathbf{w}_2^\top\boldsymbol{\mu}_\mathbf{x} + \delta_2(\mu_\mathbf{y} + \varepsilon) + b_2 \geq 1 + \kappa_1\sqrt{\mathbf{w}_2^{*\top}\Sigma\mathbf{w}_2^*}. \end{aligned} \tag{19}$$

We refer to these problems as the robust twin SVR method in its linear version (RT-SVR$_l$).

The following remark presents the decision rule for RT-SVR$_l$:

Remark 1. Formulations (18)–(19) provide two twin hyperplanes of the form $\hat{\mathbf{w}}_1^\top\mathbf{x} + \hat{\delta}_1 y + \hat{b}_1 = 0$ and $\hat{\mathbf{w}}_2^\top\mathbf{x} + \hat{\delta}_2 y + \hat{b}_2 = 0$. Assuming that $\hat{\delta}_k \neq 0$ for $k = 1, 2$; these hyperplanes can be rescaled, leading to two new functions $f_1(\mathbf{x}) = -\frac{1}{\hat{\delta}_1}(\hat{\mathbf{w}}_1^\top\mathbf{x} + \hat{b}_1)$ and $f_2(\mathbf{x}) = -\frac{1}{\hat{\delta}_2}(\hat{\mathbf{w}}_2^\top\mathbf{x} + \hat{b}_2)$. Then, the final regressor can be computed as the average between these two new functions, that is,

$$f(\mathbf{x}) = \frac{1}{2}(\bar{\mathbf{w}}_1 + \bar{\mathbf{w}}_2)^\top\mathbf{x} + \frac{1}{2}(\bar{b}_1 + \bar{b}_2), \tag{20}$$

where $\bar{\mathbf{w}}_k = -\frac{1}{\hat{\delta}_k}\hat{\mathbf{w}}_k$, $\bar{b}_k = -\frac{1}{\hat{\delta}_k}\hat{b}_k$, for $k = 1, 2$.

Next, the dual formulation of the RT-SVR$_l$ method is derived, providing insights regarding its geometrical interpretation. First, the following property is required in order to apply the duality theory properly:

Lemma 3.1. *The Lagrange multipliers related to the RT-SVR$_l$ method (problems (18)–(19)) are always different from zero.*

The proof of Lemma 3.1 is presented in Appendix A. This result allows the derivation of the dual formulation for RT-SVR$_l$.

[1] An SOC constraint on a given variable $\mathbf{x} \in \Re^n$ has the form $\|D\mathbf{x} + \mathbf{b}\| \leq \mathbf{c}^\top\mathbf{x} + d$, where $d \in \Re$, $\mathbf{c} \in \Re^n$, $\mathbf{b} \in \Re^m$ and $D \in \Re^{m\times n}$ (see [1] for more details).

Proposition 3.2. *The dual formulations for problems (18)–(19) are given by*

$$\begin{aligned} &\min_{\mathbf{z}_1, \mathbf{u}_1} \frac{1}{2}\begin{pmatrix}\mathbf{z}_1^\top & 1\end{pmatrix}(H^\top H + \theta_1 I)^{-1}\begin{pmatrix}\mathbf{z}_1 \\ 1\end{pmatrix} \\ &\text{s.t. } \mathbf{z}_1 \in \mathbf{B}(\boldsymbol{\mu}_2, \Sigma^{1/2}, \kappa_2), \end{aligned} \tag{21}$$

and

$$\begin{aligned} &\min_{\mathbf{z}_2, \mathbf{u}_2} \frac{1}{2}\begin{pmatrix}\mathbf{z}_2^\top & 1\end{pmatrix}(G^\top G + \theta_2 I)^{-1}\begin{pmatrix}\mathbf{z}_2 \\ 1\end{pmatrix} \\ &\text{s.t. } \mathbf{z}_2 \in \mathbf{B}(\boldsymbol{\mu}_1, \Sigma^{1/2}, \kappa_1), \end{aligned} \tag{22}$$

where $H = [\mathbf{A}, (\mathbf{y} + \varepsilon\mathbf{e}), \mathbf{e}]$; $G = [\mathbf{A}, (\mathbf{y} - \varepsilon\mathbf{e}), \mathbf{e}] \in \Re^{m\times n+2}$; and

$$\mathbf{B}(\boldsymbol{\mu}, \Sigma^{1/2}, \kappa) = \{\mathbf{z} \in \Re^{n+1} : \mathbf{z} = \boldsymbol{\mu} + \kappa\Sigma^{1/2}\mathbf{u}, \|\mathbf{u}\| \leq 1\}, \tag{23}$$

which denotes an ellipsoid centered at $\boldsymbol{\mu}$ whose shape is determined by $\Sigma^{1/2}$, and sized by κ.

The proof of Proposition 3.2 is presented in Appendix B. The dual form for RT-SVR$_l$ can be rewritten compactly by applying the Schur complement [18] to the matrices $(H^\top H + \theta_1 I)$ and $(G^\top G + \theta_2 I)$: Since the symmetric matrix

$$H^\top H + \theta_1 I = \begin{pmatrix} A_1 A_1^\top + \theta_1 I & A_1\mathbf{e} \\ \mathbf{e}^\top A_1^\top & \mathbf{e}^\top\mathbf{e} + \theta_1 \end{pmatrix}$$

is positive definite for each $\theta_1 > 0$, where A_1 is defined in (4), Theorem 7.7.6 in [18] implies that the matrix $C_s(\theta_1) = A_1 A_1^\top + \theta_1 I - \frac{1}{m+\theta_1}A_1\mathbf{e}\mathbf{e}^\top A_1^\top$ is invertible, and that

$$\begin{aligned} (H^\top H + \theta_1 I)^{-1} = &\begin{pmatrix} I & 0 \\ -\frac{1}{m+\theta_1}\mathbf{e}^\top A_1^\top & 1 \end{pmatrix}\begin{pmatrix} C_s(\theta_1)^{-1} & 0 \\ 0 & \frac{1}{m+\theta_1} \end{pmatrix} \\ &\begin{pmatrix} I & -\frac{1}{m+\theta_1}A_1\mathbf{e} \\ 0 & 1 \end{pmatrix}. \end{aligned} \tag{24}$$

Taking into account (24), and the fact that $\boldsymbol{\mu}_1 = \frac{1}{m}A_1\mathbf{e}$, the objective function of Formulation (21) can be rewritten as

$$\frac{1}{2}\left(\left(\mathbf{z}_1^\top - \frac{m}{m+\theta_1}\boldsymbol{\mu}_1^\top\right)C_s(\theta_1)^{-1}\left(\mathbf{z}_1 - \frac{m}{m+\theta_1}\boldsymbol{\mu}_1\right) + \frac{1}{m+\theta_1}\right).$$

Thus, the dual problem of the first twin formulation becomes:

$$\begin{aligned} &\min_{\mathbf{z}_1, \mathbf{u}_1} \frac{1}{2}\left\|C_s(\theta_1)^{-1/2}\left(\mathbf{z}_1 - \frac{m}{m+\theta_1}\boldsymbol{\mu}_1\right)\right\|^2 \\ &\text{s.t. } \mathbf{z}_1 \in \mathbf{B}(\boldsymbol{\mu}_2, \Sigma^{1/2}, \kappa_2). \end{aligned} \tag{25}$$

In a similar way, the dual form of the second twin problem can be derived:

$$\begin{aligned} &\min_{\mathbf{z}_2, \mathbf{u}_2} \frac{1}{2}\left\|C_s(\theta_2)^{-1/2}\left(\mathbf{z}_2 - \frac{m}{m+\theta_2}\boldsymbol{\mu}_2\right)\right\|^2 \\ &\text{s.t. } \mathbf{z}_2 \in \mathbf{B}(\boldsymbol{\mu}_1, \Sigma^{1/2}, \kappa_1). \end{aligned} \tag{26}$$

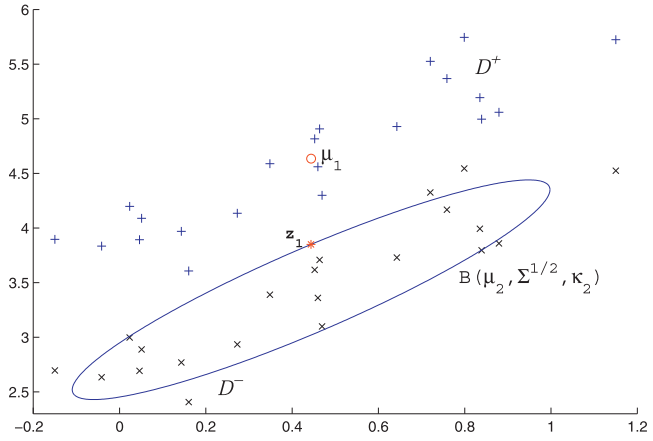
If $\theta_1 = \theta_2 = 0$, previous formulations result in models that can easily be interpreted geometrically. The following proposition states this idea:

Proposition 3.3. *If $\theta_1 = \theta_2 = 0$ is set, and if the symmetric matrices $H^\top H$, $G^\top G$ are positive definite, then the formulations (21)–(22) can be written equivalently as*

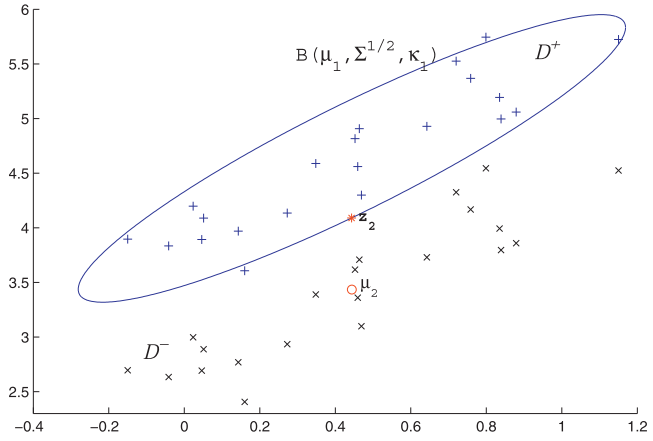
$$\begin{aligned} &\min_{\mathbf{z}_1, \mathbf{u}_1} \frac{1}{2}\left\|\Sigma^{-1/2}(\mathbf{z}_1 - \boldsymbol{\mu}_1)\right\|^2 \\ &\text{s.t. } \mathbf{z}_1 \in \mathbf{B}(\boldsymbol{\mu}_2, \Sigma^{1/2}, \kappa_2), \end{aligned} \tag{27}$$

and

$$\begin{aligned} &\min_{\mathbf{z}_2, \mathbf{u}_2} \frac{1}{2}\left\|\Sigma^{-1/2}(\mathbf{z}_2 - \boldsymbol{\mu}_2)\right\|^2 \\ &\text{s.t. } \mathbf{z}_2 \in \mathbf{B}(\boldsymbol{\mu}_1, \Sigma^{1/2}, \kappa_1). \end{aligned} \tag{28}$$



(a) Formulation (27)



(b) Formulation (28)

Fig. 1. Geometric interpretation for RT-SVR$_l$.

Proof. Note that

$$C_s(0) = A_1(I - \frac{1}{m}\mathbf{e}\mathbf{e}^\top)A_1^\top = m\Sigma_1 = m\Sigma. \tag{29}$$

Since $H^\top H$ is positive definite, $C_s(0)$ is invertible. Then, the proposition follows by replacing Eq. (29) in formulations (25)–(26). □

The above models can be interpreted as the problem of finding a point $\mathbf{z}$ on the ellipsoid $\mathbf{B}(\boldsymbol{\mu}_2, \Sigma^{1/2}, \kappa_2)$ (resp. $\mathbf{B}(\boldsymbol{\mu}_1, \Sigma^{1/2}, \kappa_1)$), associated with pattern $\mathcal{D}^-$ (resp. $\mathcal{D}^+$), whose Mahalanobis distance is minimal to the respective centers $\boldsymbol{\mu}_1$ (resp. $\boldsymbol{\mu}_2$). In Fig. 1, we illustrate the geometric interpretation for the proposed RT-SVR$_l$ for a given value of κ_k. The reader is referred to De Maesschalck et al. [11] for further information regarding the Mahalanobis distance.

3.2. Robust twin SVR - kernel version

In order to obtain a non-linear version for the proposed robust twin SVM model (RT-SVR$_k$), the weight vectors related to each twin hyperplane can be rewritten as $\mathbf{w}_k = \mathbb{X}\mathbf{s}_k + M\mathbf{r}_k$, where

$$\mathbb{X} = [A_1\ A_2] = \begin{bmatrix} \mathbf{A}^\top & \mathbf{A}^\top \\ (\mathbf{y}+\varepsilon\mathbf{e})^\top & (\mathbf{y}-\varepsilon\mathbf{e})^\top \end{bmatrix} \in \Re^{n+1\times 2m}, \tag{30}$$

M is a matrix whose columns are orthogonal to the training samples, and $\mathbf{s}_k$, $\mathbf{r}_k$ are vectors of combining coefficients with the appropriate dimensions. At optimality, the weights are simply $\mathbf{w}_k = \mathbb{X}\mathbf{s}_k$ since the proposed constraints are independent of M [34]. Following the reasoning behind the robust kernel models in [26,34], this modification leads to models where the data points appear only in the form of inner products, allowing the use of kernel functions.

The inner products between the training patterns A_1 and A_2 are given by:

$$\begin{aligned} A_1^\top A_1 &= \mathbf{A}\mathbf{A}^\top + (\mathbf{y}+\varepsilon\mathbf{e})(\mathbf{y}^\top + \varepsilon\mathbf{e}^\top), \\ A_2^\top A_2 &= \mathbf{A}\mathbf{A}^\top + (\mathbf{y}-\varepsilon\mathbf{e})(\mathbf{y}^\top - \varepsilon\mathbf{e}^\top), \\ A_1^\top A_2 &= (A_2^\top A_1)^\top = \mathbf{A}\mathbf{A}^\top + (\mathbf{y}+\varepsilon\mathbf{e})(\mathbf{y}^\top - \varepsilon\mathbf{e}^\top). \end{aligned} \tag{31}$$

For each pair of samples $\mathbf{x}_i$ and $\mathbf{x}_j$, the ijth entry of the matrix $\mathbf{A}\mathbf{A}^\top$ corresponds to the inner product $\mathbf{x}_i^\top\mathbf{x}_j$, which can be replaced by $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Let us denote by $\mathbf{K} \in \Re^{m\times m}$ the kernel matrix whose ijth entry is $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Then, each inner product $A_k^\top A_{k'}$ becomes $\mathbf{K}_{kk'} \in \Re^{m\times m}$, where

$$\begin{aligned} \mathbf{K}_{11} &= \mathbf{K} + (\mathbf{y}+\varepsilon\mathbf{e})(\mathbf{y}+\varepsilon\mathbf{e})^\top, \\ \mathbf{K}_{22} &= \mathbf{K} + (\mathbf{y}-\varepsilon\mathbf{e})(\mathbf{y}+\varepsilon\mathbf{e})^\top, \\ \mathbf{K}_{12} &= \mathbf{K}_{21}^\top = \mathbf{K} + (\mathbf{y}+\varepsilon\mathbf{e})(\mathbf{y}-\varepsilon\mathbf{e})^\top. \end{aligned} \tag{32}$$

Following the steps from the robust framework presented in [26], it holds that

$$\mathbf{w}_k^{*\top}\boldsymbol{\mu}_k = \mathbf{s}_k^\top\mathbf{g}_k, \quad \mathbf{w}_k^{*\top}\Sigma_k\mathbf{w}_k^* = \mathbf{s}_k^\top\Xi_k\mathbf{s}_k, \quad k = 1, 2, \tag{33}$$

and

$$A_1^\top\mathbf{w}_1^* = [\mathbf{K}_{11}\ \mathbf{K}_{12}]\mathbf{s}_1 = \mathbf{K}_{1\bullet}\mathbf{s}_1, \quad A_2^\top\mathbf{w}_2^* = [\mathbf{K}_{21}\ \mathbf{K}_{22}]\mathbf{s}_2 = \mathbf{K}_{2\bullet}\mathbf{s}_2, \tag{34}$$

where $\mathbf{s}_k$ is a vector of combining coefficients with the appropriate dimension, which replaces the weight vector as a decision variable in the optimization process, and

$$\mathbf{g}_k = \frac{1}{m_k}\begin{bmatrix}\mathbf{K_{1k}}\mathbf{e} \\ \mathbf{K}_{2k}\mathbf{e}\end{bmatrix}, \tag{35}$$

$$\Xi_k = \frac{1}{m_k}\begin{bmatrix}\mathbf{K_{1k}} \\ \mathbf{K}_{2k}\end{bmatrix}\left(I - \frac{1}{m_k}\mathbf{e}\mathbf{e}^\top\right)\left[\mathbf{K}_{1k}^\top\ \mathbf{K}_{2k}^\top\right], \tag{36}$$

for $k = 1, 2$. Using the relations (33) and (34) in the linear robust twin SVM model (Problems (16) and (17)), we can derive the kernel-based formulation for our proposal (RT-SVR$_k$), as follows:

$$\begin{aligned} &\min_{\mathbf{s}_1, b_1} \frac{1}{2}\|\mathbf{K}_{1\bullet}\mathbf{s}_1 + b_1\mathbf{e}_1\|^2 + \frac{\theta_1}{2}(\|\mathbf{s}_1\|^2 + b_1^2) \\ &\text{s.t. } -\mathbf{s}_1{}^\top\mathbf{g}_2 - b_1 \geq 1 + \kappa_2\sqrt{\mathbf{s}_1^\top\Xi_2\mathbf{s}_1}, \end{aligned} \tag{37}$$

and

$$\begin{aligned} &\min_{\mathbf{s}_2, b_2} \frac{1}{2}\|\mathbf{K}_{2\bullet}\mathbf{s}_2 + b_2\mathbf{e}_2\|^2 + \frac{\theta_2}{2}(\|\mathbf{s}_2\|^2 + b_2^2) \\ &\text{s.t. } \mathbf{s}_2{}^\top\mathbf{g}_1 + b_2 \geq 1 + \kappa_1\sqrt{\mathbf{s}_2^\top\Xi_1\mathbf{s}_2}. \end{aligned} \tag{38}$$

Finally, the following remark results in the decision rule for the proposed RT-SVR$_k$ method:

Remark 2. Formulations (37)–(38) lead to two hyperplanes of the form

$$\sum_{j=1}^{2m}\hat{\mathcal{K}}(\hat{\mathbf{x}}, \mathbb{X}_{\bullet j})s_1^j + b_1 = 0, \quad \sum_{j=1}^{2m}\hat{\mathcal{K}}(\hat{\mathbf{x}}, \mathbb{X}_{\bullet j})s_2^j + b_2 = 0, \tag{39}$$

with $\hat{\mathbf{x}} = (\mathbf{x}, y) \in \Re^{n+1}$. The expression $\mathbb{X}_{\bullet j}$ denotes the jth column of the matrix $\mathbb{X}$ (cf. Eq. (30)), and $\hat{\mathcal{K}}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) = \mathcal{K}(\mathbf{x}_1, \mathbf{x}_2) + y_1y_2$. Taking this last equality into account, the relations (39) can be rewritten as

$$\sum_{i=1}^{m}(s_1^i + s_1^{m+i})\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + y\delta_{ys}^1 + b_1 = 0,$$

$$\sum_{i=1}^{m}(s_2^i + s_2^{m+i})\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + y\delta_{ys}^2 + b_2 = 0,$$

where

$$\delta_{ys}^1 = \sum_{i=1}^{m}[y_i(s_1^i + s_1^{m+i}) + \varepsilon(s_1^i - s_1^{m+i})],$$

$$\delta_{ys}^2 = \sum_{i=1}^{m}[y_i(s_2^i + s_2^{m+i}) + \varepsilon(s_2^i - s_2^{m+i})].$$

Assuming that $\delta_{ys}^1, \delta_{ys}^2 \neq 0$, the hyperplanes in Eq. (39) can be rescaled, leading to the following kernel-based twin regression functions:

$$f_1(\mathbf{x}) = -\frac{1}{\delta_{ys}^1}\left(\sum_{i=1}^{m}(s_1^i + s_1^{m+i})\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + b_1\right)$$

and

$$f_2(\mathbf{x}) = -\frac{1}{\delta_{ys}^2}\left(\sum_{i=1}^{m}(s_2^i + s_2^{m+i})\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + b_2\right).$$

Then, the final regressor is constructed by averaging these twin functions: $f(\mathbf{x}) = \frac{1}{2}(f_1(\mathbf{x}) + f_2(\mathbf{x}))$.

4. Experimental results

The proposed robust twin SVR method was applied to an illustrative two-dimensional example with synthetic data, and to eleven benchmark datasets from the UCI Repository [3]. A brief description of the benchmark datasets is presented next:

- Triazines: This dataset studies the inhibition of rat mouse tumors by triazines based on structural attributes (compounds). This dataset consists of 186 samples described by 58 variables.
- Wisconsin Breast Cancer Prognosis (WBCP): This dataset aims to predict the time to recurrence for 198 breast cancer patients described by 32 features computed from digitalized images.
- Relative CPU Performance (CPU): The goal for this dataset is to estimate the relative performance values for 209 CPUs described in terms of 8 variables, such as cycle time, or memory size, among others.
- Auto MPG (A-MPG): This dataset concerns fuel consumption in miles per gallon for 398 cars described by 25 attributes.
- Boston Housing (Housing): This dataset is to predict housing values in the suburbs of Boston. It consists of 506 houses described by 13 variables, such as average number of rooms per dwelling and crime rate by town.
- Forest Fires (Fires): The goal for this dataset is to predict the burned area for wildfires in Portugal. A total of 517 events are studied, which are described by 12 variables, such as relative humidity, temperature, rain, and wind.
- Concrete Compressive Strength (Concrete): The concrete compressive strength is studied in terms of the compounds in it, such as cement, water, and Fly Ash. A total of 1080 samples described by three compounds is available.
- Wine quality (red, WQR): This dataset consists of 1599 red wine samples from Portugal. The goal is to assess the wine quality based on 11 physicochemical properties, such as acidity, residual sugar, chlorides, and density.
- Quake: This dataset studies 2178 earthquakes with magnitudes of at least 5.8 Richter that occurred between 1964 and 1986, for which the density for the focal depth, in kilometers, is estimated.
- Abalone: The goal for this dataset is to predict the age of abalone (4,177 samples) from 10 physical measurements.
- Parkinson's Disease Telemonitoring (Parkinson): This dataset consists of 5875 voice recordings from 42 Parkinson's disease patients. The goal is to predict the motor UPDRS score, the most commonly used scale in the clinical study for this disease, based on biomedical voice measures and other variables, such as age and sex (19 attributes in total).

The linear and kernel-based versions of the proposal, RT-SVR$_l$ and RT-SVR$_k$, respectively, were studied together with the following alternative approaches: standard linear regression, ε-SVR in its linear and kernel versions, the twin SVR approaches by Peng [30] (TSVR) and by Shao et al. [36] (ε-TSVR) in their linear and kernel-based forms.

The experimental setting follows that the parameters for each approach are tuned using ten-fold cross-validation, in which the whole dataset is divided into ten subsets. Each training subset includes 90% of the data, while the test set has the remaining 10%. The following sets of parameters are explored: C, σ, c_k, $\hat{c}_k$, $\theta_k \in \{2^{-7}, 2^{-6}, \ldots, 2^0, \ldots, 2^6, 2^7\}$, $\eta_k \in \{0.2, 0.4, 0.6, 0.8\}$, with $k = 1, 2$, and ε: $\{0.1, 0.2, 0.3, \ldots, 0.8, 0.9\}$. For kernel methods, we limit ourselves to the Gaussian kernel. The following relations for the twin SVR approaches are imposed in order to reduce the number of combinations in the grid search: $c_1 = c_2$, $\hat{c}_1 = \hat{c}_2$, $\theta_1 = \theta_2$ and $\eta_1 = \eta_2$. Regarding data normalization, all datasets were scaled to $[-1, 1]$. All experiments were performed in MATLAB R2016b. We used Matlab's fitlm function for linear regression, the LIBSVM toolbox [9] for ε-SVR, the code provided by Yuan-Hai Shao [36] for the TSVR and ε-TSVR methods, which is publicly available in http://www.optimal-group.org/ , and the SeDuMi toolbox for the proposed SOCP method [38].

4.1. An illustrative example

In this section, the regression function that results from the various SVR approaches discussed in this study is illustrated with a two-dimensional toy example. We compare the proposed SOCP approach with the traditional ε-SVR, TSVR [30], and ε-TSVR [36]. For all methods, their kernel-based formulations were used with the Gaussian kernel. The synthetic dataset was generated by using the *sinc function*, which is defined as

$$y = sinc(x) = \frac{\sin(x)}{x}, \quad x \in [-4\pi, 4\pi]. \tag{40}$$

A total of 252 training samples were generated using Eq. (40), with the inclusion of a Gaussian noise with zero mean and a standard deviation of 0.2. Furthermore, 500 test samples were created without the introduction of noise by assuming variable x as uniformly distributed over the interval $[-4\pi, 4\pi]$.

Fig. 2 illustrates the regression function obtained by ε-SVR, TSVR, ε-TSVR, and the proposed RT-SVR on the toy dataset. The test root-mean-square-error (RMSE) for the four models are 0.0507, 0.0495, 0.0458, and 0.0448, respectively. Therefore, the proposal shows a better model fit, as can be observed in Fig. 2 (the solid line represents the Eq. (40)).

4.2. Results summary for the benchmark data sets

The RMSE and the mean absolute percentage error (MAPE) are studied and reported as performance metrics. The best parameter configuration was selected using RMSE. These metrics have the following expression:

$$RMSE = \sqrt{\frac{1}{m}\sum_{t=1}^{m}(y_t - f(\mathbf{x}_t))^2}, \quad MAPE = \frac{1}{m}\sum_{t=1}^{m}\left|\frac{y_t - f(\mathbf{x}_t)}{y_t}\right|, \tag{41}$$

where y_t denotes the real output of a test sample $\mathbf{x}_t$. In the unusual case where $y_t = 0$, the sample was omitted for the computation of the MAPE.

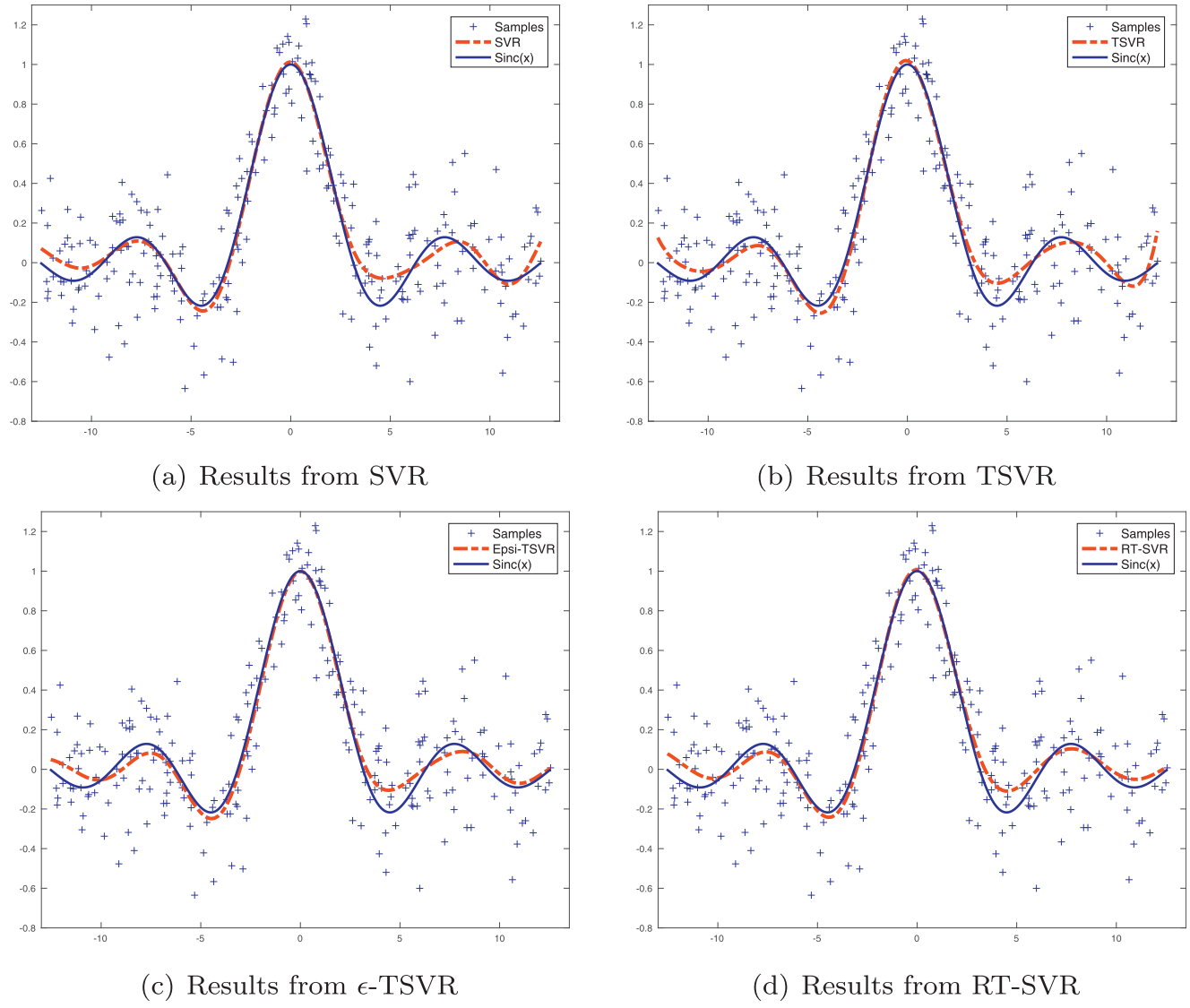


Fig. 2. Prediction with SVR [14], TSVR, ε-TSVR, and RT-SVR on the noisy samples.

Table 1
Performance (RMSE and MAPE) for various regression approaches (linear methods). All datasets.

	linear reg.		ε-SVR$_l$		TSVR$_l$		ε-TSVR$_l$		RT-SVR$_l$	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Triazines	0.157	0.190	**0.143**	**0.157**	0.157	0.188	0.145	0.175	0.144	0.161
WBCP	0.273	0.788	0.260	0.530	0.267	0.895	0.255	0.500	**0.254**	**0.496**
CPU	0.063	0.153	0.067	0.101	0.062	0.218	0.063	0.096	**0.061**	**0.089**
A-MPG	0.085	0.224	0.085	0.393	0.085	0.225	0.085	0.215	**0.083**	**0.202**
Housing	0.218	5.161	0.217	1.498	0.217	1.610	0.216	1.293	**0.209**	**0.904**
Fires	0.117	0.045	0.139	0.109	0.117	**0.036**	0.117	0.040	**0.117**	0.037
Concrete	0.261	1.752	0.262	3.649	0.261	2.439	0.261	1.623	**0.260**	**0.926**
WQR	**0.651**	0.089	**0.651**	**0.088**	**0.651**	0.089	**0.651**	0.089	**0.651**	0.089
Quake	0.189	0.025	0.190	**0.024**	0.189	0.024	0.189	0.024	**0.189**	0.024
Abalone	0.079	0.176	0.082	0.190	0.079	0.177	0.079	0.176	**0.079**	**0.176**
Parkinson	**0.435**	1.422	0.436	1.431	0.435	1.428	**0.435**	1.399	**0.435**	**1.390**

Tables 1 and 2 present the best performance for all datasets and for linear and kernel-based methods, respectively. The lowest error among all methods is highlighted in bold type for both performance metrics and for all datasets.

It can be observed in Tables 1 and 2, that our proposal achieves the best performance in most cases, in which both, performance metrics and the two families of methods (linear and kernel-based approaches) are considered. In the case of linear methods, the proposed RT-SVR$_l$ achieves the lowest RMSE in ten of the eleven datasets, and the lowest MAPE in seven of the eleven datasets (see Table 1). For kernel methods, the proposed RT-SVR$_k$ achieves the lowest RMSE in six of the eleven datasets, and the lowest MAPE in five of the eleven datasets (see Table 2). Among the alternative

Table 2
Performance (RMSE and MAPE) for various regression approaches (kernel methods). All datasets.

	ε-SVR$_k$		TSVR$_k$		ε-TSVR$_k$		RT-SVR$_k$	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Triazines	0.459	0.259	0.146	**0.235**	**0.141**	0.266	0.142	0.236
WBCP	0.253	0.629	0.268	0.647	0.255	1.003	**0.236**	**0.501**
CPU	0.015	0.012	0.017	0.014	0.045	0.040	**0.014**	**0.012**
A-MPG	0.077	0.182	0.077	0.180	0.077	0.182	**0.077**	**0.177**
Housing	0.142	0.585	0.137	0.465	0.136	**0.437**	**0.133**	0.613
Fires	0.117	0.024	0.117	0.037	0.117	0.047	**0.117**	**0.024**
Concrete	**0.142**	**0.715**	0.143	0.788	0.155	0.834	0.143	0.722
WQR	0.634	0.820	0.643	0.091	**0.625**	0.087	0.668	**0.087**
Quake	0.204	**0.023**	**0.156**	0.023	0.189	0.024	0.189	0.025
Abalone	0.076	**0.166**	0.076	0.166	0.076	0.172	**0.075**	0.167
Parkinson	0.239	0.740	0.258	**0.694**	**0.232**	0.736	0.248	0.756

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

Method	Mean rank	p value	$\alpha/(k-i)$	Action
RT-SVR$_l$	1.5000	-	-	-
ε-TSVR$_l$	2.5000	0.1573	0.0500	not reject
TSVR$_l$	3.2500	0.0133	0.0250	reject
linear reg.	3.5500	0.0022	0.0037	reject
ε-SVR$_l$	4.2000	0.0001	0.0125	reject

Table 4
Holm's post-hoc test for pairwise comparisons. Kernel methods.

Method	Mean rank	p value	$\alpha/(k-i)$	Action
RT-SVR$_k$	1.8000	-	-	-
ε-SVR$_k$	2.3500	0.3408	0.0500	not reject
TSVR$_k$	2.8500	0.0690	0.0250	not reject
ε-TSVR$_k$	3.0000	0.0377	0.0167	not reject

approaches, no method seems to outperform the others in terms of both error measures.

5. Discussion

In the previous section, we observed that the proposed method achieves the lowest error in most datasets. In order to validate these results, the Holm's test [17] is used to evaluate whether or not the best method is able to outperform the others statistically when cross-validation is used. This test is constructed as follows: first, the average rank among all datasets is computed for each method. Then, pairwise comparisons between each technique and the one with the best rank are performed (see [12] for a detailed description). This analysis is performed for the RMSE metric, the main performance measure on this study, and for both linear (Table 3) and kernel methods (Table 4). A significant level of $\alpha = 0.05$ was used in both cases.

In Tables 3 and 4, we observe that our proposals RT-SVR$_l$ and RT-SVR$_k$ have the best overall rank in both cases. Among the linear methods, RT-SVR$_l$ is able to outperform TSVR$_l$, linear regression, and ε-SVR$_l$; although ε-TSVR$_l$ is not worse statistically than our proposal. For the kernel methods, the differences are not significant.

It can be concluded that our approach clearly achieves the best overall performance of all the methods. Although RT-SVR is not able to outperform all the others statistically, it does it with linear regression and ε-SVR$_l$, which are the best known approaches among those studied in our work. We strongly believe that the gain in terms of performance is due to the proposed robust framework, which confers robustness to the twin SVR approach.

Another important result is the fact that kernel-based SVR methods are better in general than their linear counterparts. Although linear formulations are useful for gaining insights into the particular applications, in this case the use of kernels may lead to major reductions in terms of average errors.

The robust framework is designed to predict well, even with the worst data distribution, given a mean vector and a covariance matrix. In other words, the method is designed to be robust in the presence of noise, this being particularly useful in datasets with missing values. A positive predictive performance can be expected in such cases, regardless the imputation technique. Alternative robust approaches based on SOCP have been used for dealing with noise in the form of measurement errors, which has demonstrated the usefulness of such strategies [32].

It can be seen that our proposal is particularly successful when dealing with small-sized datasets in terms of data samples. Therefore, another virtue related to our proposal is the data requirements for training. Regarding dimensionality, all the datasets reported in this paper are rather low dimensional given the scope of our work. Feature selection is recommended in high-dimensional settings, since it provides important advantages, such as an improvement in predictive performance, better understanding of the outcome of the modeling process for decision-making, and reduced storage and acquisition costs. However, feature selection is more challenging in twin SVM than in standard SVM because two hyperplanes are constructed, and each one of them can consider a different subset of variables as relevant. Although this issue can be resolved using independent regularizers for each subproblem [33], we recommend synchronized feature selection, using a group penalty function that penalizes the weights related to a given variable jointly in both hyperplanes [25].

Regarding the training complexity of our proposal in comparison with the alternative approaches, the following statements can be made:

- Both TSVR and ε-TSVR require the inversion of a (kernel) matrix of size $(n+1)\times(n+1)$ and $(m+1)\times(m+1)$ for the linear and nonlinear cases, respectively, with a complexity of $\mathcal{O}(n^3)$ and $\mathcal{O}(m^3)$, respectively.
- Moreover, these two methods require three matrix multiplications. The two methods, therefore, have a complexity of $\mathcal{O}(2n^2m + m^2n)$ for the linear case, while the order is $\mathcal{O}(3m^3)$ for the kernel-based formulation.
- Taking the required iterations into account when using a quadratic solver such as SOR, the total computational complexity of the TSVR and ε-TSVR methods can be estimated as $\mathcal{O}(n^3 + 2n^2m + m^2n) + \#iteration \times \mathcal{O}(m)$ and $\mathcal{O}(m^3) + \#iteration \times \mathcal{O}(m)$, where $\#iteration$ is the number of them that SOR requires, and therefore $\#iteration \times \mathcal{O}(m)$

corresponds to the complexity of the SOR method for a m-sized problem.
- Regarding our proposal, an SOCP solver such as SeDuMi has a complexity of $\mathcal{O}(n^3)$ for the linear case, and $\mathcal{O}((2m)^3)$ for the nonlinear. Additionally, the computational cost of estimating the first and second moments needs to be included. This cost corresponds to $\mathcal{O}(mn^2)$ and $\mathcal{O}(m^3)$ for the linear and kernel-based formulations, respectively. Finally, the total complexity of our approach is $\mathcal{O}(n^3 + mn^2)$ and $\mathcal{O}(m^3)$ for the linear and nonlinear cases, respectively.

Given that n, the number of attributes, is usually smaller than the figure of training samples m, our approach is able to reduce the complexity of them such as TSVR and ε-TSVR in their linear formulations. For the kernel-based approaches, however, these three have similar complexity. Notice that we have excluded the testing complexity from this analysis, since the application of the final regressor to a test set is quite similar for the three twin SVR methods discussed.

It can be concluded that our approach achieves superior predictive performance without increasing the computational complexity when compared with other twin SVR approaches. However, a generic solver such as SeDuMi may lead to higher training times in comparison with highly-optimized SVM solvers such as LIBLINEAR [15] or LIBSVM [9]. These solvers exploit the structure of the problem to derive efficient optimization strategies and reduce the original complexity of the problem. This is the main limitation of our approach. Efficient optimization strategies for SOCP formulations are therefore suggested for future developments. A possible line of research is to extend the idea behind the Reduced Support Vector Machine (RSVM), introduced by Lee and Mangasarian [23], which uses a reduced mixture with kernels sampled from a certain candidate set.

Along the same line, solvers such as LIBSVM use incremental approaches for optimizing the SVM formulation without the need of performing algebraic operations on big kernel matrices. This has opened interesting fields of research not only in SVM optimization, but also in incremental learning. The latter is the study of both supervised and unsupervised learning algorithms when data becomes a continuous flux of information. The idea of such algorithms is to update a current model efficiently, without the need for a complete retraining [13]. Although this line of research falls outside the scope of the current work, we believe that incremental algorithms for SOCP can be quite useful for both improving the training times of our proposal, and for opening the possibility of new SVM-based algorithms for incremental learning.

6. Conclusions

In this paper, a novel SVM-based regression method is presented. Following the ideas of Peng [30] and Khemchandani et al. [21], the proposed approach constructs two twin hyperplanes to define an ε-insensitive tube. The main contribution is the proposed robust framework for maximum-margin regression. Following the work of Saketha Nath and Bhattacharyya [34] for binary classification, probabilistic constraints can be designed to impose adequate agreement between the twin regressors and the random variables that generated the training samples. These chance constraints can be cast subsequently into an SOCP formulation by assuming a pessimistic approach for the data distribution, not only providing an efficient training, but also conferring robustness to twin SVR.

Our proposal is developed first as a linear regression method (RT-SVR$_l$), and subsequently extended to include kernel functions RT-SVR$_k$). The proposed approach in its kernel-based version achieved best overall performance compared with alternative regression models using ten benchmark datasets. Furthermore, the dual formulation of RT-SVR$_l$ is derived in order to obtain the geometrical interpretation of our approach: two training patterns are obtained by shifting the dependent variable up and down by ε (the augmented sets), and our approach characterizes these patterns via ellipsoids, whose centers and shapes are determined by the first two moments of the data distribution.

Following the ideas of Bi and Bennett [7], the proposed method constructs two nonparallel hyperplanes in such a way that the two of them is closest to one of the augmented sets, $\mathcal{D}^+ = \{(\mathbf{x}_i, y_i + \varepsilon) : i = 1, \ldots, m\}$ or $\mathcal{D}^- = \{(\mathbf{x}_i, y_i - \varepsilon) : i = 1, \ldots, m\}$, and as far as possible from the other. Instead of using reduced convex hulls, these sets are represented in our approach by the means and covariance matrices of the respective training samples.

This work opens interesting possibilities for future developments. One of the most important issues in data analysis is dimensionality reduction. A low-dimensional data representation usually leads to a better predictive performance, allowing also a better understanding and visualization of the outcome of the modelling process for decision-making, among other benefits [16]. In this context, one of the problems of twin SVM is that the construction of two hyperplanes leads to little information about the variables that are relevant for the problem, or the rules that define the decision function. One possible future area of research consists of adapting the training process in order to derive the importance of the covariates automatically, leading to predictors that include only the relevant variables for the problem. A few efforts have been made to incorporate feature selection in twin SVM classification (see, for instance, [4,24,25]), but it remains an open problem for twin SVR. Another possible research line is the use of decision trees to extract rules from the SVM predictor in order to provide a better understanding of the model, as it was suggested in Huysmans et al. [19] for binary classification. A final future development is the use of ad-hoc metrics to evaluate the performance of the model, taking the costs of under- and overestimating the real value of the target variable into account. Measures as MAPE and RMSE weight errors equally, and may not be the best approaches in applications such as demand forecasting, where shortage or surplus have totally different consequences in decision making.

Acknowledgments

The first author was supported by FONDECYT project 1160894, while the second was by FONDECYT project 1160738. This research was partially funded by the Complex Engineering Systems Institute, ISCI (ICM-FIC: P05-004-F, CONICYT: FB0816). The authors are grateful to the anonymous reviewers who contributed to improve the quality of the original paper.

Appendix A. Proof of Lemma 3.1

Proof. We note that both formulations are convex problems with a Slater point. Thus, the Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient for them (see, for instance, [6]). Let L_1 be the Lagrangian function related to Formulation (18), which is given by

$$\begin{aligned} L_1(\mathbf{w}_1, \delta_1, b_1, \mathbf{v}) = & \frac{1}{2}\|\mathbf{A}\mathbf{w}_1 + \delta_1(\mathbf{y} + \varepsilon\mathbf{e}) + b_1\mathbf{e}\|^2 \\ & + \frac{\theta_1}{2}(\|\mathbf{w}_1\|^2 + \delta_1^2 + b_1^2) - \nu_0(-\mathbf{w}_1^\top \boldsymbol{\mu}_{\mathbf{x}} \\ & - \delta_1(\boldsymbol{\mu}_{\mathbf{y}} - \varepsilon) - b_1 - 1) - \kappa_2 \mathbf{v}_1^\top \Sigma^{1/2} \mathbf{w}_1^*, \end{aligned} \tag{37}$$

with $\mathbf{v} = (\nu_0, \mathbf{v}_1) \in \Re \times \Re^{n+1}$. Then, the KKT conditions for Formulation (18) are given by

$$\mathbf{A}^\top(\mathbf{A}\mathbf{w}_1 + \delta_1(\mathbf{y} + \varepsilon\mathbf{e}) + b_1\mathbf{e}) + \theta_1\mathbf{w}_1 + \nu_0\boldsymbol{\mu}_{\mathbf{x}} - \kappa_2\Sigma_{\mathbf{w}}^{1/2}\mathbf{v}_1 = \mathbf{0}, \tag{38}$$

$$(\mathbf{y}+\varepsilon\mathbf{e})^\top(\mathbf{A}\mathbf{w}_1+\delta_1(\mathbf{y}+\varepsilon\mathbf{e})+b_1\mathbf{e})+\theta_1\delta_1+v_0(\boldsymbol{\mu}_\mathbf{y}-\varepsilon) -\kappa_2\Sigma_\delta^{1/2}\mathbf{v}_1=0, \tag{39}$$

$$\mathbf{e}^\top(\mathbf{A}\mathbf{w}_1+\delta_1(\mathbf{y}+\varepsilon\mathbf{e})+b_1\mathbf{e})+\theta_1 b_1+v_0=0, \tag{40}$$

$$-\mathbf{w}_1^\top\boldsymbol{\mu}_\mathbf{x}-\delta_1(\boldsymbol{\mu}_\mathbf{y}-\varepsilon)-b_1-1\geq\kappa_2\|\Sigma^{1/2}\mathbf{w}_1^*\|, \tag{41}$$

$$v_0\geq\|\mathbf{v}_1\|, \tag{42}$$

$$v_0(-\mathbf{w}_1^\top\boldsymbol{\mu}_\mathbf{x}-\delta_1(\boldsymbol{\mu}_\mathbf{y}-\varepsilon)-b_1-1)+\kappa_2\mathbf{v}_1^\top\Sigma^{1/2}\mathbf{w}_1^*=0, \tag{43}$$

where $\Sigma^{1/2}=[\Sigma_\mathbf{w}^{1/2};\Sigma_\delta^{1/2}]$, with $\Sigma_\mathbf{w}^{1/2}\in\Re^{n\times n+1}$ and $\Sigma_\delta^{1/2}\in\Re^{1\times n+1}$. The following equation can be obtained by multiplying (38) by $\mathbf{w}_1$, (39) by δ_1, (40) by b_1, and then summing the resulting expressions:

$$\|\mathbf{A}\mathbf{w}_1+\delta_1(\mathbf{y}+\varepsilon\mathbf{e})+b_1\mathbf{e}\|^2+\theta_1(\|\mathbf{w}_1\|^2+\delta_1^2+b_1^2)+v_0\mathbf{w}_1^\top\boldsymbol{\mu}_\mathbf{x} +v_0(\delta_1(\mu_\mathbf{y}-\varepsilon)+b_1)-\kappa_2\mathbf{v}_1^\top\Sigma^{1/2}\mathbf{w}_1^*=0. \tag{44}$$

Using (43) in the above equality leads to the following expression for v_0:

$$v_0=\|\mathbf{A}\mathbf{w}_1+\delta_1(\mathbf{y}+\varepsilon\mathbf{e})+b_1\mathbf{e}\|^2+\theta_1(\|\mathbf{w}_1\|^2+\delta_1^2+b_1^2). \tag{45}$$

Note that $v_0=0$ if and only if $\mathbf{w}_1=\mathbf{0}$, $\delta_1=b_1=0$. This result contradicts relation (41). Hence, the Lagrange multiplier $\mathbf{v}$ related to the conic constraint in Formulation (18) is always different from zero. The proof corresponding to the conic constraint in the second twin problem (Formulation (19)) can be developed similarly. □

Appendix B. Proof of Proposition 3.2

Proof. Let us denote by $\boldsymbol{\omega}_1=[\mathbf{w}_1^\top,\delta_1,b_1]^\top\in\Re^{n+2}$. Then, the Lagrangian (37) can be rewritten as

$$L_1=\frac{1}{2}\boldsymbol{\omega}_1^\top(H^\top H+\theta_1 I)\boldsymbol{\omega}_1-\kappa_2\mathbf{v}_1^\top\Sigma^{1/2}\mathbf{w}_1^*-v_0(-\mathbf{w}_1^\top\boldsymbol{\mu}_\mathbf{x} -\delta_1(\boldsymbol{\mu}_\mathbf{y}-\varepsilon)-b_1-1).$$

Using Eq. (44) in the Lagrangian L_1, the latter can be rewritten as

$$L_1=-\frac{1}{2}\boldsymbol{\omega}_1^\top(H^\top H+\theta_1 I)\boldsymbol{\omega}_1+v_0. \tag{46}$$

Since $v_0\neq0$, we can denote by $\hat{\mathbf{v}}_1=\mathbf{v}_1/v_0$ and by

$$\hat{\mathbf{z}}=\begin{bmatrix}\boldsymbol{\mu}_\mathbf{x}-\kappa_2\Sigma_\mathbf{w}^{1/2}\hat{\mathbf{v}}_1\\(\boldsymbol{\mu}_\mathbf{y}-\varepsilon)-\kappa_2\Sigma_\delta^{1/2}\hat{\mathbf{v}}_1\\1\end{bmatrix}=\begin{bmatrix}\boldsymbol{\mu}_2-\kappa_2\Sigma^{1/2}\hat{\mathbf{v}}_1\\1\end{bmatrix}.$$

Then, the relations (38)–(40) can be written compactly as $(H^\top H+\theta_1 I)\boldsymbol{\omega}_1=-v_0\hat{\mathbf{z}}$. Since the symmetric matrix $H^\top H+\theta_1 I$ is positive definite, it holds that

$$\boldsymbol{\omega}_1=-v_0(H^\top H+\theta_1 I)^{-1}\hat{\mathbf{z}},$$

for any $\theta_1>0$. Using the above relations in (B.1), the Lagrangian L_1 becomes

$$L_1=-\frac{v_0^2}{2}\hat{\mathbf{z}}^\top(H^\top H+\theta_1 I)^{-1}\hat{\mathbf{z}}+v_0.$$

Hence, the dual problem for (18) can be stated as follows:

$$\max_{\mathbf{z},\hat{\mathbf{v}}_1,v_0}\ -\frac{1}{2}v_0^2\hat{\mathbf{z}}^\top(H^\top H+\theta_1 I)^{-1}\hat{\mathbf{z}}+v_0 \quad \text{s.t. } \|\hat{\mathbf{v}}_1\|\leq 1,\ v_0>0. \tag{47}$$

Notice that the objective function of the dual problem (B.2) is concave with respect to v_0, and it attains its maximum value at

$$v_0^*=\frac{1}{\hat{\mathbf{z}}^\top(H^\top H+\theta_1 I)^{-1}\hat{\mathbf{z}}}, \tag{48}$$

with optimal value

$$\frac{1}{2}\frac{1}{\hat{\mathbf{z}}^\top(H^\top H+\theta_1 I)^{-1}\hat{\mathbf{z}}}.$$

The dual problem (21) is obtained by using this fact. It can be proven that the dual form for the second twin problem (Formulation (19)) becomes Eq. (22). □

References

[1] F. Alizadeh, D. Goldfarb, Second-order cone programming, Math. Program. 95 (2003) 3–51.
[2] 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.
[3] K. Bache, M. Lichman, UCI machine learning repository, 2013.
[4] L. Bai, Z. Wang, Y.-H. Shao, N.-Y. Deng, A novel feature selection method for twin support vector machine, Knowl. Based Syst. 59 (2014) 1–8.
[5] S. Balasundaram, Y. Meena, Training primal twin support vector regression via unconstrained convex minimization, Appl. Intell. 44 (4) (2016) 931–955.
[6] D. Bertsekas, Nonlinear Programming, 2nd, Athena Scientific, 1999.
[7] J. Bi, P. Bennett, A geometric approach to support vector regression, Neurocomputing 55 (2003) 78–108.
[8] P. Bosch, J. López, H. Ramírez, H. Robotham, Support vector machine under uncertainty: an application for hydroacoustic classification of fish-schools in Chile, Expert Syst. Appl. 40 (10) (2013) 4029–4034.
[9] 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.
[10] X. Chen, J. Yang, J. Liang, Q. Ye, Smooth twin support vector regression, Neural Comput. Appl. 21 (2012) 505–513.
[11] R. De Maesschalck, D. Jouan-Rimbaud, D. Massart, The mahalanobis distance, Chemom. Intell. Lab. Syst. 50 (2000) 1–18.
[12] J. Demšar, Statistical comparisons of classifiers over multiple data set, J. Mach. Learn. Res. (2006) 1–30.
[13] C.P. Diehl, G.Cauwenberghs, Svm incremental learning, adaptation and optimization, in: Proceedings of the IEEE International Joint Conference on Neural Networks, 4, 2003.
[14] H. Drucker, C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, in: Advances in Neural Information Processing Systems (NIPS), 9, MIT Press, 1997, pp. 155–161.
[15] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin., Liblinear: a library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.
[16] S. García, J. Luengo, F. Herrera, Tutorial on practical tips of the most influential data preprocessing algorithms in data mining, Knowl. Based Syst. 98 (2016) 1–29.
[17] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (2) (1979) 65–70.
[18] R. A. Horn, C. R. Johnson (Eds.), Matrix Analysis, Cambridge University Press, New York, NY, USA, 1st edition, 1990.
[19] J. Huysmans, B. Baesens, J. Vanthienen, Using rule extraction to improve the comprehensibility of predictive models, 2006, Research 0612, K.U.Leuven KBI.
[20] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.
[21] R. Khemchandani, K. Goyal, S. Chandra, Twsvr: regression via twin support vector machine, Neural Netw. 74 (2016) 14–21.
[22] G. Lanckriet, L. Ghaoui, C. Bhattacharyya, M. Jordan, A robust minimax approach to classification, J. Mach. Learn. Res. 3 (2003) 555–582.
[23] Y.J. Lee, O. Mangasarian, Rsvm: reduced support vector machines, in: Data Mining Institute, Computer Sciences Department, University of Wisconsin, 2001, pp. 00–07.
[24] J. López, S. Maldonado, Group-penalized feature selection and robust twin svm classification via second-order cone programming, Neurocomputing 235 (2017) 112–121.
[25] S. Maldonado, J. López, Synchronized feature selection for support vector machines with twin hyperplanes, Knowl. Based Syst. 132 (2017) 119–128.
[26] S. Maldonado, J. López, M. Carrasco, A second-order cone programming formulation for twin support vector machines, Appl. Intell. 45 (2) (2016) 265–276.
[27] S. Maldonado, J. Pérez, C. Bravo, Cost-based feature selection for support vector machines - an application in credit scoring, Eur. J. Oper. Res. 261 (2) (2017) 656–665.
[28] Y. Nesterov, A. Nemirovsky, Interior point polynomial methods in convex programming: Theory and applications, Soc. Ind. Appl. Math. (1994).
[29] X. Peng, Primal twin support vector regression and its sparse approximation, Neurocomputing 73 (2010) 2846–2858.

[30] X. Peng, Tsvr: an efficient twin support vector machine for regression, Neural Netw. 23 (3) (2010) 365–372.
[31] X. Peng, Efficient twin parametric insensitive support vector regression model, Neurocomputing 79 (2012) 26–38.
[32] Z. Qi, Y. Tian, Y. Shi, Robust twin support vector machine for pattern classification, Pattern Recognit. 46 (1) (2013) 305–316.
[33] R. R. Rastogi, P. Ananda, S. Chandra, L1-norm twin support vector machine-based regression, Optimization 66 (11) (2017) 1895–1911.
[34] 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.
[35] B. Schölkopf, A.J. Smola., Learning with Kernels, MIT Press, 2002.
[36] Y.-H. Shao, C.-H. Zhang, Z.-M. Yang, L. Jing, N. Deng, An epsilon-twin support vector machine for regression, Neural Comput. Appl. 23 (2013) 175–185.
[37] M. Singh, J. Chadha, P. Ahuja, Jayadeva, S. Chandra, Reduced twin support vector regression, Neurocomputing 74 (2011) 1474–1477.
[38] J. Sturm, Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, Optim. Methods Softw. 11 (12) (1999) 625–653. Special issue on Interior Point Methods (CD supplement with software).
[39] M. Tanveer, K. Shubham, M. Aldhaifallah, K. Nisar, An efficient implicit regularized lagrangian twin support vector regression, Appl. Intell. 44 (4) (2016) 831–848.
[40] D.M.J. Tax, R. Duin, Support vector data description, Mach. Learn. 54 (2004) 45–66.
[41] D. Tomar, S. Agarwal, A comparison on multi-class classification methods based on least squares twin support vector machine, Knowl. Based Syst. 81 (2015) 131–147.
[42] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[43] J. Weston, C. Watkins, Multi-class support vector machines, in: Proceedings of the Seventh European Symposium on Artificial Neural Networks., 1999.
[44] X. Yang, G. Zhang, J. Lu, J. Ma, A kernel fuzzy c-means clustering based fuzzy support vector machine algorithm for classification problems with outliers or noises, Fuzzy Syst. IEEE Trans. 19 (1) (2011) 105–115.
[45] Y. Zhao, Y. Lu, Y. Tian, L. Li, Q. Ren, X. Chai, Image processing based recognition of images with a limited number of pixels using simulated prosthetic vision, Inf. Sci. 180 (16) (2010) 2915–2924.