Neurocomputing 305 (2018) 59–69

Contents lists available at ScienceDirect

Neurocomputing

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

Ellipsoidal support vector regression based on second-order cone programming

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

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

ARTICLE INFO

Article history:
Received 8 December 2017
Revised 26 March 2018
Accepted 5 April 2018
Available online 4 May 2018

Communicated by Wei Chiang Hong

Keywords:
Support vector regression
Robust optimization
Second-order cone programming
Kernel methods

ABSTRACT

In this paper, we propose a novel method for Support Vector Regression (SVR) based on second-order cones. The proposed approach defines a robust worst-case framework for the conditional densities of the input data. Linear and kernel-based second-order cone programming formulations for SVR are proposed, while the duality theory allows us to derive interesting geometrical properties for this strategy: the method maximizes the margin between two ellipsoids obtained by shifting the response variable up and down by a fixed parameter. Experiments for regression on twelve well-known datasets confirm the superior performance of our proposal compared to alternative methods such as standard SVR and linear regression.

1. Introduction

Regression is a very important machine learning task. Regression can be performed by Support Vector Regression (SVR), a popular statistical learning technique that aims at constructing a function with minimal error but as "flat" as possible at the same time [11]. This balance between empirical loss and complexity leads to a better generalization of new data points, according to the structural risk minimization principle [31].

Robust optimization deals with optimization problems in which a certain measure of robustness is sought against uncertainty by creating a deterministic equivalent, called the robust counterpart. Robust optimization is popular because it does not assume that probability distributions are known. In machine learning, robustness is the property that characterizes how effective an algorithm is while being tested on a new independent dataset. In the other words, the robust algorithm is the one with the testing error which is close to the training error. A robust performance of an algorithm is the one which does not deteriorate very much when being trained and tested with slightly different data, making the algorithm prone to overfitting [8,23].

Second-order cone programming (SOCP) [4,22] is a convex optimization technique that has been used successfully in machine learning methods like Support Vector Machine (SVM) for binary and multi-class classification [9,23,27,33]. In particular, the robust framework proposed by Saketha Nath and Bhattacharyya [27] is designed to construct robust classifiers in such a way that each in-sample class recall can be higher than a predefined value η, even for the worst possible data distribution for given means and covariance matrices [24,27].

It is important to notice that this line of research differs from the one proposed by Zhong and Fukushima [33], which is designed to deal with measurement errors and noisy data in general. Robust approaches like the one proposed in [33] were extended regression in Chuang et al. [7] to deal with uncertainty due to measurement errors in the covariates. All the previously mentioned approaches are different in nature compared to the one proposed here since they do not characterize the training patterns as ellipsoids.

In this work, we extend the robust framework proposed by Saketha Nath and Bhattacharyya to SVR, adapting the formulation proposed by Bi and Bennett [5] for this method (RH-SVR). The RH-SVR approach maximizes the margin that separates the two reduced convex hulls which result from shifting the dependent variable up and down by a parameter ε [5]. Geometrically speaking, our approach constructs ellipsoids based on the first two moments of the distribution of the two resulting sets of points, instead of considering the reduced convex hulls. The linear version of the proposal is derived first, demonstrating its geometrical properties by using the duality theory, and is then further extended to kernel methods in order to construct nonlinear regressors.

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

https://doi.org/10.1016/j.neucom.2018.04.035
0925-2312/

This paper is organized as follows: in Section 2 the relevant SVR formulations are discussed briefly. The proposed method based on SOCP for SVR is presented in Section 3. The results of our experiments are described in Section 4. Finally, our conclusions and ideas for future developments are presented in Section 5.

2. Literature overview on support vector regression

Support Vector Regression has been applied successfully in various domains. For example, it has been used in medical diagnosis for predicting diseases [32], such as the time to tumor recurrence in various types of cancer [25]. Other applications are time-series forecasting for predicting electric load consumption [12], monthly precipitation [14], or vessel traffic flow [21]. SVR has also been used for determining collar dimensions around bridge piers in order to reduce scours, an important cause of bridge failures [18]. Finally, SVR has also been used in robotics for optimal designs of robot gripper and finger mechanisms [19,28].

In this section, we describe the standard SVR formulation [11] and RH-SVR [5], an SVR extension based on the concept of reduced convex hulls, which is useful for the design of our approach.

Given a set of training samples $\mathbf{x}_i \in \Re^n$, and the corresponding response value $y_i \in \Re$, for $i = 1, \ldots, m$, the linear SVR method finds a regression function of the form $f(x) = \mathbf{w}^\top \mathbf{x} + b$ by solving the following minimization problem:

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

where $A = [\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_m]^\top \in \Re^{m\times n}$, $\mathbf{y} = (y_1, y_2, \ldots, y_m) \in \Re^m$, $\mathbf{e} \in \Re^m$ is a vector of ones, $\boldsymbol{\xi}, \boldsymbol{\xi}^* \in \Re^m$ are positive slack vectors that indicate if the samples are inside the ε-insensitive tube or not, and $C > 0$ is the regularization factor that weights the trade-off between the fitting errors and the flatness of the linear regression function [11].

A non-linear regression function can be constructed by mapping the data points onto a higher-dimensional feature space $\mathscr{H}$. Since the training samples appear only in the form of dot products in the dual formulation of Problem (1), this projection can be performed by a kernel function $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_s)$, satisfying Mercer's condition (see [26]), which defines a dot product in $\mathscr{H}$ [29]. This kernel-based formulation follows:

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

where $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ are the dual variables associated with the constraints of Problem (1), and $K(A, A^\top) \in \Re^{m\times m}$ is the kernel matrix whose elements are $k_{is} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_s)$. Arguably the most popular kernel function is the Gaussian, which has the following form: $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_s) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_s||^2}{2\sigma^2}\right)$, where $\sigma > 0$ controls the kernel width [29]. The proper choice of the kernel function is still a matter of research, which can be tackled e.g. via meta-learning [1].

Bi and Bennett proposed an SVR extension by rewriting Formulation (1) as the problem of finding the closest points in the reduced convex hulls of two sets of data points, which consist of the training samples augmented with the dependent variable shifted up and down by $\varepsilon > 0$ [5].

Formally, let $\mathcal{D}^+ = \{(\mathbf{x}_i, y_i + \varepsilon) : i = 1, \ldots, m\}$ and $\mathcal{D}^- = \{(\mathbf{x}_i, y_i - \varepsilon) : i = 1, \ldots, m\}$ be the augmented sets for the shifted response variable, and let us consider the data matrices

$$A_1 = \begin{bmatrix} \mathbf{A}^\top \\ (\mathbf{y} + \varepsilon\mathbf{e})^\top \end{bmatrix}, \quad A_2 = \begin{bmatrix} \mathbf{A}^\top \\ (\mathbf{y} - \varepsilon\mathbf{e})^\top \end{bmatrix} \in \Re^{n+1\times m}, \tag{3}$$

associated with $\mathcal{D}^+$ and $\mathcal{D}^-$. The problem of maximizing the margin between the closest points in the reduced convex hulls of $\mathcal{D}^+$ and $\mathcal{D}^-$ becomes:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{v}} \quad & \frac{1}{2}||A_1\mathbf{u} - A_2\mathbf{v}||^2 \\ \text{s.t. } & \mathbf{e}^\top\mathbf{u} = 1, \ \mathbf{e}^\top\mathbf{v} = 1, \\ & 0 \le \mathbf{u} \le D\mathbf{e}, \ 0 \le \mathbf{v} \le D\mathbf{e}, \end{aligned} \tag{4}$$

where $D > 0$ is a fixed parameter used to reduce the convex hulls, limiting the influence of extreme data points [5]. Bi and Bennett compute the dual formulation of Problem (4), and demonstrate that this dual model is equivalent to ε-SVR (cf. Formulation (1)) under certain conditions for C, ε, and D [5].

3. Proposed robust SVR-SOCP formulation

In this section, we present a novel approach for maximum margin regression using second-order cones.

In machine learning, robustness refers to how effective a model is when it is being tested on new data. The performance of a robust model should not deteriorate very much when it is trained and tested on data with slightly different distributions, i.e. it generalizes well without overfitting. Inspired by the robust optimization theory, our strategy proposes a chance-constrained formulation, which is cast further into an SOCP formulation by assuming a worst-case setting for the data distribution.

In our proposal, we provide robustness to Support Vector Regression, capitalizing the virtues of the RH-SVR formulation [5]. This method is a different representation of the epsilon-SVR model, which constructs an epsilon-insensitive tube by shifting the target variable up and down by epsilon and adding it to the data points. This strategy, which is different from minimizing the traditional epsilon-insensitive loss function, leads to two training patterns whose separation margin is maximized [5].

Under some conditions for the hyperparameters, the optimal hyperplanes obtained with RH-SVR and epsilon-SVR are equivalent. However, the geometrical properties of RH-SVR make it suitable for applying the robust min-max framework developed by Saketha Nath and Bhattacharyya for binary classification [27]. For this task, traditional soft-margin SVM constructs a separating hyperplane by maximizing the separation between the two training patterns given by the points of the two training samples, which are represented by their respective convex hulls. Saketha Nath and Bhattacharyya proposed using ellipsoids for representing these two patterns instead of the convex hulls, leading to a SOCP model based on robust optimization, which proved to be superior in terms of predictive performance [27].

Our main contribution is threefold: First, we identified a suitable formulation for applying the Saketha Nath and Bhattacharyya framework for the regression task. The RH-SVR model provides both a natural and creative starting point for developing robust regressors since it maximizes the margin between two convex hulls. Secondly, we propose a robust SOCP model by replacing these convex hulls with ellipsoids whose shapes are governed by the first two moments of the up-bound and down-bound distribution functions. Finally, a kernel-based formulation is derived from this robust model, conferring flexibility for modeling complex nonlinear patterns.

This paper, therefore, contributes by proposing two novel formulations that obtained superior predictive performance compared with other regression methods. To the best of our knowledge, this approach has not been reported previously in the SVM literature.

3.1. Second-order cone programming SVR, linear version

The proposal is first described as a chance-constrained optimization problem. Following the framework presented in [27], let $\mathbf{X}_1$ and $\mathbf{X}_2$ be random vectors that generate the samples of $\mathcal{D}^+$ and $\mathcal{D}^-$, respectively. The main idea is to construct a hyperplane that agrees with the data from both augmented sets at least to a rate η_k, $k = 1, 2$. A slack variable $\xi \geq 0$ is introduced as a relaxation of the chance constraint, while the objective functions represent the trade-off between margin maximization (minimization of the Euclidean norm of the weight vector) and model fit provided by the value of ξ [24], controlled by a parameter $C > 0$. The chance-constrained model has the following form:

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

with $\mathbf{w}^* = [\mathbf{w}^\top, \delta]^\top \in \Re^{n+1}$. The vector $\mathbf{w}$ represents the weights related to each of the n covariates, while δ multiplies the shifted response variable in the separating hyperplane.

Following the robust setting suggested by Saketha Nath and Bhattacharyya for classification [27], each of the shifted training pattern $k = 1, 2$ should be on the right side of the hyperplane, at least to a rate η_k, even for the *worst data distribution* for given means and covariance matrices given by $(\boldsymbol{\mu}_k, \Sigma_k)$ for $k = 1, 2$, where $\Sigma_k \in \Re^{n+1 \times n+1}$ are positive semidefinite symmetric matrices. For this goal, the probability constraints in (5) are replaced by their *robust* counterparts:

$$\begin{aligned} &\inf_{\mathbf{X}_1 \sim (\boldsymbol{\mu}_1, \Sigma_1)} \Pr\{\mathbf{w}^{*\top}\mathbf{X}_1 + b \geq 1 - \xi\} \geq \eta_1, \\ &\inf_{\mathbf{X}_2 \sim (\boldsymbol{\mu}_2, \Sigma_2)} \Pr\{\mathbf{w}^{*\top}\mathbf{X}_2 + b \leq -1 + \xi\} \geq \eta_2. \end{aligned} \tag{6}$$

In order to obtain a deterministic formulation, we use the multivariate Chebyshev-Cantelli inequality:

Lemma 3.1. *[20, Lemma 1] Let* $\mathbf{X}$ *be a* n*-dimensional random variables with mean and covariance* $(\boldsymbol{\mu}, \Sigma)$*, where* Σ *is a positive semidefinite symmetric matrix. Given* $\mathbf{a} \in \Re^n$*,* $b \in \Re$ *and* $\eta \in (0, 1)$*, the condition*

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

holds if and only if

$$\mathbf{a}^\top \boldsymbol{\mu} + b \geq \kappa \sqrt{\mathbf{a}^\top \Sigma \mathbf{a}},$$

where $\kappa = \sqrt{\frac{\eta}{1-\eta}}$.

Applying Lemma 3.1 to the constraints given in (6), we can construct the following deterministic problem:

$$\begin{aligned} \min_{\mathbf{w}^*, b, \xi} \quad & \frac{1}{2}\|\mathbf{w}^*\|^2 + C\xi \\ \text{s.t.} \quad & \mathbf{w}^{*\top}\boldsymbol{\mu}_1 + b \geq 1 - \xi + \kappa_1 \sqrt{\mathbf{w}^{*\top}\Sigma_1 \mathbf{w}^*}, \\ & -(\mathbf{w}^{*\top}\boldsymbol{\mu}_2 + b) \geq 1 - \xi + \kappa_2 \sqrt{\mathbf{w}^{*\top}\Sigma_2 \mathbf{w}^*}, \\ & \xi \geq 0, \end{aligned} \tag{7}$$

or, equivalently

$$\begin{aligned} \min_{\mathbf{w}, b, \delta, \xi} \quad & \frac{1}{2}(\|\mathbf{w}\|^2 + \delta^2) + C\xi \\ \text{s.t.} \quad & \mathbf{w}^\top \boldsymbol{\mu}_\mathbf{x} + \delta(\boldsymbol{\mu}_\mathbf{y} + \varepsilon) + b \geq 1 - \xi + \kappa_1 \sqrt{\mathbf{w}^{*\top}\Sigma_1 \mathbf{w}^*}, \\ & -\mathbf{w}^\top \boldsymbol{\mu}_\mathbf{x} - \delta(\boldsymbol{\mu}_\mathbf{y} - \varepsilon) - b \geq 1 - \xi + \kappa_2 \sqrt{\mathbf{w}^{*\top}\Sigma_2 \mathbf{w}^*}, \\ & \xi \geq 0, \end{aligned} \tag{8}$$

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

$$\boldsymbol{\mu}_1 = \begin{bmatrix} \boldsymbol{\mu}_\mathbf{x} \\ \boldsymbol{\mu}_\mathbf{y} + \varepsilon \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} \boldsymbol{\mu}_\mathbf{x} \\ \boldsymbol{\mu}_\mathbf{y} - \varepsilon \end{bmatrix} \in \Re^{n+1}. \tag{9}$$

The vectors $\boldsymbol{\mu}_\mathbf{x}$ and $\boldsymbol{\mu}_\mathbf{y}$ represent the means for the covariates and target variable, respectively, and are computed as $\boldsymbol{\mu}_\mathbf{x} = \frac{1}{m}\mathbf{A}^\top \mathbf{e} \in \Re^n$ and $\boldsymbol{\mu}_\mathbf{y} = \frac{1}{m}\mathbf{y}^\top \mathbf{e} \in \Re$. The covariance matrix Σ_k can be computed as

$$\Sigma_k = \frac{1}{m_k} A_k \left(I - \frac{1}{m_k}\mathbf{e}\mathbf{e}^\top\right) A_k^\top,$$

where m_k denotes the number of columns of A_k. By using this relation and Eq. (3), we have that

$$\Sigma_1 = \Sigma_2 = \Sigma = \begin{bmatrix} \Sigma_\mathbf{x} & \Sigma_{\mathbf{xy}} \\ \Sigma_{\mathbf{xy}}^\top & \Sigma_\mathbf{y} \end{bmatrix} \in \Re^{n+1 \times n+1}, \tag{10}$$

with

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

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

and

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

The first two constraints appearing in Problem (8) are called second-order cone (SOC) constraints[1] [2]. Thus, we refer to this problem as the SVR-SOCP$_l$ formulation.

Next, we present the regression rule for the SVR-SOCP$_l$ method, but before formalizing it, it is important to assure that this function is not undefined, which may be the case for $\delta = 0$.

Remark 1. Let $(\hat{\mathbf{w}}, \hat{\delta}, \hat{b}, \hat{\xi})$ be a solution of the formulation (8). By [24, Lemma 1] we have that $\hat{\xi} \in [0, 1]$, and that if $\hat{\xi} = 1$, we get $\hat{\mathbf{w}} = 0$, $\hat{\delta} = \hat{b} = 0$. We assume that $\hat{\mathbf{w}} \neq 0$. Then, $\hat{\xi} \neq 1$. On the other hand, from the constraints of (8) it follows that

$$\hat{\delta}\varepsilon \geq 1 - \hat{\xi} + \frac{(\kappa_1 + \kappa_2)}{2} \sqrt{\hat{\mathbf{w}}^\top \Sigma_\mathbf{x} \hat{\mathbf{w}} + 2\hat{\delta}\Sigma_{\mathbf{xy}}^\top \hat{\mathbf{w}} + \hat{\delta}^2 \Sigma_\mathbf{y}}.$$

Since $\varepsilon > 0$, the above relation implies that $\hat{\delta} > 0$.

Remark 2. Formulation (8) leads to a hyperplane of the form $\hat{\mathbf{w}}^\top \mathbf{x} + \hat{\delta} y + \hat{b} = 0$. Assuming that $\hat{\mathbf{w}} \neq 0$, we have that $\hat{\delta} > 0$ (cf. Remark 1), thus, we can rescale this hyperplane in order to obtain the regression function, which is given by

$$f(\mathbf{x}) = -\frac{1}{\hat{\delta}}(\hat{\mathbf{w}}^\top \mathbf{x} + \hat{b}). \tag{11}$$

Our proposal has an interesting geometrical property: it can be seen as a margin-maximization problem between two ellipsoids, whose centers and shapes are controlled by the means and covariances of $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively. In other words, our proposal has a similar interpretation compared to the RH-SVR method, with the

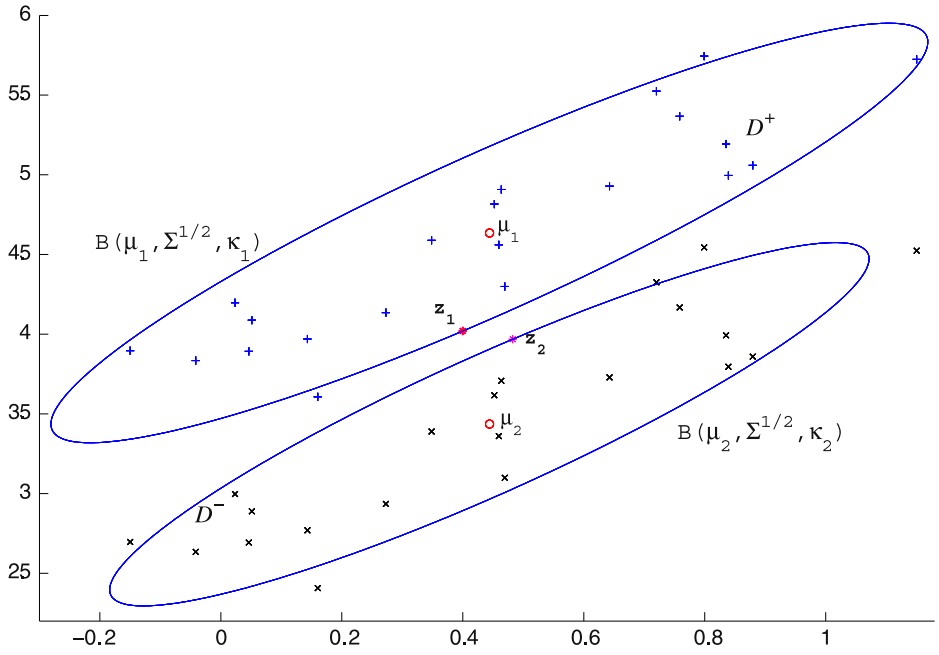


Fig. 1. Geometric interpretation for SVR-SOCP.

only difference being that the convex hulls are replaced by ellipsoids. This fact is illustrated in Fig. 1 with a two-dimensional toy example.

Remark 3. It follows from [24] (see [24, Appendix B] for details) that the dual formulation of Problem (7) is given by

$$\begin{aligned} \min_{\mathbf{z}_1,\mathbf{z}_2} \ & \frac{1}{2}\|\mathbf{z}_1 - \mathbf{z}_2\|^2 \\ \text{s.t. } & \mathbf{z}_k \in \mathbf{B}(\boldsymbol{\mu}_k, \Sigma^{1/2}, \kappa_k),\ k = 1, 2, \\ & \|\mathbf{z}_1 - \mathbf{z}_2\| \geq \frac{2}{\sqrt{C}}, \end{aligned} \tag{12}$$

where $\mathbf{B}(\boldsymbol{\mu}, \Sigma^{1/2}, \kappa) = \{\mathbf{z} \in \Re^{n+1} : \mathbf{z} = \boldsymbol{\mu} - \kappa\Sigma^{1/2}\mathbf{u},\ \ \|\mathbf{u}\| \leq 1\}$. This set denotes an ellipsoid centered at $\boldsymbol{\mu}$ whose shape is determined by $\Sigma^{1/2}$, and size by κ. Thus, the dual problem (12) can be seen as finding the minimum distance between two ellipsoids generated by the sets $\mathcal{D}^+$ and $\mathcal{D}^-$, under the constraint $\|\mathbf{z}_1 - \mathbf{z}_2\| \geq \frac{2}{\sqrt{C}}$.

The inclusion of the slack variable ξ avoids the intersection between the sets that results from the dual formulation of the SOCP problem, in which, in that case, the only feasible solution is $\mathbf{w} = \mathbf{0}$ [24,27].

Following the work by Bi and Bennett [5], the SVR-SOCP$_l$ method can be rewritten in such a way that δ can be omitted.

Proposition 3.1. *Let us consider the following problem*

$$\begin{aligned} \min_{\tilde{\mathbf{w}},\tilde{b},\tilde{\xi}} \ & \frac{1}{2}\|\tilde{\mathbf{w}}\|^2 + \tilde{C}\tilde{\xi} \\ \text{s.t. } & \tilde{\mathbf{w}}^\top\boldsymbol{\mu}_{\mathbf{x}} + \tilde{b} - \boldsymbol{\mu}_{\mathbf{y}} \leq \tilde{\varepsilon} + \tilde{\xi} - \kappa_1\sqrt{\tilde{\mathbf{w}}^{*\top}\Sigma\tilde{\mathbf{w}}^*}, \\ & \boldsymbol{\mu}_{\mathbf{y}} - (\tilde{\mathbf{w}}^\top\boldsymbol{\mu}_{\mathbf{x}} + \tilde{b}) \leq \tilde{\varepsilon} + \tilde{\xi} - \kappa_2\sqrt{\tilde{\mathbf{w}}^{*\top}\Sigma\tilde{\mathbf{w}}^*}, \\ & \tilde{\xi} \geq 0, \end{aligned} \tag{13}$$

where $\tilde{\mathbf{w}}^* = [-\tilde{\mathbf{w}}^\top, 1]^\top \in \Re^{n+1}$, $\tilde{\varepsilon} > 0$ *and* $\tilde{C} > 0$. *Suppose that (8) and (13) have* $\mathbf{w}$ *and* $\tilde{\mathbf{w}}$ *as nonzero solutions, respectively. Then, the formulations (7) and (13) are equivalent for appropriate choices of* ε, $\tilde{\varepsilon}$, C *and* $\tilde{C}$.

The proof for Proposition 3.1 is presented in Appendix A. As mentioned in Section 2, Bi and Bennett used this strategy to rewrite the RH-SVR method in such a way that it becomes equivalent to ε-SVR. We can also observe similarities between our proposal, written as in Formulation (13), and ε-SVR. Primarily, the objective functions are similar since both represent the trade-off between the Euclidean norm of the weight vector and the model fit. Secondly, the constraints conform with the definition of the ε-insensitive loss function for both the elements of the training patterns that are predicted to be larger and lower than the response variable for the first and second conic constraints, respectively.

It is important to note that the proposed formulation (13) differs from the one proposed by Huang et al. [17] to deal with measurement errors. Their proposal solves the following SOCP problem (see [17, Eq. (32)]):

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\xi}^*} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\mathbf{e}^\top(\boldsymbol{\xi} + \boldsymbol{\xi}^*) + D\mathbf{e}^\top\boldsymbol{\alpha} \\ \text{s.t. } & \mathbf{w}^\top\bar{\mathbf{x}}_i + b - \bar{y}_i \leq \varepsilon + \xi_i, \quad i = 1, \ldots, m \\ & \bar{y}_i - (\mathbf{w}^\top\bar{\mathbf{x}}_i + b) \leq \varepsilon + \xi_i^*, \quad i = 1, \ldots, m \\ & \|\Sigma_{zz}^{i\ 1/2}\mathbf{a}\| \leq \alpha_i\sqrt{\beta}, \quad i = 1, \ldots, m \\ & \boldsymbol{\xi}, \boldsymbol{\xi}^* \geq 0, \end{aligned} \tag{14}$$

where C, $D > 0$, $\beta \in (0, 1)$, $\bar{\mathbf{x}}_i$ and $\bar{y}_i$ denote the mean and covariance of the observations $\mathbf{x}_i$ and y_i, respectively, Σ_{zz}^i denotes the covariance of $\mathbf{z}_i = [\mathbf{x}_i^\top, y_i]^\top$, and $\mathbf{a} = [\mathbf{w}^\top; -1]^\top \in \Re^{n+1}$. Clearly, this model has m SOC constraints designed for dealing with a subset of noisy covariates.

3.2. Second-order cone programming SVR, Kernel-based version

In order to obtain a non-linear version for SVR-SOCP, first we compute the product $A_k^\top A_{k'}$, for $k, k' = 1, 2$. These products are given by:

$$A_1^\top A_1 = AA^\top + (\mathbf{y} + \varepsilon\mathbf{e})(\mathbf{y}^\top + \varepsilon\mathbf{e}^\top),$$

[1] An SOC constraint on the variable $\mathbf{x} \in \Re^n$ is of 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 $D \in \Re^{m\times n}$.

$$A_2^\top A_2 = AA^\top + (\mathbf{y} - \varepsilon\mathbf{e})(\mathbf{y}^\top - \varepsilon\mathbf{e}^\top),$$

and

$$A_1^\top A_2 = (A_2^\top A_1)^\top = AA^\top + (\mathbf{y} + \varepsilon\mathbf{e})(\mathbf{y}^\top - \varepsilon\mathbf{e}^\top).$$

Note that the ijth entry of the matrix $AA^\top$ is the inner product $\mathbf{x}_i^\top \mathbf{x}_j$. Then, we can replace each ijth entry of the matrix $AA^\top$ by $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathcal{K}$ denotes a kernel function satisfying Mercer's condition (see [26]). Let us denote by $\mathbf{K} \in \Re^{m\times m}$ the matrix whose ijth entry is $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Thus, each product $A_k^\top A_{k'}$ will be replaced by $\mathbf{K}_{kk'} \in \Re^{m\times m}$, where

$$\mathbf{K}_{11} = \mathbf{K} + (\mathbf{y} + \varepsilon\mathbf{e})(\mathbf{y}^\top + \varepsilon\mathbf{e}^\top), \tag{15}$$

$$\mathbf{K}_{22} = \mathbf{K} + (\mathbf{y} - \varepsilon\mathbf{e})(\mathbf{y}^\top - \varepsilon\mathbf{e}^\top), \tag{16}$$

$$\mathbf{K}_{12} = \mathbf{K}_{21}^\top = \mathbf{K} + (\mathbf{y} + \varepsilon\mathbf{e})(\mathbf{y}^\top - \varepsilon\mathbf{e}^\top). \tag{17}$$

Following the framework presented in Saketha Nath and Bhattacharyya [27], the following relations hold for $k = 1, 2$:

$$\mathbf{w}^{*\top}\boldsymbol{\mu}_k = \mathbf{s}^\top \mathbf{g}_k, \quad \mathbf{w}^{*\top}\Sigma_k \mathbf{w}^* = \mathbf{s}^\top \Xi_k \mathbf{s}, \tag{18}$$

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

$$\mathbf{g}_k = \frac{1}{m_k}\begin{bmatrix}\mathbf{K_{1k}e}\\ \mathbf{K}_{2k}\mathbf{e}\end{bmatrix}, \tag{19}$$

$$\Xi_k = \frac{1}{m_k}\begin{bmatrix}\mathbf{K_{1k}}\\ \mathbf{K}_{2k}\end{bmatrix}\left(I_{m_k} - \frac{1}{m_k}\mathbf{e}\mathbf{e}^\top\right)\left[\mathbf{K}_{1k}^\top\ \mathbf{K}_{2k}^\top\right], \tag{20}$$

for $k = 1, 2$.

Replacing the equalities (15)–(17) in (19) and (20), we obtain that

$$\mathbf{g}_1 = \begin{bmatrix}\frac{1}{m}\mathbf{Ke} + (\boldsymbol{\mu}_\mathbf{y} + \varepsilon)(\mathbf{y} + \varepsilon\mathbf{e})\\ \frac{1}{m}\mathbf{Ke} + (\boldsymbol{\mu}_\mathbf{y} + \varepsilon)(\mathbf{y} - \varepsilon\mathbf{e})\end{bmatrix},$$

$$\mathbf{g}_2 = \begin{bmatrix}\frac{1}{m}\mathbf{Ke} + (\boldsymbol{\mu}_\mathbf{y} - \varepsilon)(\mathbf{y} + \varepsilon\mathbf{e})\\ \frac{1}{m}\mathbf{Ke} + (\boldsymbol{\mu}_\mathbf{y} - \varepsilon)(\mathbf{y} - \varepsilon\mathbf{e})\end{bmatrix},$$

and

$$\Xi_1 = \Xi_2 = \Xi = \frac{1}{m}\begin{bmatrix}\mathbf{K} + (\mathbf{y} + \varepsilon\mathbf{e})\mathbf{y}^\top\\ \mathbf{K} + (\mathbf{y} - \varepsilon\mathbf{e})\mathbf{y}^\top\end{bmatrix}\left(I - \frac{1}{m}\mathbf{e}\mathbf{e}^\top\right)\begin{bmatrix}\mathbf{K} + (\mathbf{y} + \varepsilon\mathbf{e})\mathbf{y}^\top\\ \mathbf{K} + (\mathbf{y} - \varepsilon\mathbf{e})\mathbf{y}^\top\end{bmatrix}^\top.$$

Note that in the last equality we have used the fact that $(\mathbf{y}^\top \pm \varepsilon\mathbf{e}^\top)(I - \frac{1}{m}\mathbf{e}\mathbf{e}^\top) = \mathbf{y}^\top(I - \frac{1}{m}\mathbf{e}\mathbf{e}^\top)$.

Using the relations presented in Eq. (18) and taking into account Formulation (7), we can derive the kernel-based formulation for SVR-SOCP, which we denote as SVR-SOCP$_k$, as follows:

$$\begin{aligned}\min_{\mathbf{s},b,\xi}\quad & \frac{1}{2}\mathbf{s}^\top\widetilde{\mathbf{K}}\mathbf{s} + C\xi\\ \text{s.t.}\quad & \mathbf{s}^\top\mathbf{g}_1 + b \geq 1 - \xi + \kappa_1\sqrt{\mathbf{s}^\top\Xi\mathbf{s}},\\ & -(\mathbf{s}^\top\mathbf{g}_2 + b) \geq 1 - \xi + \kappa_2\sqrt{\mathbf{s}^\top\Xi\mathbf{s}},\\ & \xi \geq 0,\end{aligned} \tag{21}$$

where

$$\widetilde{\mathbf{K}} = \begin{bmatrix}\mathbf{K} & \mathbf{K}\\ \mathbf{K} & \mathbf{K}\end{bmatrix} + \begin{bmatrix}\mathbf{y} + \varepsilon\mathbf{e}\\ \mathbf{y} - \varepsilon\mathbf{e}\end{bmatrix}\begin{bmatrix}\mathbf{y} + \varepsilon\mathbf{e}\\ \mathbf{y} - \varepsilon\mathbf{e}\end{bmatrix}^\top \in \Re^{2m\times 2m}.$$

Similar to the linear case, the regression rule for the SVR-SOCP$_k$ method is presented in two steps in order to guarantee the existence of the solution.

Remark 4. Let $(\hat{\mathbf{s}}, \hat{b}, \hat{\xi})$ be a solution of the formulation (21) such that $\hat{\mathbf{s}} \neq 0$. Since such a solution satisfies the constraints of (21), we deduce from this that

$$\varepsilon\hat{\mathbf{s}}^\top\begin{bmatrix}\mathbf{y} + \varepsilon\mathbf{e}\\ \mathbf{y} - \varepsilon\mathbf{e}\end{bmatrix} \geq 1 - \hat{\xi} + \frac{\kappa_1 + \kappa_2}{2}\sqrt{\hat{\mathbf{s}}^\top\Xi\hat{\mathbf{s}}}.$$

Since $\varepsilon > 0$ and $\hat{\xi} \in [0, 1)$ (cf. [24, Lemma 1]), the above relation implies that

$$\begin{aligned}\delta_{ys} &= [(\mathbf{y} + \varepsilon\mathbf{e})^\top\ (\mathbf{y} - \varepsilon\mathbf{e})^\top]\hat{\mathbf{s}}\\ &= \sum_{i=1}^{m}[y_i(\hat{s}_i + \hat{s}_{m+i}) + \varepsilon(\hat{s}_i - \hat{s}_{m+i})] > 0.\end{aligned} \tag{22}$$

Remark 5. Formulation (21) leads to a hyperplane of the form

$$\sum_{j=1}^{2m}\hat{\mathcal{K}}(\hat{\mathbf{x}}, \mathbb{X}_{\bullet j})\hat{s}_j + \hat{b} = 0, \tag{23}$$

with $\hat{\mathbf{x}} = (\mathbf{x}, y) \in \Re^{n+1}$. The expression $\mathbb{X}_{\bullet j}$ denotes the jth column of the matrix $\mathbb{X}$, which is given by

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

and $\hat{\mathcal{K}}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) = \mathcal{K}(\mathbf{x}_1, \mathbf{x}_2) + y_1y_2$. Taking into account this equality, the relation (23) can be rewritten as

$$\begin{aligned}0 &= \sum_{i=1}^{m}\hat{s}_i(\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + y(y_i + \varepsilon)) + \sum_{i=1}^{m}\hat{s}_{m+i}(\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + y(y_i - \varepsilon))\\ &\quad + \hat{b}\\ &= \sum_{i=1}^{m}(\hat{s}_i + \hat{s}_{m+i})\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + \left(\sum_{i=1}^{m}[y_i(\hat{s}_i + \hat{s}_{m+i}) + \varepsilon(\hat{s}_i - \hat{s}_{m+i})]\right)\\ &\quad + \hat{b}.\end{aligned}$$

Assuming that $\hat{\mathbf{s}} \neq 0$, the above Remark implies that $\delta_{ys} > 0$. Then, we can rescale the hyperplane given in (23), and thus get the following regression function with kernel

$$f(\mathbf{x}) = -\frac{1}{\delta_{ys}}\left(\sum_{i=1}^{m}(\hat{s}_i + \hat{s}_{m+i})\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + \hat{b}\right). \tag{24}$$

4. Experiments and discussion

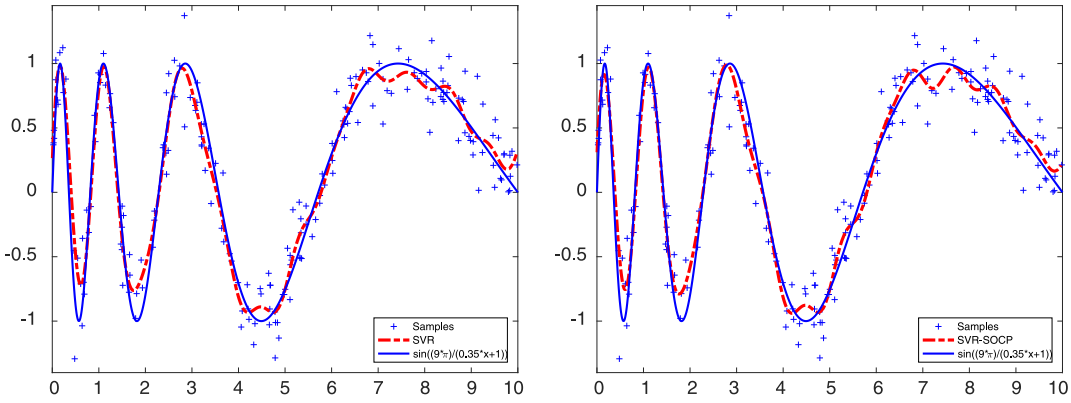
In this section, experimental results on a toy example and ten benchmark datasets are reported. This section is organized as follows: An illustrative example using a one-dimensional toy dataset is presented in Section 4.1. The experimental setting and the benchmark datasets are described in Section 4.2. The performance summary is presented in Section 4.3, including a discussion around the main findings. Finally, the running times are reported in Section 4.4.

4.1. A first illustrative example on synthetic data

We first compare the performance of the proposed formulation (with Gaussian kernel) on a synthetic dataset generated by the function

$$y = \sin\left(\frac{9\pi}{0.35x + 1}\right), \quad x \in [0, 10]. \tag{25}$$

For this, we generated 200 training samples using Eq. (25) with the addition of a Gaussian noise with zero mean and a standard



(a) Estimated function using ε-SVR (b) Results from SVR-SOCP

Fig. 2. An illustrative toy example for ε-SVR and SVR-SOCP.

Table 1
Number of samples and number of features for all data sets.

Dataset	#samples	#features
Triazines	186	58
WBCP	198	32
CPU	209	8
A-MPG	398	25
Housing	506	13
Fires	517	12
Concrete	1080	3
WQR	1599	11
Quake	2178	3
SML2010	4135	22
Abalone	4177	10
Parkinsons	5875	22

deviation of 0.2, and 1000 test samples without any noise. We used a variable x uniformly distributed over the interval [0,10]. Fig. 2 illustrates the estimated function obtained by ε-SVR [11] and the proposed SVR-SOCP for this synthetic dataset (dashed lines). The solid line represents Eq. (25).

The root-mean-square error (RMSE) values on the test set for ε-SVR and SVR-SOCP are 0.1060 and 0.0991, respectively. This result confirms that our proposal has a better predictive performance compared to ε-SVR, demonstrating the virtues of the proposed robust framework.

4.2. Datasets and experimental setting

We applied the proposed and alternative approaches for regression (linear regression, ε-SVR in its linear and kernel-based form, and the proposed SVR-SOCP in its linear and kernel-based form) to twelve benchmark datasets from the UCI Repository [3]: Triazines, Wisconsin Breast Cancer Prognosis (WBCP), Relative CPU Performance (CPU), Auto MPG (A-MPG), Boston Housing (Housing), Forest Fires (Fires), Concrete Compressive Strength (Concrete), Wine Quality - Red (WQR), Quake, SML2010, Abalone and Parkinsons. Table 1 summarizes the relevant information for each benchmark data set:

For model evaluation we used a two-level (nested) cross-validation strategy: training and test subsets were obtained using a 10-fold CV (outer loop), and the training subset was split further into training and validation subsets in order to find the right hyperparameter setting. The final regression model was then performed with the full training subset from the outer loop for the best parameters, and the performance was computed by averaging the ten test results, whose samples remained unseen during the hyperparameter selection procedure. The average RMSE as the main performance metric. This methodology is described as a flow chart in Fig. 3.

Data pre-processing tasks, such as missing-value handling, data normalization, and feature selection, are very important steps in the process of knowledge discovery, which are usually neglected by machine learning researchers. Since machine learning methods are highly susceptible to inconsistencies and noise, the raw data needs to be pre-processed to ease the modeling process [16]. Additionally, raw data usually contains much irrelevant and redundant information, which deteriorates the predictive performance of machine learning methods. A low dimensional data representation not only reduces the risk of overfitting, but it also leads to a better understanding of the process that generated the data, reduces data collection costs, and speeds up the training process [15]. This step is therefore included in the flow chart.

For SVM approaches, we explored the following set of values for parameters C and σ (kernel methods only) using grid search: $\{2^{-7}, 2^{-6}, \ldots, 2^{0}, \ldots, 2^{6}, 2^{7}\}$, and the following values for parameter ε: $\{0.1, 0.2, 0.3, \ldots, 0.8, 0.9\}$. For the proposed method we studied the following values of $\eta_1 = \eta_2 \in \{0.2, 0.4, 0.6, 0.8\}$. These sets of parameters have been used in previous SVM studies for classification (see e.g. [24]).

Regarding model implementation, we used Matlab's fitlm function for linear regression, LIBSVM toolbox [6] for ε-SVR, and the SeDuMi toolbox for the proposed SOCP method [30]. All experiments were performed on an HP Envy dv6 with 16 GB RAM, 750 GB SSD, an Intel Core Processor i7-2620M (2.70 GHz), and using Matlab 2014a and Microsoft Windows 8.1 OS (64-bits).

4.3. Performance summary and discussion

Table 2 summarizes the results obtained from the validation procedure for all regression approaches and for all datasets using RMSE (the average of the ten folds). The standard deviation for each mean RMSE value is presented between parentheses. The best performance among all the methods in terms of each metric is highlighted in bold type. Approaches that are significantly worse than the best at a 5% level are highlighted with an asterisk. Student t-tests were used for comparing the method with the lowest RMSE with the remaining ones. The results using mean absolute percentage error (MAPE) and mean absolute error (MAE) are presented in Appendix B.

It can be observed in Table 2 that no single method outperformed the others in all the experiments, although the pro-

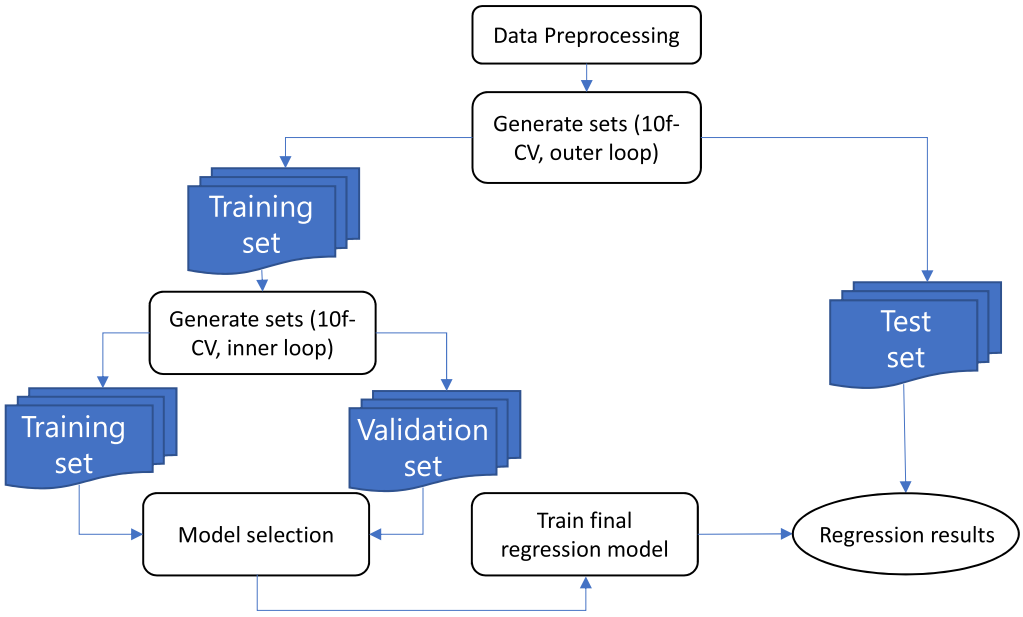


Fig. 3. Flow chart for the model evaluation procedure.

Table 2
Performance summary in terms of RMSE for various regression approaches. All datasets.

Dataset	lin. reg.	ε-SVR$_l$	SVR-SOCP$_l$	ε-SVR$_k$	SVR-SOCP$_k$
Triazines	0.1572	0.1469	0.1414	0.1394	**0.1378**
	(0.1064)	(0.0880)	(0.0827)	(0.0911)	(0.0895)
WBCP	0.2725	0.2578	0.2545	0.2593	**0.2541**
	(0.1281)	(0.1243)	(0.1257)	(0.1246)	(0.1273)
CPU	0.0628	0.0672*	0.0658	0.0300	**0.0271**
	(0.0577)	(0.0482)	(0.0582)	(0.0588)	(0.0352)
A-MPG	0.0848	0.0853	0.0843	**0.0775**	0.0781
	(0.0458)	(0.0497)	(0.0463)	(0.0478)	(0.0513)
Housing	0.2178	0.2173	0.2167	0.1504	**0.1338**
	(0.1750)	(0.1804)	(0.3874)	(0.0858)	(0.0930)
Fires	0.1174	0.1394	**0.1161**	0.1195	0.1168
	(0.1575)	(0.1524)	(0.1582)	(0.1520)	(0.1586)
Concrete	0.2612*	0.2615*	0.2613*	0.1425	**0.1369**
	(0.1019)	(0.1052)	(0.1036)	(0.0789)	(0.0741)
WQR	0.6510	0.6514	0.6510	**0.6219**	0.6240
	(0.2427)	(0.2485)	(0.2452)	(0.2730)	(0.2883)
Quake	**0.1892**	0.2045	0.1893	0.1904	0.1896
	(0.0548)	(0.0595)	(0.0550)	(0.0568)	(0.0577)
SML2010	**0.0039**	0.0453	0.0050	0.0458	0.0103
	(0.0029)	(0.0134)	(0.0027)	(0.0143)	(0.0051)
Abalone	0.0791	0.0817	0.0789	**0.0761**	0.0763
	(0.0243)	(0.0230)	(0.0196)	(0.0191)	(0.0216)
Parkinsons	0.4351*	0.4386*	0.4383*	0.2385	**0.2309**
	(0.0800)	(0.0872)	(0.0854	(0.0770)	(0.0567)

posed SVR-SOCP$_k$ method achieves best results in six of the twelve datasets, and it is never significantly worse than the best approach. The proposed method in its linear form shows the best performance among the linear strategies.

We assessed the overall performance and robustness of each method by performing the analysis procedure proposed in [13], in which the sum of the accuracy ratios is computed for different classification methods in order to summarize the results in a single value. For regression, we can extend this strategy for error metrics as follows: the RMSE ratio for method a and dataset i is

$$RMSERatio_i(a) = \frac{RMSE(a)}{min_j\ RMSE(j)}, \tag{26}$$

where $RMSE(j)$ is the root-mean-square error for technique j when trained over dataset i. The smaller the value of $RMSERatio_i(a)$, the better the performance of method a in dataset i. The best method a^* in i will have $RMSERatio_i(a^*) = 1$. The metric $\Sigma_i RMSERatio_i(a)$ represents a good measure of overall performance and robustness for a method a, and the smaller its value, the better the performance. Fig. 4 presents the distribution of $RMSERatio_i(a)$ for all five methods and all datasets.

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

Method	Mean Rank	Mean RMSE	p value	$\alpha/(k-i)$	Action
SVR-SOCP$_k$	1.66	0.168	–	–	not reject
ε-SVR$_k$	2.50	0.174	0.1967	0.0500	not reject
SVR-SOCP$_l$	2.96	0.209	0.0454	0.0250	not reject
lin. reg.	3.38	0.211	0.0081	0.0167	reject
ε-SVR$_l$	4.50	0.216	<0.0001	0.0125	reject

In Fig. 4, we observe that SVR-SOCP$_k$ has the best average performance, being very close to the optimal performance measure of 12. We also observe that the kernel-based SVM approaches always have better overall performance than their linear counterparts. The gain in using these nonlinear approaches is noteworthy in comparison with linear methods.

In order to further analyze the performance of the various regression methods, a statistical analysis is performed based on the information provided in Table 2. As suggested in Demšar [10], the Holm's test is used to assess whether or not one strategy outperforms the others in terms of RMSE. This strategy performs pairwise comparisons between each strategy and the one with the best performance (lowest RMSE). The result for the Holm's test is presented in Table 3 for the various regression strategies.

It can be concluded from Table 3 that our proposal SVR-SOCP$_k$ has the best average rank (1.66), statistically outperforming linear regression and ε-SVR$_l$ (p value below the threshold defined by the Holm's test). The ε-SVR$_k$ and SVR-SOCP$_l$ methods are not statistically worse than the proposed robust approach in its kernel-based form.

4.4. Running times

Next, the training times are compared for each technique. The average running time using 10-fold crossvalidation is computed

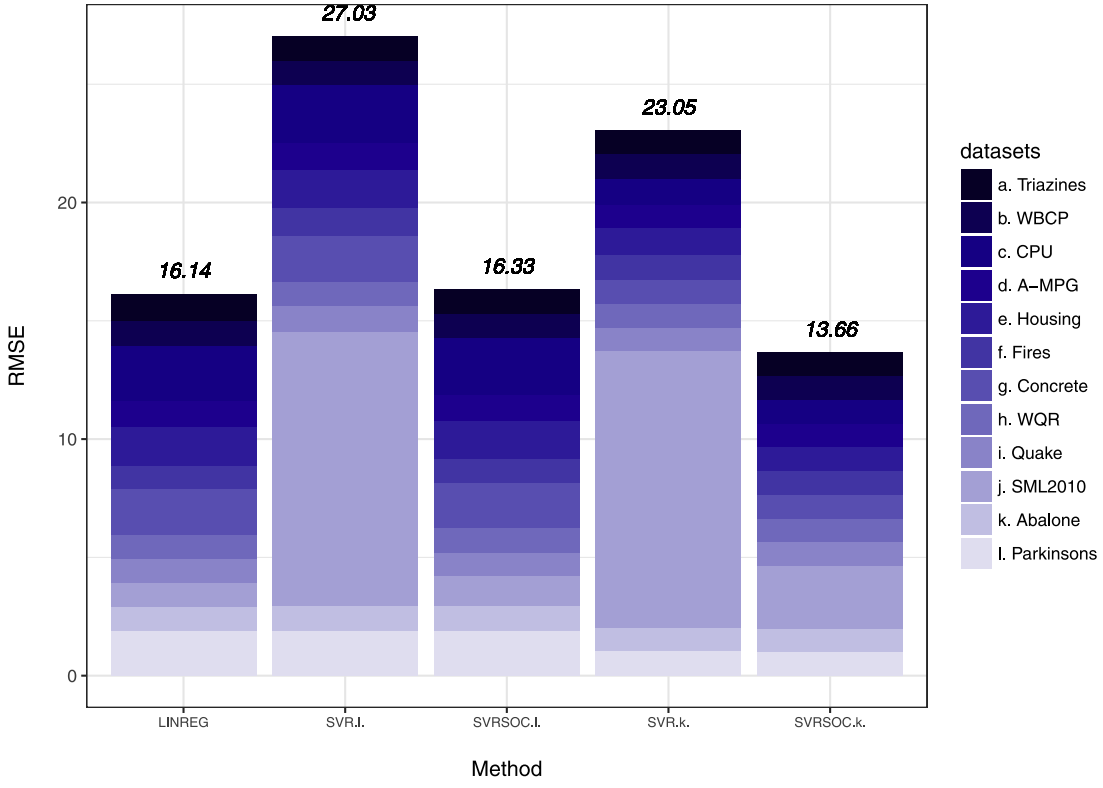


Fig. 4. Sum of RMSE ratios for all methods.

Table 4
Average running times, in seconds, for all methods and datasets. Approaches highlighted with *, †, and ‡ were implemented in the LIBSVM, SeDuMi, and Matlab's Quadprog solvers, respectively.

Dataset	lin. reg.	ε-SVR$_l^*$	ε-SVR$_k^*$	ε-SVR$_l^\dagger$	RH-SVR$_k^\ddagger$	SVR-SOCP$_l^\dagger$	SVR-SOCP$_k^\dagger$
Triazines	0″.1234	0″.0328	0".0141	1″.5391	0″.9063	1″.4000	0″.4281
WBCP	0″.0969	0″.6453	0″.0375	1″.0078	1″.0000	1″.3141	0″.2813
CPU	0″.0766	0″.0125	0″.0063	0″.8266	2″.0625	0″.2938	0″.3719
A-MPG	0″.0953	0″.0516	0″.0266	1″.3391	3″.6094	0″.7109	0″.7641
Housing	0″.0844	0″.1109	0″.0516	0″.9563	5″.6250	0″.3031	1″.1875
Fires	0″.0781	0″.7688	0″.0063	0″.9719	18″.734	0″.3156	1″.5891
Concrete	0″.0953	0″.3281	0″.1453	0″.8656	19″.578	0″.2938	4″.7281
WQR	0″.0813	0″.4844	0″.4094	1″.0109	65″.500	0″.2953	23″.242
Quake	0″.0828	0″.4313	0″.3328	1″.1578	63″.234	0″.3109	60″.213
SML2010	0″.2600	0″.1400	1″.5700	32″.750	351″.80	0″.5800	403″.60
Abalone	0″.3500	0″.3625	0″.3094	2″.191	272″.91	0″.3438	494″.12
Parkinsons	0″.3100	3″.0400	3″.5400	26″.340	834″.45	0″.5900	1117″.42

and presented in Table 4 with its respective standard deviation in parentheses. The best configuration of hyperparameters obtained with the model selection procedure is considered. The kernel version for the RH-SVR method is also included in this analysis. Although this method was not included in the performance analysis since it is similar to ε-SVR$_k$, it represents a relevant comparison to SVR-SOCP$_k$ when analyzing complexity and running times.

It can be observed in Table 4 that the training times are tractable, being below 1 or 2 s in most cases. RH-SVR$_k$ and the proposed SVR-SOCP$_k$ (sixth and last columns, respectively) have the largest training times, mainly because the potentially large kernel matrix that needs to be computed, and the use of a generic solver, such as SeDuMi or QP. To illustrate the influence of using a generic solver in contrast to a highly-optimized, specially-tailored optimization solution such as LIBSVM, we implemented ε-SVR$_l$ using both LIBSVM (column 2) and SeDuMi (column 4). It can be observed that SVR-SOCP$_l$ (column 5) has faster training times in general compared with ε-SVR$_l$ when the same solver (SeDuMi) is used. It can be concluded that a specially-tailored optimization strategy would be extremely useful for making SVR-SOCP$_k$ scalable for large datasets.

5. Conclusions and future work

In this paper, we present a novel SVM method for regression based on second-order cone programming formulations. Our work extends the ideas of RH-SVR [5], a maximum-margin that separates the two reduced convex hulls that result from shifting the target variable up and down by a parameter ε [5]. Our method provides a robust framework in which the margin between two ellipsoids that represent each shifted pattern is maximized. The method is extended as a kernel method in order to construct nonlinear regression functions, which led to best overall performance in ten datasets, compared to linear regression and ε-SVR.

Empirically, we observed that the best performance, in average, was achieved by our proposal in its kernel-based form. Although the difference between the best approach and the others is rather small in most datasets, there are four noteworthy cases: kernel-

based methods led important gains in terms of performance for the Relative CPU Performance, Boston Housing, Forest Fires, and Parkinsons datasets.

Several research opportunities for future work were identified. First, the proposed approach can be extremely useful in time series analysis, a regression task with many applications, such as energy load or wind speed forecasting. Secondly, efficient implementations for SOCP optimization in SVM are required in order to deal with big data problems, exploiting the structure of the model instead of using a generic solver such as SeDuMi. Finally, our method can be used in high-dimensional applications, and it can be extended to perform simultaneous feature selection and model estimation. Approaches like KP-SVR [25], for example, penalize the use of features in the regression function by adding an extra regularizer, in order to reduce the risk of overfitting by adequately balancing generalization, sparsity, and model fit.

Acknowledgments

This research was partially funded by the Complex Engineering Systems Institute, ISCI (ICM-FIC: P05-004-F, CONICYT: FB0816), and by Fondecyt projects 1160738 and 1160894. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

Appendix A. Proof for Proposition 3.1

Proof. In order to show that formulations (7) and (13) are equivalent, we analyze their respective Karush–Kuhn–Tucker (KKT) conditions. We note that both formulations are convex problems with a Slater point. Then, the KKT conditions are necessary and sufficient for them (see e.g. [4]). The Lagrangian functions associated with Formulations (8) and (13) are given by

$$\begin{aligned} L(\mathbf{w},\delta,b,\xi,\mathbf{v}_i,t) = \frac{1}{2}(\|\mathbf{w}\|^2+\delta^2)+C\xi \\ -\nu_{10}(\mathbf{w}^\top\boldsymbol{\mu}_\mathbf{x}+\delta(\boldsymbol{\mu}_\mathbf{y}+\varepsilon)+b-1) \\ -\nu_{10}\xi-\kappa_1\mathbf{v}_{11}{}^\top\Sigma^{1/2}\mathbf{w}^* \\ +\nu_{20}(\mathbf{w}^\top\boldsymbol{\mu}_\mathbf{x}+\delta(\boldsymbol{\mu}_\mathbf{y}-\varepsilon)) \\ -\nu_{20}(-b-1+\xi)-\kappa_2\mathbf{v}_{21}{}^\top\Sigma^{1/2}\mathbf{w}^*-t\xi \end{aligned} \tag{26}$$

and

$$\begin{aligned} L(\tilde{\mathbf{w}},\tilde{b},\tilde{\xi},\tilde{\mathbf{v}}_i,\tilde{t}) = \frac{1}{2}\|\tilde{\mathbf{w}}\|^2+\tilde{C}\tilde{\xi}-\tilde{\nu}_{10}(-\tilde{\mathbf{w}}^\top\boldsymbol{\mu}_\mathbf{x}-\tilde{b}+\boldsymbol{\mu}_\mathbf{y}+\tilde{\varepsilon}+\tilde{\xi}) \\ -\kappa_1\tilde{\mathbf{v}}_{11}^\top\Sigma^{1/2}\tilde{\mathbf{w}}^*-\tilde{\nu}_{20}(\tilde{\mathbf{w}}^\top\boldsymbol{\mu}_\mathbf{x}+\tilde{b}-\boldsymbol{\mu}_\mathbf{y}+\tilde{\varepsilon}+\tilde{\xi}) \\ -\kappa_2\tilde{\mathbf{v}}_{21}^\top\Sigma^{1/2}\tilde{\mathbf{w}}^*-\tilde{t}\tilde{\xi}, \end{aligned} \tag{27}$$

respectively, where $\mathbf{v}_i=(\nu_{i0},\mathbf{v}_{i1})$, $\tilde{\mathbf{v}}_i=(\tilde{\nu}_{i0},\tilde{\mathbf{v}}_{i1})\in\Re\times\Re^{n+1}$, for $i=1,2$.

Then, the KKT conditions for Formulation (7) can be derived from (26):

$$\mathbf{w}-\nu_{10}\boldsymbol{\mu}_\mathbf{x}+\nu_{20}\boldsymbol{\mu}_\mathbf{x}-\Sigma_\mathbf{w}^{1/2}(\kappa_1\mathbf{v}_{11}+\kappa_2\mathbf{v}_{21})=0, \tag{28}$$

$$-\nu_{10}+\nu_{20}=0, \tag{29}$$

$$\delta-\nu_{10}(\boldsymbol{\mu}_\mathbf{y}+\varepsilon)+\nu_{20}(\boldsymbol{\mu}_\mathbf{y}-\varepsilon)-\Sigma_\delta^{1/2}(\kappa_1\mathbf{v}_{11}+\kappa_2\mathbf{v}_{21})=0, \tag{30}$$

$$C-\nu_{10}-\nu_{20}-t=0, \tag{31}$$

$$\mathbf{w}^\top\boldsymbol{\mu}_\mathbf{x}+\delta(\boldsymbol{\mu}_\mathbf{y}+\varepsilon)+b-1+\xi\geq\kappa_1\|\Sigma^{1/2}\mathbf{w}^*\|, \tag{32}$$

$$\nu_{10}\geq\|\mathbf{v}_{11}\|, \tag{33}$$

$$-\mathbf{w}^\top\boldsymbol{\mu}_\mathbf{x}-\delta(\boldsymbol{\mu}_\mathbf{y}-\varepsilon)-b-1+\xi\geq\kappa_2\|\Sigma^{1/2}\mathbf{w}^*\|, \tag{34}$$

$$\nu_{20}\geq\|\mathbf{v}_{21}\|, \tag{35}$$

$$t\geq 0,\ \xi\geq 0, \tag{36}$$

$$t\xi=0, \tag{37}$$

$$\nu_{10}(\mathbf{w}^\top\boldsymbol{\mu}_\mathbf{x}+\delta(\boldsymbol{\mu}_\mathbf{y}+\varepsilon)+b-1+\xi)=-\kappa_1\mathbf{v}_{11}^\top\Sigma^{1/2}\mathbf{w}^*, \tag{38}$$

$$\nu_{20}(-\mathbf{w}^\top\boldsymbol{\mu}_\mathbf{x}-\delta(\boldsymbol{\mu}_\mathbf{y}-\varepsilon)-b-1+\xi)=-\kappa_2\mathbf{v}_{21}^\top\Sigma^{1/2}\mathbf{w}^*, \tag{39}$$

where $\Sigma^{1/2}=[\Sigma_\mathbf{w}^{1/2};\Sigma_b^{1/2}]$, with $\Sigma_\mathbf{w}^{1/2}\in\Re^{n\times n+1}$ and $\Sigma_b^{1/2}\in\Re^{1\times n+1}$.

From Remark 1 we have that $\delta>0$. Then, dividing the relations (28)–(36) by δ, and dividing the relations (37)–(39) by δ^2, and using $\tilde{\mathbf{w}}=-\frac{\mathbf{w}}{\delta}$, $\tilde{b}=-\frac{b}{\delta}$, $\tilde{\xi}=\frac{\xi}{\delta}$, $\tilde{\mathbf{v}}_i=\frac{\mathbf{v}_i}{\delta}$, $\tilde{t}=\frac{t}{\delta}$, $\tilde{\varepsilon}=\varepsilon-\frac{1}{\delta}$ and $\tilde{C}=\frac{C}{\delta}$ we obtain the following relations:

$$\tilde{\mathbf{w}}+\tilde{\nu}_{10}\boldsymbol{\mu}_\mathbf{x}-\tilde{\nu}_{20}\boldsymbol{\mu}_\mathbf{x}+\Sigma_\mathbf{w}^{1/2}(\kappa_1\tilde{\mathbf{v}}_{11}+\kappa_2\tilde{\mathbf{v}}_{21})=0, \tag{40}$$

$$\tilde{\nu}_{10}-\tilde{\nu}_{20}=0, \tag{41}$$

$$\tilde{C}-\tilde{\nu}_{10}-\tilde{\nu}_{20}-\tilde{t}=0, \tag{42}$$

$$-\tilde{\mathbf{w}}^\top\boldsymbol{\mu}_\mathbf{x}-\tilde{b}+\boldsymbol{\mu}_\mathbf{y}+\tilde{\varepsilon}+\tilde{\xi}\geq\kappa_1\|\Sigma^{1/2}\tilde{\mathbf{w}}^*\|, \tag{43}$$

$$\tilde{\nu}_{10}\geq\|\tilde{\mathbf{v}}_{11}\|, \tag{44}$$

$$\tilde{\mathbf{w}}^\top\boldsymbol{\mu}_\mathbf{x}+\tilde{b}-\boldsymbol{\mu}_\mathbf{y}+\tilde{\varepsilon}+\tilde{\xi}\geq\kappa_2\|\Sigma^{1/2}\tilde{\mathbf{w}}^*\|, \tag{45}$$

$$\tilde{\nu}_{20}\geq\|\tilde{\mathbf{v}}_{21}\|, \tag{46}$$

$$\tilde{t}\geq 0,\ \tilde{\xi}\geq 0, \tag{47}$$

$$\tilde{t}\tilde{\xi}=0, \tag{48}$$

$$\tilde{\nu}_{10}(-\tilde{\mathbf{w}}^\top\boldsymbol{\mu}_\mathbf{x}-\tilde{b}+\boldsymbol{\mu}_\mathbf{y}+\tilde{\varepsilon}+\tilde{\xi})+\kappa_1\tilde{\mathbf{v}}_{11}^\top\Sigma^{1/2}\tilde{\mathbf{w}}^*=0, \tag{49}$$

$$\tilde{\nu}_{20}(\tilde{\mathbf{w}}^\top\boldsymbol{\mu}_\mathbf{x}+\tilde{b}-\boldsymbol{\mu}_\mathbf{y}+\tilde{\varepsilon}+\tilde{\xi})+\kappa_2\tilde{\mathbf{v}}_{21}^\top\Sigma^{1/2}\tilde{\mathbf{w}}^*=0. \tag{50}$$

These expressions represent the KKT conditions for Formulation (13). On the other hand, by summing the relations (49)–(50), and by using (40), we get

$$\begin{aligned} 0 &= 2\tilde{\nu}_{10}(\tilde{\varepsilon}+\tilde{\xi})+(\tilde{\mathbf{w}}^*)^\top\Sigma^{1/2}(\kappa_1\tilde{\mathbf{v}}_{11}+\kappa_2\tilde{\mathbf{v}}_{21}) \\ &= 2\tilde{\nu}_{10}(\tilde{\varepsilon}+\tilde{\xi})-\tilde{\mathbf{w}}^\top\Sigma_\mathbf{w}^{1/2}(\kappa_1\tilde{\mathbf{v}}_{11}+\kappa_2\tilde{\mathbf{v}}_{21})+ \\ &\quad \Sigma_\delta^{1/2}(\kappa_1\tilde{\mathbf{v}}_{11}+\kappa_2\tilde{\mathbf{v}}_{21}). \end{aligned} \tag{51}$$

Since $\tilde{\mathbf{w}} = -\Sigma_{\mathbf{w}}^{1/2}(\kappa_1\tilde{\mathbf{v}}_{11} + \kappa_2\tilde{\mathbf{v}}_{21})$ (cf. (40)–(41)), from (51) we have

$$\|\tilde{\mathbf{w}}\|^2 + 2\tilde{\nu}_{10}\tilde{\xi} = -2\varepsilon\tilde{\nu}_{10} - \Sigma_{\delta}^{1/2}(\kappa_1\tilde{\mathbf{v}}_{11} + \kappa_2\tilde{\mathbf{v}}_{21}) > 0.$$

Clearly $\tilde{\nu}_{10}$ is different from zero. Otherwise one would have that $\|\tilde{\mathbf{w}}\| = 0$, contradicting our assumption. Hence, we can define

$$\delta = \frac{2\tilde{\nu}_{10}}{1 - 2\tilde{\varepsilon}\tilde{\nu}_{10} - \Sigma_{\delta}^{1/2}(\kappa_1\tilde{\mathbf{v}}_{11} + \kappa_2\tilde{\mathbf{v}}_{21})} > 0. \tag{52}$$

Then, by multiplying the relations (40)–(47) by δ, the relations (48)–(50) by δ^2, and using $\mathbf{w} = -\delta\tilde{\mathbf{w}}$, $b = -\delta\tilde{b}$, $\boldsymbol{\xi} = \delta\tilde{\boldsymbol{\xi}}$, $\mathbf{v}_i = \delta\tilde{\mathbf{v}}_i$, $t = \delta\tilde{t}$, $\varepsilon = \tilde{\varepsilon} + \frac{1}{\delta}$ and $C = \delta\tilde{C}$, we obtain the relations (28)–(29) and (31)–(39). Finally, the relation (30) is obtained from (52), using $\tilde{\mathbf{v}}_i = \frac{\mathbf{v}_i}{\delta}$ and $\nu_{10} = \nu_{20}$. □

Appendix B. Performance summary in terms of MAPE and MAE

Tables B1 and B2.

Table B1
Performance summary in terms of MAPE for different regression approaches. All datasets.

Dataset	lin. reg.	ε-SVR$_l$	SVR-SOCP$_l$	ε-SVR$_k$	SVR-SOCP$_k$
Triazines	0.2844 (0.1324)	0.2728 (0.1297)	0.2728 (0.1181)	0.2649 (0.1267)	0.2576 (0.1217)
WBCP	1.8999 (0.6028)	1.7528 (0.5446)	1.8448 (0.5534)	1.7567 (0.5491)	1.7333 (0.5969)
CPU	0.1530 (0.0817)	0.1085 (0.0596)	0.1027 (0.0816)	0.0209 (0.0404)	0.0209 (0.0288)
A-MPG	0.8635 (0.0532)	0.4948 (0.0507)	0.4439 (0.0520)	0.1898 (0.0430)	0.1882 (0.0303)
Housing	0.6520 (0.2602)	0.6683 (0.2538)	0.6304 (0.2451)	0.4954 (0.2483)	0.4358 (0.2253)
Fires	0.0447 (0.0244)	0.1054 (0.0216)	0.0425 (0.0251)	0.0688 (0.0202)	0.0349 (0.0252)
Concrete	1.7521 (0.7112)	3.6487 (0.7525)	2.9548 (0.7410)	2.2972 (0.2576)	1.0361 (0.3492)
WQR	0.0891 (0.0061)	0.0887 (0.0061)	0.0926 (0.0062)	0.0824 (0.0081)	0.0882 (0.0077)
Quake	0.0249 (0.0009)	0.0238 (0.0011)	0.0244 (0.0010)	0.0246 (0.0010)	0.0245 (0.0010)
SML2010	0.0440 (0.0223)	0.7308 (0.4458)	0.0507 (0.0160)	0.7329 (0.3299)	0.1010 (0.0305)
Abalone	0.1765 (0.0098)	0.1910 (0.0102)	0.1770 (0.0093)	0.1730 (0.0070)	0.1703 (0.006)
Parkinsons	1.4223 (0.2425)	1.4803 (0.2656)	1.4401 (0.2369)	0.7398 (0.1533)	0.7662 (0.1148)

Table B2
Performance summary in terms of MAE for different regression approaches. All datasets.

Dataset	lin. reg.	ε-SVR$_l$	SVR-SOCP$_l$	ε-SVR$_k$	SVR-SOCP$_k$
Triazines	0.1162 (0.0228)	0.1032 (0.0213)	0.1040 (0.0187)	0.1000 (0.0192)	0.0997 (0.0212)
WBCP	0.2267 (0.0283)	0.2153 (0.0309)	0.2132 (0.0316)	0.2185 (0.0311)	0.2113 (0.0311)
CPU	0.0366 (0.0090)	0.0515 (0.0083)	0.0371 (0.0098)	0.0097 (0.0101)	0.0113 (0.0061)
A-MPG	0.0646 (0.0080)	0.0647 (0.0091)	0.0638 (0.0081)	0.0581 (0.0087)	0.0588 (0.0091)
Housing	0.1513 (0.0348)	0.1508 (0.0351)	0.1495 (0.0349)	0.0981 (0.0166)	0.0909 (0.0172)
Fires	0.0367 (0.0125)	0.1090 (0.0126)	0.0341 (0.0132)	0.0318 (0.0103)	0.0341 (0.0134)
Concrete	0.2066 (0.0170)	0.2068 (0.0176)	0.2073 (0.0171)	0.0946 (0.0105)	0.1004 (0.0095)
WQR	0.5038 (0.0292)	0.5005 (0.0288)	0.5051 (0.0293)	0.4479 (0.0422)	0.4786 (0.0428)
Quake	0.1486 (0.0059)	0.1406 (0.0067)	0.1479 (0.0060)	0.1489 (0.0063)	0.148 (0.0061)
Abalone	0.0566 (0.0020)	0.0623 (0.0017)	0.0567 (0.0019)	0.0525 (0.0014)	0.0537 (0.001)
SML2010	0.0024 (0.0001)	0.0377 (0.0019)	0.0029 (0.0001)	0.0380 (0.0013)	0.0062 (0.0003)
Parkinsons	0.3680 (0.0090)	0.3683 (0.0095)	0.3709 (0.0092)	0.1661 (0.0082)	0.1554 (0.0052)

(continued on next page)

References

[1] S. Ali, K.A. Smith-Miles, A meta-learning approach to automatic kernel selection for support vector machines, Neurocomputing 20 (1–3) (2006) 173–186.
[2] F. Alizadeh, D. Goldfarb, Second-order cone programming, Math. Program. 95 (2003) 3–51.
[3] K. Bache, M. Lichman, UCI machine learning repository, 2013.
[4] D. Bertsekas, Nonlinear Programming, 2nd, Athena Scientific, 1999.
[5] J. Bi, P. Bennett, A geometric approach to support vector regression, Neurocomputing 55 (2003) 78–108.
[6] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[7] C.-C. Chuang, S.-F. Su, J.-T. Jeng, C.-C. Hsiao, Robust support vector regression networks for function approximation with outliers, IEEE Trans. Neural Netw. 13 (6) (2002) 1322–1330.
[8] M. Claesen, F.D. Smet, J. Suykens, B.D. Moora, A robust ensemble approach to learn from positive and unlabeled data using svm base models, Neurocomputing 160 (2015) 73–84.
[9] R. Debnath, M. Muramatsu, H. Takahashi, An efficient support vector machine learning method with second-order cone programming for large-scale problems, Appl. Intell. 23 (3) (2005) 219–239.
[10] J. Demšar, Statistical comparisons of classifiers over multiple data set, J. Mach. Learn. Res. 7 (2006) 1–30.
[11] H. Drucker, C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, in: Advances in Neural Information Processing Systems (NIPS), Vol. 9, MIT Press, 1997, pp. 155–161.
[12] G.-F. Fan, L.-L. Peng, W.-C. Hong, F. Sun, Electric load forecasting by the svr model with differential empirical mode decomposition and auto regression, Neurocomputing 173 (2016) 958–970.
[13] X. Geng, D.-C. Zhan, Z.-H. Zhou, Supervised nonlinear dimensionality reduction for visualization and classification, IEEE Trans. Syst., Man, Cybern., Part B: Cybern. 35 (6) (2005) 1098–1107.
[14] M. Gocic, S. Shamshirband, Z. Razak, D. Petković, S. Ch, S. Trajkovic, Long-term precipitation analysis and estimation of precipitation concentration index using three support vector machine methods, Adv. Meteorol. 2016 (2016) 1–11.
[15] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction, Foundations and Applications, Springer, Berlin, 2006.
[16] J. Han J.and Pei, M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2011.
[17] G. Huang, S. Song, C. Wu, K. You, Robust support vector regression for uncertain input and output data, IEEE Trans. Neural Netw. Learn. Syst. 23 (11) (2012) 1690–1700.
[18] A. Jahangirzadeh, S. Shamshirband, S. Aghabozorgi, S. Akib, H. Basser, N.B. Anuar, M.L.M. Kiah, A cooperative expert based support vector regression (co-esvr) system to determine collar dimensions around bridge pier, Neurocomputing 140 (2014) 172–184.
[19] S. Jović, A.S. Danesh, E. Younesi, O. Aničić, D. Petković, Forecasting of underactuated robotic finger contact forces by support vector regression methodology, Int. J. Pattern Recognit. Artif. Intell. 30 (2016) 1–11.
[20] G. Lanckriet, L. Ghaoui, C. Bhattacharyya, M. Jordan, A robust minimax approach to classification, J. Mach. Learn. Res. 3 (2003) 555–582.
[21] M.-W. Li, D.-F. Han, W.-L. Wang, Vessel traffic flow forecasting by rsvr with chaotic cloud simulated annealing genetic algorithm and kpca, Neurocomputing 157 (2015) 243–255.
[22] M. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, Linear Algebra Appl. 284 (1998) 193–228.
[23] J. López, S. Maldonado, Group-penalized feature selection and robust twin svm classification via second-order cone programming, Neurocomputing 235 (2017) 112–121.
[24] S. Maldonado, J. López, Alternative second-order cone programming formulations for support vector classification, Inf. Sci. 268 (2014) 328–341.
[25] S. Maldonado, R. Weber, Feature selection for support vector regression via kernel penalization, in: Proceedings of the 2010 International Joint Conference on Neural Networks, Barcelona, Spain, 2010, pp. 1973–1979.
[26] J. Mercer, Functions of positive and negative type, and their connection with the theory of integral equations, Philos. Trans. R. Soc. Lond. 209 (1909) 415–446.

[27] 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.
[28] D. Petković, S. Shamshirband, H. Saboohi, T. Ang, N. Anuar, N. Pablović, Support vector regression methodology for prediction of input displacement of adaptive compliant robotic gripper, Appl. Intell. 41 (2014) 887–896.
[29] B. Schölkopf, A.J. Smola., Learning with Kernels, MIT Press, 2002.
[30] J. Sturm, Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, Opt. Methods Softw. 11 (12) (1999) 625–653. Special issue on Interior Point Methods (CD supplement with software)
[31] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[32] M.H. Zangooei, J. Habibi, R. Alizadehsani, Disease diagnosis with a hybrid method svr using nsga-ii, Neurocomputing 136 (2014) 14–29.
[33] P. Zhong, M. Fukushima, Second-order cone programming formulations for robust multiclass classification, Neural Comput. 19 (2007) 258–282.

Sebastián Maldonado received his B.S. and M.S. degree from the University of Chile, in 2007, and his Ph.D. degree from the University of Chile, in 2011. He is currently Associate Professor at the School of Engineering and Applied Sciences, Universidad de los Andes, Santiago, Chile. His research interests include statistical learning, data mining and business analytics. Sebastián Maldonado has published more than 60 scientific contributions including more than 30 Thomson Reuters' ISI papers in the last five years.

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