Neural Comput & Applic (2013) 23:175–185
DOI 10.1007/s00521-012-0924-3

ORIGINAL ARTICLE

An ε-twin support vector machine for regression

Yuan-Hai Shao · Chun-Hua Zhang · Zhi-Min Yang · Ling Jing · Nai-Yang Deng

Received: 4 September 2011 / Accepted: 19 March 2012 / Published online: 8 April 2012

Abstract This study proposes a new regressor—ε-twin support vector regression (ε-TSVR) based on TSVR. ε-TSVR determines a pair of ε-insensitive proximal functions by solving two related SVM-type problems. Different form only empirical risk minimization is implemented in TSVR, the structural risk minimization principle is implemented by introducing the regularization term in primal problems of our ε-TSVR, yielding the dual problems to be stable positive definite quadratic programming problems, so can improve the performance of regression. In addition, the successive overrelaxation technique is used to solve the optimization problems to speed up the training procedure. Experimental results for both artificial and real datasets show that, compared with the popular ε-SVR, LS-SVR and TSVR, our ε-TSVR has remarkable improvement of generalization performance with short training time.

Keywords Machine learning · Support vector machines · Regression · Twin support vector machine · Successive overrelaxation

Y.-H. Shao · Z.-M. Yang
Zhijiang College, Zhejiang University of Technology, Hangzhou 310024, People's Republic of China
e-mail: shaoyuanhai21@163.com

Z.-M. Yang
e-mail: yzm9966@126.com

C.-H. Zhang (✉)
Department of Mathematics, Information School, Renmin University of China, Beijing 100872, People's Republic of China
e-mail: zhangchunhua@ruc.edu.cn

L. Jing · N.-Y. Deng (✉)
College of Science China Agricultural University, Beijing 100083, People's Republic of China
e-mail: dengnaiyang@cau.edu.cn

1 Introduction

Support vector machines (SVMs), being computationally powerful tools for pattern classification and regression [1–3], have been successfully applied to a variety of real-world problems [5–7]. Regards to support vector classification, there exist some classical methods such as *C*-support vector classification (*C*-SVC) [2, 4], least square support vector classification (LS-SVC) [8], etc. The basic idea of all of these classifiers is to find the decision function by maximizing the margin between two parallel hyperplanes. Recently, some nonparallel hyperplane classifiers such as generalized eigenvalue proximal support vector machine (GEPSVM) and twin support vector machine (TWSVM) were proposed in [9, 10]. TWSVM seeks two nonparallel proximal hyperplanes such that each hyperplane is closest to one of two classes and as far as possible from the other class. A fundamental difference between TWSVM and *C*-SVC is that TWSVM solves two smaller sized quadratic programming problems (QPPs), whereas *C*-SVC solves one larger QPP. This makes TWSVM work faster than *C*-SVC. In addition, TWSVM is excellent at dealing with the "Cross Planes" dataset. Thus, the methods of constructing the nonparallel hyperplanes have been studied extensively [11–17].

Regards to the support vector regression, there exist some corresponding methods such as ε-support vector regression (ε-SVR), least square support vector regression (LS-SVR) [8, 18], etc. For linear ε-SVR, its primal problem can be understood in the following way: finds a linear function $f(x)$ such that, on the one hand, more training samples locate in the ε-intensive tube between $f(x) - \varepsilon$ and $f(x) + \varepsilon$, on the other hand, the function $f(x)$ is as flat as possible, leading to introduce the regularization term. Thus, the structural risk minimization principle is implemented.

Linear LS-SVR works in a similar way. Different from ε-SVR and LS-SVR, Peng [19] proposed a regressor in the spirit of TWSVM, termed as twin support vector regression (TSVR). The formulation of TSVR is in the spirit of TWSVM via two nonparallel planes and also solves two smaller sized QPPs, whereas the classical SVR solves one larger QPP. Experimental results in [19] showed the effectiveness of TSVR over ε-SVR in terms of both generalization performance and training time. Thus, TSVR has been studied in [20–24].

It is well known that one significant advantage of ε-SVR is the implementation of the structural risk minimization principle [25]. However, only the empirical risk is considered in the primal problems of TSVR. In addition, we noticed that the matrix $(G^\top G)^{-1}$ appears in the dual problems derived from the primal problems of TSVR. So, the extra condition that $G^\top G$ is nonsingular, must be assumed. This is not perfect from the theoretical point of view although it has been handled by modifying the dual problems technically and elegantly.

In this study, we proposes another regressor in the spirit of TWSVM, named ε-twin support vector regression (ε-TSVR). Similar to TSVR, linear ε-TSVR constructs two nonparallel ε-insensitive proximal functions by solving two smaller QPPs. However, there are differences as the following: (1) The main difference is that, in the primal problems of TSVR, the empirical risk is minimized, whereas, in our ε-TSVR, the structural risk is minimized by adding a regularization term. Similar to ε-SVR, the minimization of this term requires that two functions are as flat as possible. (2) The dual problems of our primal problems can be derived without any extra assumption and need not to be modified any more. We think that our method is more rigorous and complete than TSVR from theoretical point of view. (3) In order to shorten training time, an effective successive overrelaxation (SOR) technique is applied to our ε-TSVR. The preliminary experiments show that our ε-TSVR is not only faster, but also has better generalization.

This study is organized as follows. Section 2 briefly dwells on the standard ε-SVR, LS-SVR, and TSVR. Section 3 proposes our ε-TSVR, and the SOR technique is used to solve the optimization problems in ε-TSVR. Experimental results are described in Sect. 4 and concluding remarks are given in Sect. 5.

2 Background

Consider the following regression problem, suppose that the training set is denoted by (A, Y), where A is a $l \times n$ matrix and the i-th row $A_i \in R^n$ represents the i-th training sample, $i = 1, 2, \ldots, l$. Let $Y = (y_1; y_2; \ldots; y_l)$ denotes the response vector of training sample, where $y_i \in R$. Here, we briefly describe some methods that are closely related to our method, including ε-SVR, LS-SVR and TSVR. For simplicity, we only consider the linear regression problems.

2.1 ε-Support vector regression

Linear ε-SVR [3, 25–27] searches for an optimal linear regression function

$$f(x) = w^\top x + b, \tag{1}$$

where $w \in R^n$ and $b \in R$. To measure the empirical risk, the ε-intensive loss function

$$R^{\varepsilon}_{\text{emp}}[f] = \frac{1}{l}\sum_{i=1}^{l} |y_i - f(x_i)|_{\varepsilon}, \tag{2}$$

is used, where $|y_i - f(x_i)|_{\varepsilon} = \max\{0, |y_i - f(x_i)| - \varepsilon\}$. By introducing the regularization term $\frac{1}{2}||w||^2$ and the slack variables ξ, ξ^*, the primal problem of ε-SVR can be expressed as

$$\begin{aligned} \min_{w,b,\xi,\xi^*} \quad & \tfrac{1}{2}||w||^2 + C(e^\top \xi + e^\top \xi^*), \\ \text{s.t.} \quad & Y - (Aw + eb) \le \varepsilon e + \xi, \quad \xi \ge 0, \\ & (Aw + eb) - Y \le \varepsilon e + \xi^*, \quad \xi^* \ge 0, \end{aligned} \tag{3}$$

where $C > 0$ is a parameter determining the trade-off between the empirical risk and the regularization term. Note that a small $\frac{1}{2}\|w\|^2$ corresponds to the linear function (1) that is flat [25, 26]. In the case of support vector classification, the structural risk minimization principle is implemented by this regularization term $\frac{1}{2}\|w\|^2$. In the case of support vector regression, this term is also added to minimize the structural risk.

2.2 Least squares support vector regression

Similar to ε-SVR, linear LS-SVR [8, 18] also searches for an optimal linear regression function

$$f(x) = w^\top x + b, \tag{4}$$

and the following loss function is used to measure the empirical risk

$$R_{\text{emp}}[f] = \frac{1}{l}\sum_{i=1}^{l} (y_i - f(x_i))^2. \tag{5}$$

By adding the regularization term $\frac{1}{2}||w||^2$ and the slack variable ξ, the primal problem can be expressed as

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}||w||^2 + \tfrac{C}{2}\xi^\top \xi, \\ \text{s.t.} \quad & Y - (Aw + eb) = \xi, \end{aligned} \tag{6}$$

where $C > 0$ is a parameter.

2.3 Twin support vector regression

Different from ε-SVR and LS-SVR, linear TSVR seeks a pair of up-bound and down-bound functions

$$f_1(x) = w_1^\top x + b_1 \quad \text{and} \quad f_2(x) = w_2^\top x + b_2, \tag{7}$$

where $w_1 \in R^n$, $w_2 \in R^n$, $b_1 \in R$ and $b_2 \in R$. Here, the empirical risks are measured by:

$$R_{\text{emp}}^{\varepsilon_1}[f_1] = \sum_{i=1}^{l} \max\{0, (y_i - f_1(x_i) - \varepsilon_1)^2\} + c_1 \sum_{i=1}^{l} \max\{0, -(y_i - f_1(x_i) - \varepsilon_1)\} \tag{8}$$

and

$$R_{\text{emp}}^{\varepsilon_2}[f_2] = \sum_{i=1}^{l} \max\{0, (f_2(x_i) - y_i - \varepsilon_2)^2\} + c_2 \sum_{i=1}^{l} \max\{0, -(f_2(x_i) - y_i - \varepsilon_2)\} \tag{9}$$

where $c_1 > 0$ and $c_2 > 0$ are parameters. By introducing the slack variable ξ, ξ^*, η and η^*, the primal problems are expressed as

$$\begin{aligned} \min_{w_1,b_1,\xi,\xi^*} \quad & \tfrac{1}{2}\xi^{*\top}\xi^* + c_1 e^\top \xi, \\ \text{s.t.} \quad & Y - (Aw_1 + eb_1) \geq \varepsilon_1 e - \xi, \quad \xi \geq 0, \\ & Y - (Aw_1 + eb_1) = \varepsilon_1 e + \xi^*, \end{aligned} \tag{10}$$

and

$$\begin{aligned} \min_{w_2,b_2,\eta,\eta^*} \quad & \tfrac{1}{2}\eta^{*\top}\eta^* + c_2 e^\top \eta, \\ \text{s.t.} \quad & (Aw_2 + eb_2) - Y \geq \varepsilon_2 e - \eta, \quad \eta \geq 0, \\ & (Aw_2 + eb_2) - Y = \varepsilon_2 e + \eta^*. \end{aligned} \tag{11}$$

Their dual problems are

$$\begin{aligned} \max_{\alpha} \quad & -\tfrac{1}{2}\alpha^\top G(G^\top G)^{-1}G^\top \alpha + f^\top G(G^\top G)^{-1}G^\top \alpha - f^\top \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 e. \end{aligned} \tag{12}$$

and

$$\begin{aligned} \max_{\gamma} \quad & -\tfrac{1}{2}\gamma^\top G(G^\top G)^{-1}G^\top \gamma - h^\top G(G^\top G)^{-1}G^\top \gamma + h^\top \gamma \\ \text{s.t.} \quad & 0 \leq \gamma \leq c_2 e. \end{aligned} \tag{13}$$

respectively when $G^\top G$ is positive definite, where $G = [A \quad e]$, $f = Y - e\varepsilon_1$ and $h = Y + e\varepsilon_2$.

In order to deal with the case when $G^\top G$ is singular and avoid the possible ill-conditioning, the above dual problems are modified artificially as:

$$\begin{aligned} \max_{\alpha} \quad & -\tfrac{1}{2}\alpha^\top G(G^\top G + \sigma I)^{-1}G^\top \alpha + f^\top G(G^\top G + \sigma I)^{-1}G^\top \alpha - f^\top \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 e. \end{aligned} \tag{14}$$

and

$$\begin{aligned} \max_{\gamma} \quad & -\tfrac{1}{2}\gamma^\top G(G^\top G + \sigma I)^{-1}G^\top \gamma - h^\top G(G^\top G + \sigma I)^{-1}G^\top \gamma + h^\top \gamma \\ \text{s.t.} \quad & 0 \leq \gamma \leq c_2 e. \end{aligned} \tag{15}$$

by adding a term σI, where σ is a small positive scalar, and I is an identity matrix of appropriate dimensions.

The augmented vectors can be obtained from the solution α and γ of (14) and (15) by

$$v_1 = (G^\top G + \sigma I)^{-1}G^\top (f - \alpha), \tag{16}$$

and

$$v_2 = (G^\top G + \sigma I)^{-1}G^\top (h + \gamma), \tag{17}$$

where $v_1 = [w_1 \quad b_1]$, $v_2 = [w_2 \quad b_2]$.

3 ε-Twin support vector regression

3.1 Linear ε-TSVR

Following the idea of TWSVM and TSVR, in this section, we introduce a novel approach that we have termed as ε-twin support vector regression (ε-TSVR). As mentioned earlier, ε-TSVR also finds two ε-insensitive proximal linear functions:

$$f_1(x) = w_1^\top x + b_1 \quad \text{and} \quad f_2(x) = w_2^\top x + b_2. \tag{18}$$

Here, the empirical risks are measured by:

$$R_{\text{emp}}^{\varepsilon_1}[f_1] = \sum_{i=1}^{l} \max\{0, (y_i - f_1(x_i))^2\} + c_1 \sum_{i=1}^{l} \max\{0, -(y_i - f_1(x_i) + \varepsilon_1)\} \tag{19}$$

and

$$R_{\text{emp}}^{\varepsilon_2}[f_2] = \sum_{i=1}^{l} \max\{0, (f_2(x_i) - y_i)^2\} + c_2 \sum_{i=1}^{l} \max\{0, -(f_2(x_i) - y_i + \varepsilon_2)\} \tag{20}$$

where $c_1 > 0$ and $c_2 > 0$ are parameters, and $\sum_{i=1}^{l} \max\{0, -(y_i - f_1(x_i) + \varepsilon_1)\}$ and $\sum_{i=1}^{l} \max\{0, -(f_2(x_i) - y_i + \varepsilon_2)\}$ are the one-side ε-insensitive loss function [25].

By introducing the regularization terms $\frac{1}{2}(w_1^\top w_1 + b_1^2)$ and $\frac{1}{2}(w_2^\top w_2 + b_1^2)$, the slack variables ξ, ξ^*, η and η^*, the primal problems can be expressed as

$$\begin{aligned} \min_{w_1,b_1,\xi,\xi^*} \quad & \tfrac{1}{2}c_3(w_1^\top w_1 + b_1^2) + \tfrac{1}{2}\xi^{*\top}\xi^* + c_1 e^\top \xi, \\ \text{s.t.} \quad & Y - (Aw_1 + eb_1) = \xi^*, \\ & Y - (Aw_1 + eb_1) \geq -\varepsilon_1 e - \xi, \quad \xi \geq 0, \end{aligned} \tag{21}$$

and

$$\begin{aligned} \min_{w_2,b_2,\eta,\eta^*} \quad & \tfrac{1}{2}c_4(w_2^\top w_2 + b_2^2) + \tfrac{1}{2}\eta^{*\top}\eta^* + c_2 e^\top \eta, \\ \text{s.t.} \quad & (Aw_2 + eb_2) - Y = \eta^*, \\ & (Aw_2 + eb_2) - Y \geq -\varepsilon_2 e - \eta, \quad \eta \geq 0, \end{aligned} \tag{22}$$

where c_1, c_2, c_1, ε_1 and ε_2 are positive parameters.

Now, we discuss the difference between the primal problems of TSVR and our ε-TSVR, by comparing problem (10) and problem (21).

1. The main difference is that there is an extra regularization term $\frac{1}{2}c_3(\|w_1\|^2 + b_1^2)$ in (21). Now, we show that the structural risk is minimized in (21) due to this term. Remind the primal problem of ε-SVR, where the structural risk minimization is implemented by minimizing the regularization term $\frac{1}{2}\|w\|^2$ and a small $\|w\|^2$ corresponds to the linear function (1) that is flat. In fact, by introducing the transformation from R^n to R^{n+1}: $\mathbf{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$, the functions $f_1(x) = w_1^\top x + b_1$ can be expressed as:

$$\mathbf{w_1}^\top \mathbf{x} = [w_1^\top, b_1] \begin{bmatrix} x \\ 1 \end{bmatrix} \tag{23}$$

showing that the flatness of the linear function $f_1(x)$ in the $\mathbf{x}$-space can be measured by $\|w_1\|^2 + b_1^2$. So, it is easy to see that the minimization of $\|w_1\|^2 + b_1^2$ requires that the linear function (23) is as flat as possible in $\mathbf{x}$-space. Thus, the structural risk minimization principle is implemented. Note that the corresponding extra regularization term was proposed in our TBSVM [15] for classification problems. Peng [22] also added a similar regularization term in TSVR.

2. Except the regularization term $\frac{1}{2}c_3(\|w_1\|^2 + b_1^2)$, the other terms in (21) are different from (10) just because we choose the empirical risk (19) in ε-TSVR, whereas the empirical risk (8) is used in TSVR. Figure 1 gives the geometric interpretation of linear TSVR and ε-TSVR formulations for an example.

In order to get the solutions to problems (21) and (22), we need to derive their dual problems. The Lagrangian of the problem (21) is given by

$$\begin{aligned} L(w_1, b_1, \xi, \alpha, \beta) = & \frac{1}{2}(Y - (Aw_1 + eb_1))^\top (Y - (Aw_1 + eb_1)) \\ & + \frac{1}{2}c_3(\|w_1\|^2 + b_1^2) + c_1 e^\top \xi \\ & - \alpha^\top (Y - (Aw_1 + eb_1) + \varepsilon_1 e + \xi) - \beta^\top \xi, \end{aligned} \tag{24}$$

where $\alpha = (\alpha_1, \ldots, \alpha_l)$ and $\beta = (\beta_1, \ldots, \beta_l)$ are the vectors of Lagrange multipliers. The Karush-Kuhn-Tucker (K.K.T.) condition for w_1, b_1, ξ and α, β are given by:

$$-A^\top (Y - Aw_1 - eb_1) + c_3 w_1 + A^\top \alpha = 0, \tag{25}$$

$$-e^\top (Y - Aw_1 + eb_1) + c_3 b_1 + e^\top \alpha = 0, \tag{26}$$

$$c_1 e - \beta - \alpha = 0, \tag{27}$$

$$Y - (Aw_1 + eb_1) \geq -\varepsilon_1 e - \xi, \quad \xi \geq 0, \tag{28}$$

$$\alpha^\top (Y - (Aw_1 + eb_1) + \varepsilon_1 e + \xi) = 0, \quad \beta^\top \xi = 0, \tag{29}$$

$$\alpha \geq 0, \ \beta \geq 0. \tag{30}$$

Since $\beta \geq 0$, from the (27) we have

$$0 \leq \alpha \leq c_1 e. \tag{31}$$

Obviously, (25)–(26) imply that

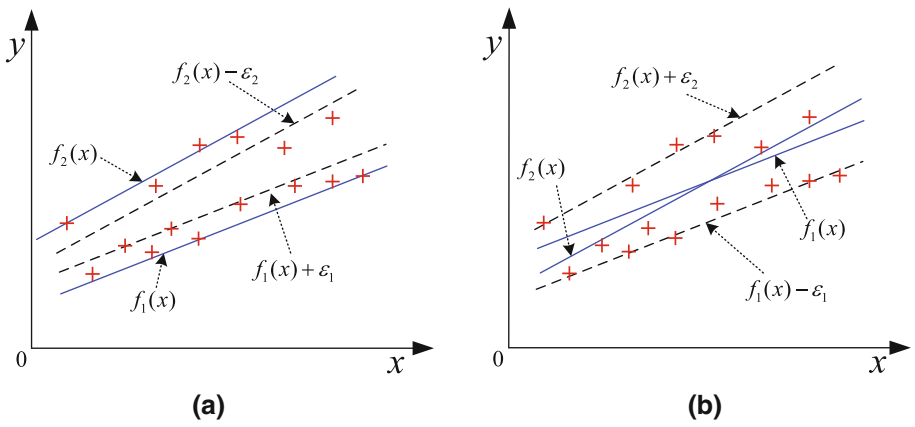

Fig. 1 The geometric interpretation for TSVR (**a**), ε-TSVR (**b**)

$$-\begin{bmatrix}A^\top\\ e^\top\end{bmatrix}Y+\left(\begin{bmatrix}A^\top\\ e^\top\end{bmatrix}[A\quad e]+c_3I\right)\begin{bmatrix}w_1\\ b_1\end{bmatrix}+\begin{bmatrix}A^\top\\ e^\top\end{bmatrix}\alpha=0. \tag{32}$$

Defining $G=[A\quad e]$, $v_1=[w_1^\top\ b_1]^\top$, the Eq. (32) can be rewritten as:

$$-G^\top Y+(G^\top G+c_3I)v_1+G^\top\alpha=0, \tag{33}$$

or

$$v_1=(G^\top G+c_3I)^{-1}G^\top(Y-\alpha). \tag{34}$$

Then putting (34) into the Lagrangian and using the above K.K.T. conditions, we obtain the dual problem of the problem (21):

$$\begin{array}{ll}\max\limits_{\alpha} & -\frac{1}{2}\alpha^\top G(G^\top G+c_3I)^{-1}G^\top\alpha^\top+Y^\top G(G^\top G\\ & +c_3I)^{-1}G^\top\alpha-(e^\top\varepsilon_1+Y^\top)\alpha\\ \text{s.t.} & 0\le\alpha\le c_1e,\end{array} \tag{35}$$

It is easy to see that the formulation of the problem (35) is similar to that of the problem (12) when the parameter c_3 in (35) is replaced by σ. However, σ in (12) is just a fixed small positive number, so the structural risk minimization principle can not be reflected completely as c_3 does. The experimental results in Sect. 4 will show that adjusting the value of c_3 can improve the classification accuracy indeed.

In the same way, the dual of the problem (22) is obtained:

$$\begin{array}{ll}\max\limits_{\gamma} & -\frac{1}{2}\gamma^\top G(G^\top G+c_4I)^{-1}G^\top\gamma^\top-Y^\top G(G^\top G\\ & +c_4I)^{-1}G^\top\gamma+(Y^\top-e^\top\varepsilon_2)\gamma\\ \text{s.t.} & 0\le\gamma\le c_2e,\end{array} \tag{36}$$

where γ is the Lagrange multiplier. The augmented vector $v_2=[w_2^\top\quad b_2]^\top$ is given by

$$v_2=(G^\top G+c_4I)^{-1}G^\top(Y+\gamma). \tag{37}$$

Once the solutions (w_1, b_1) and (w_2, b_2) of the problems (21) and (22) are obtained from the solutions of (35) and (36), the two proximal functions $f_1(x)$ and $f_2(x)$ are obtained. Then the estimated regressor is constructed as follows

$$f(x)=\frac{1}{2}(f_1(x)+f_2(x))=\frac{1}{2}(w_1+w_2)^\top x+\frac{1}{2}(b_1+b_2). \tag{38}$$

3.2 Kernel ε-TSVR

In order to extend our results to nonlinear regressors, consider the following two ε-insensitive proximal functions:

$$\begin{array}{l}f_1(x)=K(x^\top,A^\top)u_1+b_1\quad\text{and}\\ f_2(x)=K(x^\top,A^\top)u_2+b_2,\end{array} \tag{39}$$

where K is an appropriately chosen kernel. We construct the primal problems:

$$\begin{array}{ll}\min\limits_{u_1,b_1,\xi} & \frac{1}{2}c_3(u_1^\top u_1+b_1^2)+\frac{1}{2}\xi^\top\xi^*+c_1e^\top\xi,\\ \text{s.t.} & Y-(K(A,A^\top)u_1+eb_1)\ge-\varepsilon_1e-\xi,\quad\xi\ge0,\\ & Y-(K(A,A^\top)u_1+eb_1)=\xi^*,\end{array} \tag{40}$$

and

$$\begin{array}{ll}\min\limits_{u_2,b_2,\eta} & \frac{1}{2}c_4(u_2^\top u_2+b_2^2)+\frac{1}{2}\eta^\top\eta^*+c_2e^\top\eta,\\ \text{s.t.} & (K(A,A^\top)u_1+eb_1)-Y\ge-\varepsilon_2e-\eta,\quad\eta\ge0,\\ & (K(A,A^\top)u_2+eb_2)-Y=\eta^*.\end{array} \tag{41}$$

where c_1, c_2, c_3 and c_4 are positive parameters. In a similar way, we obtain their dual problems as the following:

$$\begin{array}{ll}\max\limits_{\alpha} & -\frac{1}{2}\alpha^\top H(H^\top H+c_3I)^{-1}H^\top\alpha^\top-(e^\top\varepsilon_1+Y^\top)\alpha\\ & +Y^\top H(H^\top H+c_3I)^{-1}H^\top\alpha\\ \text{s.t.} & 0\le\alpha\le c_1e.\end{array} \tag{42}$$

and

$$\begin{array}{ll}\max\limits_{\gamma} & -\frac{1}{2}\gamma^\top H(H^\top H+c_4I)^{-1}H^\top\gamma^\top+(Y^\top-e^\top\varepsilon_2)\gamma\\ & -Y^\top H(H^\top H+c_4I)^{-1}H^\top\gamma\\ \text{s.t.} & 0\le\gamma\le c_2e,\end{array} \tag{43}$$

where $H=[K(A,A^\top)\quad e]$. Then the augmented vector $v_1=[u_1^\top\quad b_1]^\top$ and $v_2=[u_2^\top\quad b_2]^\top$ are given by

$$v_1=(H^\top H+c_3I)^{-1}H^\top(Y-\alpha), \tag{44}$$

and

$$v_2=(H^\top H+c_4I)^{-1}H^\top(Y+\gamma). \tag{45}$$

Once the solutions (u_1,b_1) and (u_2,b_2) of the problems (40) and (41) are obtained from the solution of (42) and (43), the two functions $f_1(x)$ and $f_2(x)$ are obtained. Then the estimated regressor is constructed as follows

$$\begin{aligned}f(x)&=\frac{1}{2}(f_1(x)+f_2(x))\\&=\frac{1}{2}(u_1^\top+u_2^\top)K(A,x)+\frac{1}{2}(b_1+b_2).\end{aligned} \tag{46}$$

3.3 A fast ε-TSVR solver–successive overrelaxation technique

In our ε-TSVR, there are four strictly convex quadratic problems to be solved: (35), (36), (42), (43). It is easy to

see that these problems can be rewritten as the following unified form:

$$\begin{aligned} \max_{\alpha} \quad & d\alpha - \tfrac{1}{2}\alpha^{\top} Q\alpha, \\ \text{s.t.} \quad & \alpha \in S = \{0 \le \alpha \le ce\}, \end{aligned} \tag{47}$$

where Q is positive definite. For example, the above problem becomes the problem (35), when $Q = G(G^{\top}G + c_3 I)^{-1}G^{\top}$, $c = c_1$ and $d = Y^{\top}Q - (e^{\top}\varepsilon_1 + Y^{\top})$. Because the problem (35) entails the inversion of a possibly massive $m \times m$ matrix, to reduce the complexity of computation, we make immediate use of the Sherman–Morrison–Woodbury formula [28] for matrix inversion as used in [13, 14, 16, 29], which results in: $(c_3 I + G^{\top}G)^{-1} = \frac{1}{c_3}(I - G^{\top}(c_3 I + GG^{\top})^{-1}G)$.

The above problem (47) can be solved efficiently by the following successive overrelaxation (SOR) technique, see [15, 30].

Algorithm 3.1 Choose $t \in (0, 2)$. Start with any $\alpha^0 \in R^n$. Having α^i compute α^{i+1} as follows

$$\alpha^{i+1} = (\alpha^i - tE^{-1}(Q\alpha^i - d + L(\alpha^{i+1} - \alpha^i)))_{\sharp}, \tag{48}$$

until $\|\alpha^{i+1} - \alpha^i\|$ is less than some prescribed tolerance, where the nonzero elements of $L \in R^{m\times m}$ constitute the strictly lower triangular part of the symmetric matrix Q, and the nonzero elements of $E \in R^{m\times m}$ constitute the diagonal of Q.

SOR has been proved that this algorithm converges linearly to a solution. It should be pointed out that we also improve the original TSVR by applying the SOR technique to solve the problems (12) and (13). The experimental results in the following section will show that the SOR

Table 1 Performance metrics and their calculations

Metrics	Calculation
SSE	$\mathrm{SSE} = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2$
SST	$\mathrm{SST} = \sum_{i=1}^{m}(y_i - \bar{y})^2$
SSR	$\mathrm{SSR} = \sum_{i=1}^{m}(\hat{y}_i - \bar{y})^2$
NMSE	$\mathrm{NMSE} = \mathrm{SSE}/\mathrm{SST} = \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}$
R^2	$R^2 = \mathrm{SSR}/\mathrm{SST} = \frac{\sum_{i=1}^{m}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}$
MAPE	$\mathrm{MAPE} = \frac{\sum_{i=1}^{m}\lvert y_i - \hat{y}_i\rvert / y_i}{m} \times 100\,\%$

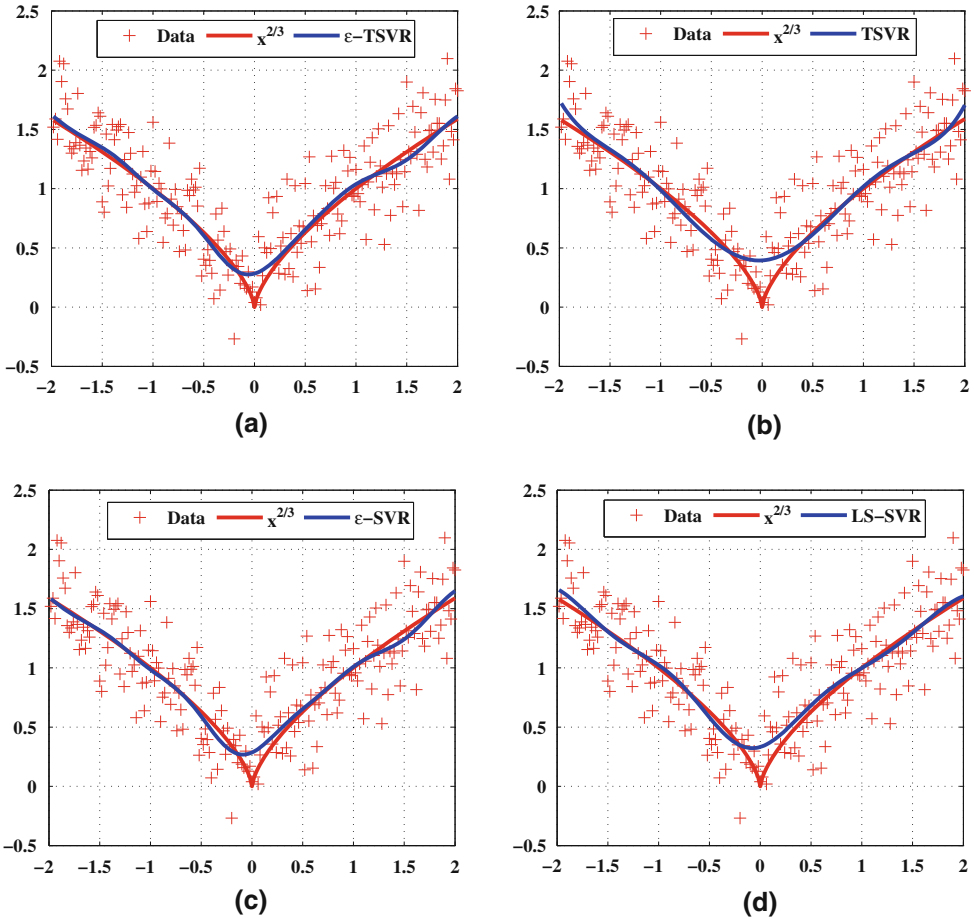

Fig. 2 Predictions of our ε-TSVR, TSVR, ε-SVR and LS-SVR on $y = x^{\frac{2}{3}}$ function

technique has remarkable acceleration effect on both our ε-TSVR and TSVR.

3.4 Comparison with TPISVR

We mentioned that our ε-TSVR is motivated by the TSVR. And different from only empirical risk is implemented in TSVR, a regularization term is added in ε-TSVR. Similar to ε-TSVR, TPISVR [23] also introduced a regularization term in its primal problems. Now, we show some comparisons of our ε-TSVR with TPISVR. The primal problems of linear TPISVR can be expressed as

$$\begin{aligned} \min_{w_1,b_1,\xi} \quad & \tfrac{1}{2}||w_1||^2 + \tfrac{v_1}{l} e^\top (Aw_1 + eb_1) + \tfrac{c_1}{l} e^\top \xi, \\ \text{s.t.} \quad & Y - (Aw_1 + eb_1) \geq -\xi, \quad \xi \geq 0, \end{aligned} \tag{49}$$

and

$$\begin{aligned} \min_{w_2,b_2,\eta} \quad & \tfrac{1}{2}||w_2||^2 + \tfrac{v_2}{l} e^\top (Aw_1 + eb_1) + \tfrac{c_2}{l} e^\top \eta, \\ \text{s.t.} \quad & Y - (Aw_1 + eb_1) \leq \eta, \quad \eta \geq 0, \end{aligned} \tag{50}$$

where c_1, c_2, v_1 and v_2 are positive parameters. From the above two QPPs, we can see that our ε-TSVR derives the similar characteristics as TPISVR, such as both of them has the same decisions and also loses the sparsity. However, by

Table 2 Result comparisons of our ε-TSVR, TSVR, ε-SVR and LS-SVR on datasets (51) and (52)

Dataset	Regressor	SSE	NMSE	R^2	CPU sec.
(51)	ε-TSVR	0.5249	0.0152	0.9988	0.008
	TSVR	1.2183	0.0352	0.9973	0.009
	ε-SVR	0.5672	0.0164	0.9986	39.4
	LS-SVR	0.8117	0.0235	0.9983	0.015
(52)	ε-TSVR	1.1016	0.0203	0.9911	0.073
	TSVR	1.8790	0.0347	0.9862	0.079
	ε-SVR	1.7298	0.0319	0.9858	35.8
	LS-SVR	2.1613	0.0399	0.9846	0.012

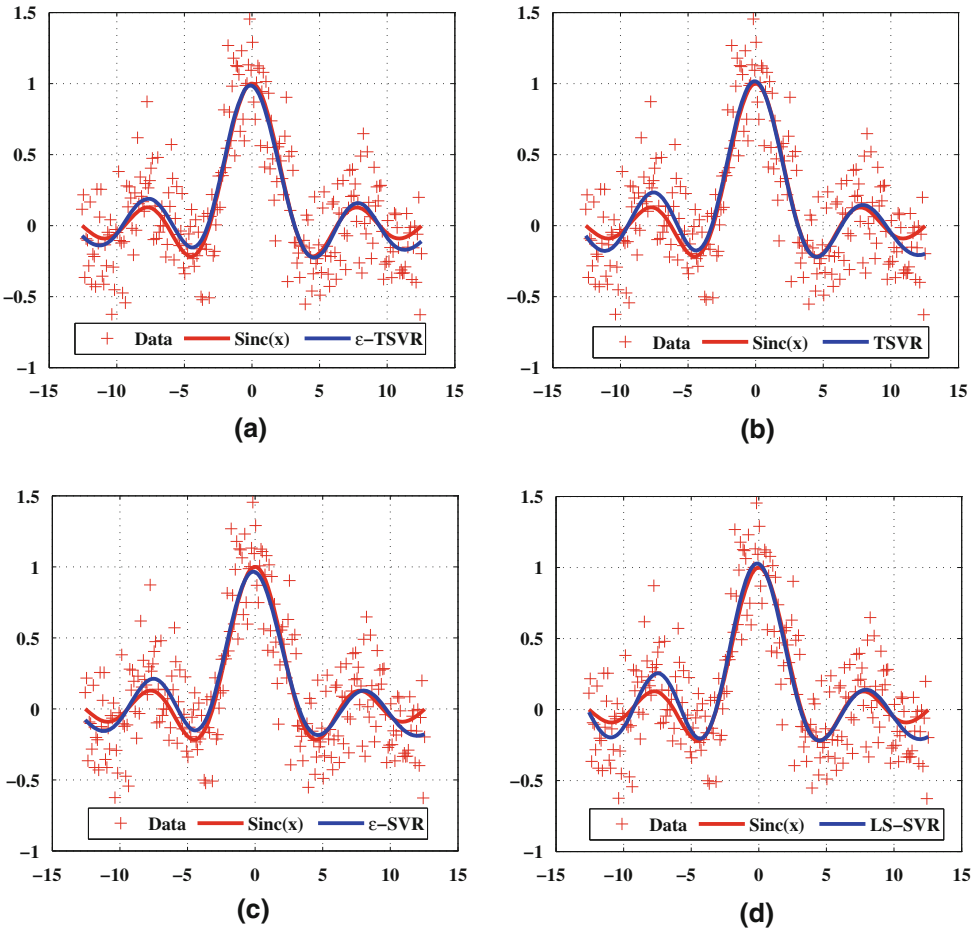

Fig. 3 Predictions of our ε-TSVR, TSVR, ε-SVR and LS-SVR on Sinc function

comparing the formulations of our ε-TSVR and TPISVR, we can see that both the objective functions and the constraints are differences, and Fig. 1a, b show the geometric interpretations of the TSVR (TPISVR) and our ε-TSVR, respectively. From Fig. 1, we can see that the two algorithms have some differences in geometric interpretations. TPISVR constructs a pair of up-bound and down-bound functions, while our ε-TSVR constructs a pair of ε-insensitive proximal functions. In addition, in our ε-TSVR, only have the boundary constraints, this strategy makes SOR can be used for our ε-TSVR (and TSVR), which obtains the less learning cost than TPISVR. But in our ε-TSVR, the parameters have no meaning of bounds of the fractions of SVs and margin errors as in TPISVR.

4 Experimental results

In this section, some experiments are made to demonstrate the performance of our ε-TSVR compared with the TSVR, ε-SVR, and LS-SVR on several datasets, including two type artificial datasets and nine benchmark datasets. All methods are implemented in Matlab 7.0 [31] environment on a PC with an Intel P4 processor (2.9 GHz) with 1 GB RAM. ε-SVR and LS-SVR are solved by the optimization toolbox: QP and LS-SVMlab in Matlab [31, 32] respectively. Our ε-TSVR and TSVR are solved by SOR technique. The values of the parameters in four methods are obtained through searching in the range 2^{-8}–2^{8} by tuning a set comprising of random 10 % of the dataset. In our experiments, we set $c_1 = c_2$, $c_3 = c_4$ and $\varepsilon_1 = \varepsilon_2$, to degrade the computational complexity of parameter selection.

4.1 Performance criteria

In order to evaluate the performance of the algorithms, some evaluation criteria [19, 33, 34] commonly used should be introduced firstly. Without loss of generality, let l be the number of training samples and denote m as the number of testing samples, $\hat{y}_i$ as the prediction value of y_i, and $\bar{y} = \frac{1}{m}\sum_i y_i$ as the average value of $y_1, \ldots, y_m$. Then the definitions of some criteria are stated in Table 1.

4.2 Artificial datasets

In order to compare our ε-TSVR with TSVR, ε-SVR, LS-SVR, we choose the same artificial datasets to the ones in [19] and [35]. Firstly, consider the function: $y = x^{\frac{2}{3}}$. To effectively reflect the performance of the methods, training samples are polluted by Gaussian noises with zero means and 0.2 standard deviation, i.e. we have the following training samples (x_i, y_i):

$$y_i = x_i^{\frac{2}{3}} + \xi_i, \quad x \sim U[-2, 2],\ \xi_i \sim N(0, 0.2^2), \tag{51}$$

where $U[a, b]$ represents the uniformly random variable in [a, b] and $N(\bar{a}, \bar{b}^2)$ represents the Gaussian random variable with means $\bar{a}$ and $\bar{b}$ standard deviation, respectively. To avoid biased comparisons, ten independent groups of noisy samples are generated randomly using Matlab toolbox, which consists of 200 training samples and 200 none noise test samples.

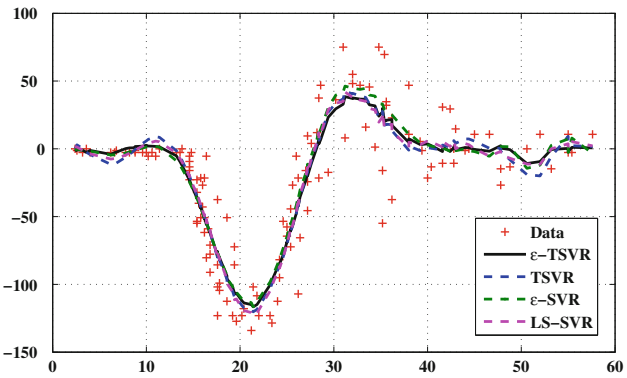

Fig. 4 Predictions of ε-TSVR, TSVR, ε-SVR and LS-SVR on the "Motorcycle" dataset

Table 3 The description for real datasets

Dataset	No. of samples	No. of features
Motorcycle	133	1
Diabetes	43	2
Servo	167	4
Auto price	159	15
Machine CPU	209	17
Wisconsin B.C.	194	32
Auto-Mpg	398	7
Boston housing	506	13
Concrete CS	1,030	8

Table 4 Result comparisons of ε-TSVR, TSVR, ε-SVR and LS-SVR on Motorcycle and Diabetes datasets

Dataset	Regressor	NMSE	R^2	CPU sec.
Motorcycle	ε-TSVR	0.2132	0.7992	0.067
	TSVR	0.2446	0.7800	0.064
	ε-SVR	0.2214	0.7963	43.60
	LS-SVR	0.2261	0.7760	0.001
Diabetes	ε-TSVR	0.7041	0.6340	0.002
	TSVR	0.7115	0.5825	0.002
	ε-SVR	0.7418	0.6075	0.30
	LS-SVR	0.7573	0.5961	0.001

Figure 2a–d illustrate the estimated functions obtained by these four methods, respectively. It can be observed that our ε-TSVR obtain the best approximation among four methods. The results of the performance criteria are listed in Table 2. Our ε-TSVR derives the smallest SSE, NMSE and the largest R^2 among the four methods. This indicates the statistical information in the training dataset is well presented by our ε-TSVR with fairly small regression

Table 5 Result comparisons of ε-TSVR, TSVR, ε-SVR and LS-SVR on UCI datasets

Dataset	Regressor	NMSE	R^2	MAPE	CPU sec.
Servo	ε-TSVR	0.2130 ± 0.1035	0.9609 ± 0.1563	0.3334 ± 0.1439	0.011
	TSVR	0.2209 ± 0.1530	0.9303 ± 0.1643	0.3992 ± 0.2020	0.009
	ε-SVR	0.2415 ± 0.1620	0.9130 ± 0.1543	0.3762 ± 0.1551	10.8
	LS-SVR	0.2579 ± 0.1120	0.9101 ± 0.1356	0.3749 ± 0.2999	0.002
Auto Price	ε-TSVR	0.3348 ± 0.0837	0.9240 ± 0.2356	0.3939 ± 0.0938	0.008
	TSVR	0.3199 ± 0.0951	0.9018 ± 0.2008	0.3765 ± 0.0940	0.014
	ε-SVR	0.3485 ± 0.0867	1.0824 ± 0.2101	0.3394 ± 0.0951	9.8
	LS-SVR	0.3691 ± 0.1104	0.9166 ± 0.2461	0.4158 ± 0.0940	0.016
Machine CPU	ε-TSVR	0.2104 ± 0.0302	0.8775 ± 0.3069	0.1618 ± 0.0633	0.012
	TSVR	0.2158 ± 0.0921	0.8834 ± 0.3485	0.1623 ± 0.0931	0.012
	ε-SVR	0.2001 ± 0.0554	0.8911 ± 0.2146	0.1614 ± 0.0882	21.6
	LS-SVR	0.2041 ± 0.0981	0.8814 ± 0.2416	0.1746 ± 0.0907	0.004
Wisconsin B.C.	ε-TSVR	0.8470 ± 0.1149	0.5685 ± 0.1052	0.9139 ± 0.1062	0.020
	TSVR	0.9065 ± 0.1097	0.4641 ± 0.1164	0.9544 ± 0.3762	0.027
	ε-SVR	0.9093 ± 0.1182	0.5173 ± 0.1032	1.0626 ± 0.1197	14.5
	LS-SVR	0.9235 ± 0.1074	0.4628 ± 0.1305	0.9493 ± 0.1922	0.009
Auto-Mpg	ε-TSVR	0.0980 ± 0.0268	1.0284 ± 0.1343	0.3163 ± 0.0319	0.036
	TSVR	0.1167 ± 0.0349	1.0698 ± 0.1643	0.3214 ± 0.0382	0.041
	ε-SVR	0.1160 ± 0.0235	0.8999 ± 0.1358	0.3298 ± 0.0161	210.8
	LS-SVR	0.1188 ± 0.0438	0.8898 ± 0.1427	0.3445 ± 0.0863	0.012
Boston housing	ε-TSVR	0.1146 ± 0.0263	1.0401 ± 0.1028	–	0.176
	TSVR	0.1238 ± 0.0721	1.0333 ± 0.1314	–	0.172
	ε-SVR	0.1135 ± 0.0682	1.0632 ± 0.1212	–	581.5
	LS-SVR	0.1435 ± 0.0524	0.8667 ± 0.1475	–	0.085
Concrete CS	ε-TSVR	0.1004 ± 0.0221	0.9897 ± 0.0764	0.2405 ± 0.0689	0.720
	TSVR	0.1027 ± 0.0325	0.9801 ± 0.0725	0.2414 ± 0.0779	0.690
	ε-SVR	0.1664 ± 0.0342	0.9072 ± 0.0521	0.9207 ± 0.0709	1,744.5
	LS-SVR	0.1969 ± 0.0281	0.8185 ± 0.0941	0.9111 ± 0.0279	0.130

Table 6 The best parameters of ε-TSVR and TSVR on UCI datasets

Dataset	ε-TSVR				TSVR			
	$c_1 = c_2$	$c_3 = c_4$	$\varepsilon_1 = \varepsilon_2$	p	$c_1 = c_2$	σ	$\varepsilon_1 = \varepsilon_2$	p
Motorcycle	0.25	1	4	8	0.0156	10^{-7}	8	16
Diabetes	1	0.0625	4	2	0.0625	10^{-7}	4	8
Servo	0.0039	0.0039	0.0078	2	4	10^{-7}	2	4
Auto Price	4	0.0313	0.25	256	2	10^{-7}	32	4
Machine CPU	0.5	8	0.125	8	64	10^{-7}	0.0156	8
Wisconsin B.C.	128	4	0.25	–	2	10^{-7}	1	–
Auto-Mpg	64	0.0078	0.0313	16	0.0625	10^{-7}	1	16
Boston housing	256	0.0625	0.5	1	64	10^{-7}	0.0039	1
Concrete CS	0.25	0.0039	0.0156	1	0.0039	10^{-7}	0.1250	4

errors. Besides, Table 2 also compares the training CPU time for these four methods. It can be seen that ε-TSVR is the fastest learning method, indicating that SOR technique can improve the training speed.

Furthermore, we generate the datasets by the sinc function that is polluted by Gaussian noises with zero means and 0.2 standard deviation. We have the training samples (x_i, y_i):

$$y_i = \frac{\sin(x_i)}{x_i} + \xi_i, \quad x \sim U[-4\pi, 4\pi],\ \xi_i \sim N(0, 0.2^2). \tag{52}$$

Our dataset consists of 252 training samples and 503 test samples. Figure 3a–d illustrate the estimated functions obtained by four methods and Table 2 lists the corresponding performances. These results also show the superiority of our method.

4.3 UCI datasets

For further evaluation, we test nine benchmark datasets: the Motorcycle [36], Diabetes, Boston housing, Auto-Mpg, Machine CPU, Servo, Concrete Compressive Strength, Auto price, and Wisconsin breast cancer datasets [37], which are commonly used in testing machine learning algorithms. More detailed description can be found in Table 3.

Because the two datasets "Motorcycle" and "Diabetes" have smaller number of samples and features, the criterions leave-one-out are used on them. Figure 4 shows the regression comparison of our ε-TSVR, TSVR, ε-SVR and LS-SVR on the "Motorcycle" dataset. Table 4 lists the learning results of these four methods on the Motorcycle and Diabetes datsets. It can be seen that our ε-TSVR outperforms the other three methods. For instance, our ε-TSVR obtains the largest R^2 and the smallest *NMSE* among the four methods. As for the computation time, although LS-SVR spends on the least CPU time, our ε-TSVR needs far less CPU time than ε-SVR, indicating that ε-TSVR is an efficient algorithm for regression.

For the other seven datasets, the criterions "NMSE", "R^2" and "MAPE" are used. Table 5 shows the testing results of the proposed ε-TSVR, TSVR, ε-SVR and LS-SVR on these seven datasets. The results in Table 5 are similar with that appeared in Table 4 and therefore confirm the superiority of our method further. Table 6 lists the best parameters selected by ε-TSVR and TSVR on the above UCI datasets. It can be seen that the c_3 and c_4 vary and are usually not take a smaller value in our ε-TSVR, while σ is a fixed small positive scalar in TSVR. This implies that the regularization terms in our ε-TSVR are useful.

5 Conclusions

For regression problems, an improved version ε-TSVR based on TSVR is proposed in this study. The main contribution is that the structural risk minimization principle is implemented by adding the regularization term in the primal problems of our ε-TSVR. This embodies the marrow of statistical learning theory. The parameters c_3 and c_4 introduced are the weights between the regularization term and the empirical risk, so they can be chosen flexibly. In addition, the application of SOR technique is also an excellent contribution, since it speeds up the training procedure. Computational comparisons between our ε-TSVR and other methods including TSVR, ε-SVR, and LS-SVR have been made on several datasets, indicating that our ε-TSVR is not only faster, but also shows better generalization. We believe that its nice generalization mainly comes from the fact that the parameters c_3 and c_4 are adjusted properly. Our ε-TSVR Matlab codes can be downloaded from: http://math.cau.edu.cn/dengnaiyang.html.

It should be pointed out that as does the TWSVM, our ε-TSVR also loses sparsity, so it is interesting to find a sparsity algorithm for our ε-TSVR. Note that, for LS-SVR, there are some relevant improvements such as the work in [38, 39]. May be they are useful references in further study. Besides, the parameters selection is a practical problem and should be addressed in the future too.

Acknowledgments We thank the anonymous reviewers for their valuable suggestions. This work is supported by the National Natural Science Foundation of China (No. 10971223, No. 11071252, No. 11161045 and No. 61101231) and the Zhejiang Provincial Natural Science Foundation of China (No. Y1100237, No. Y1100629).

References

1. Cortes C, Vapnik VN (1995) Support vector networks. Mach Learn 20:273–297
2. Vapnik VN (1998) Statistical learning theory. Wiley, New York
3. Burges C (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2:121–167
4. Deng NY, Tian YJ, Zhang CH (2012) Support vector machines: theory, algorithms, and extensions. CRC Press, Boca Raton
5. Noble WS (2004) Support vector machine applications in computational biology. In: Schöelkopf B, Tsuda K, Vert J-P (eds) Kernel methods in computational biology. MIT Press, Cambridge, pp 71–92
6. Lee S, Verri A (2002) Pattern recognition with support vector machines. In: First international workshop, Springer, Niagara Falls, Canada
7. Ince H, Trafalis TB (2002) Support vector machine for regression and applications to financial forecasting. In: International joint conference on neural networks, Como, Italy, IEEE-INNS-ENNS
8. Suykens JAK, Lukas L, van Dooren P, De Moor B, Vandewalle J (1999) Least squares support vector machine classifiers: a large scale algorithm. In: Proceedings of European conference of circuit theory design, pp 839–842

9. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector classification via generalize deigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
10. Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
11. Kumar MA, Gopal M (2008) Application of smoothing technique on twin support vector machines. Pattern Recognit Lett 29(13): 1842–1848
12. Shao YH, Deng NY (2012) A novel margin based twin support vector machine with unity norm hyperplanes. Neural Comput Appl. doi:10.1007/s00521-012-0894-5
13. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543
14. Ghorai S, Mukherjee A, Dutta PK (2009) Nonparallel plane proximal classifier. Signal Process 89(4):510–522
15. Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–968
16. Shao YH, Deng NY (2012) A coordinate descent margin based-twin support vector machine for classification. Neural Netw 25:114–121
17. Peng X (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recognit 44(10–11):2678–2692
18. Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
19. Peng X (2010) TSVR: an efficient twin support vector machine for regression. Neural Netw 23(3):365–372
20. Zhong P, Xu Y, Zhao Y (2011) Training twin support vector regression via linear programming. Neural Comput Appl. doi: 10.1007/s 00521-011-0526-6
21. Chen X, Yang J, Liang J, Ye Q (2011) Smooth twin support vector regression. Neural Comput Appl. doi:10.1007/s00521-010-0454-9
22. Peng X (2010) Primal twin support vector regression and its sparse approximation. Neurocomputing 73(16–18):2846–2858
23. Peng X (2012) Efficient twin parametric insensitive support vector regression model. Neurocomputing 79:26–38
24. Chen X, Yang J, Liang J (2011) A flexible support vector machine for regression. Neural Comput Appl. doi:10.1007/s00521-011-0623-5
25. Schölkopf B, Smola A (2002) Learning with kernels. MIT Press, Cambridge
26. Bi J, Bennett KP (2003) A geometric approach to support vector regression. Neurocomputing 55:79–108
27. Smola A, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222
28. Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. The John Hopkins University Press, Baltimore
29. Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: Proceedings of seventh international conference on knowledge and data discovery, San Francisco, pp 77–86
30. Mangasarian OL, Musicant DR (1999) Successive overrelaxation for support vector machines. IEEE Trans Neural Netw 10(5):1032–1037
31. http://www.mathworks.com (2007)
32. Pelckmans K, Suykens JAK, Van Gestel T, De Brabanter D, Lukas L, Hamers B, De Moor B, Vandewalle J (2003) LS-SVMlab: a Matlab/C toolbox for least squares support vector machines. Available at http://www.esat.kuleuven.ac.be/sista/lssvmlab
33. Weisberg S (1985) Applied linear regression seconded. Wiley, New York
34. Staudte RG, Sheather SJ (1990) Robust estimationand testing: Wiley series in probability and mathematical statistics. Wiley, New York
35. Lee CC, Chung PC, Tsai JR, Chang CI (1999) Robust radial basis function neural networks. IEEE Trans Syst Man Cybern B Cybern 29(6):674–685
36. Eubank RL (1999) Nonparametric regression and spline smoothing statistics: textbooks and monographs, vol 157, seconded. Marcel Dekker, New York
37. Blake CL, Merz CJ (1998) UCI repository for machine learning databases. Department of Information and Computer Sciences, University of California, Irvine, http://www.ics.uci.edu/mlearn/MLRepository.html
38. Jiao L, Bo L, Wang L (2007) Fast sparse approximation for least squares support vector machine. IEEE Trans Neural Netw 18:1–13
39. Wen W, Hao Z, Yang X (2008) A heuristic weight-setting strategy and iteratively updating algorithm for weighted least-squares support vector regression. Neurocomputing 71:3096–3103