Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Primal least squares twin support vector regression*

Hua-juan HUANG[1], Shi-fei DING[†‡1,2], Zhong-zhi SHI[2]

([1]*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China*)

([2]*Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*)

[†]E-mail: dingsf@cumt.edu.cn

Received Jan. 27, 2013; Revision accepted June 17, 2013; Crosschecked Aug. 7, 2013

Abstract: The training algorithm of classical twin support vector regression (TSVR) can be attributed to the solution of a pair of quadratic programming problems (QPPs) with inequality constraints in the dual space. However, this solution is affected by time and memory constraints when dealing with large datasets. In this paper, we present a least squares version for TSVR in the primal space, termed primal least squares TSVR (PLSTSVR). By introducing the least squares method, the inequality constraints of TSVR are transformed into equality constraints. Furthermore, we attempt to directly solve the two QPPs with equality constraints in the primal space instead of the dual space; thus, we need only to solve two systems of linear equations instead of two QPPs. Experimental results on artificial and benchmark datasets show that PLSTSVR has comparable accuracy to TSVR but with considerably less computational time. We further investigate its validity in predicting the opening price of stock.

Key words: Twin support vector regression, Least squares method, Primal space, Stock prediction
doi:10.1631/jzus.CIIP1301 **Document code:** A **CLC number:** TP18

1 Introduction

Support vector machine (SVM) is a computationally powerful kernel-based tool for binary data classification and regression (Vapnik, 1995; Ding and Qi, 2012; Huang and Ding, 2012). Based on the structural risk minimization principle, SVM has successfully solved the high dimensionality and local minimum problems. Therefore, compared with other machine learning methods, such as artificial neural networks (Ding *et al.*, 2011; Liu *et al.*, 2012; Xu *et al.*, 2012), SVM has better generalization performance. So far, SVM has achieved excellent performance in many real-world predictive data mining applications such as text categorization (Pan *et al.*, 2013), time series prediction (Chen and Fan, 2012), pattern recognition (Moraes *et al.*, 2013), and image processing (Wu, 2012).

Although SVM has achieved good learning performance, its training time is high, which has limited its application in dealing with large datasets. To reduce the computational complexity of SVM, many improved algorithms have been presented, such as the chunking algorithm (Boser *et al.*, 1992), decomposition algorithm (Osuna *et al.*, 1997), and sequential minimal optimization (SMO) (Platt, 1999). Experimental results showed that these algorithms could improve the efficiency of SVM. However, the implementation of these algorithms is very complex. On the other hand, some deformation algorithms based on the standard SVM have been proposed in recent years. For example, Mangasarian and Wild (2006) proposed a nonparallel plane classifier based on SVM, named generalized eigenvalue proximal support vector machine (GEPSVM). GEPSVM aims at generating two nonparallel hyperplanes by solving two generalized eigenvalue problems such that each hyperplane is close to its class and as far as possible from the other class. GEPSVM has good learning speed, but its classification accuracy is low. Jayadeva *et al.* (2007) proposed a new machine learning

‡ Corresponding author
* Project supported by the National Basic Research Program (973) of China (No. 2013CB329502), the National Natural Science Foundation of China (No. 61379101), and the Fundamental Research Funds for the Central Universities, China (No. 2012LWB39)

method called twin support vector machine (TWSVM) for binary classification in the spirit of GEPSVM. TWSVM would generate two non-parallel planes, such that each plane is closer to one of the two classes and as far as possible from the other. In TWSVM, a pair of smaller sized quadratic programming problems (QPPs) is solved, instead of a single large one in SVM, making the computational speed of TWSVM approximately four times higher than that of the traditional SVM. Because of its excellent performance, TWSVM has been applied to many areas such as speaker recognition (Cong *et al.*, 2008), and medical detection (Zhang *et al.*, 2009).

As for support vector regression (SVR), there are also many improved algorithms. In particular, Suykens and van de Walle (2001) proposed least squares support vector regression (LS-SVR) by introducing the least squares method. In LS-SVR, the inequality constraints of SVR are transformed into equality constraints. This strategy can reduce the complexity of SVR and thus improve the learning speed. However, the robustness of LS-SVR is not as good as that of SVR. Peng (2010b) introduced a new nonparallel plane regression in the spirit of TWSVM, termed twin support vector regression (TSVR). TSVR also aims at generating two nonparallel functions such that each function determines the ε-insensitive down- or up-bounds of the unknown regressor. Similar to TWSVM, TSVR needs only to solve a pair of smaller QPPs, instead of a single large one in SVR. Furthermore, the number of constraints of each QPP in TSVR is only half of that of the classical SVR, which makes TSVR work faster than SVR. Similar to SVR, TSVR solves the QPPs in the dual space. However, this solution is affected by time and memory constraints when dealing with large datasets, leading to a low learning speed of TSVR.

To improve the learning speed of TSVR, in this paper we enhance TSVR to least squares TSVR (LSTSVR) by introducing the least squares method and then solve LSTSVR directly in the primal space instead of the dual space. Based on this idea, the primal least squares twin support vector regression (PLSTSVR) is proposed. First, introducing the least squares method, the inequality constraints of TSVR are transformed into equality constraints. Furthermore, we attempt to directly solve the two QPPs with equality constraints in the primal space instead of the dual space; thus, we need only to solve two systems of linear equations instead of two QPPs. Experimental results on artificial and benchmark datasets show that PLSTSVR surpasses TSVR and SVR in speed and accuracy. We further investigate its validity in predicting the opening price of stock.

2 Background

Denote $\boldsymbol{A}\in\mathbb{R}^{l\times n}$ as the input sample matrix, where $\boldsymbol{A}_i=(A_{i1}, A_{i2}, \ldots, A_{in})$, i=1, 2, …, l. Let $\boldsymbol{Y}=(y_1, y_2, \ldots, y_l)^{\mathrm{T}}$ denote the output vector, in which y_i (i=1, 2, …, l) are the output values. The problem formulations and the dual problems of SVR and TSVR are discussed in the following.

2.1 Support vector regression

For the linear case, introducing the ε-insensitive loss function, we would like to find the regression function

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

where $\boldsymbol{w}\in\mathbb{R}^n$ and $b\in\mathbb{R}$. Similar to the idea of SVM, the function $f(\boldsymbol{x})$ should be made as flat as possible through minimizing $||\boldsymbol{w}||^2$. At the same time, the fitting error beyond precision is considered by introducing the slack variables $\boldsymbol{\xi}, \boldsymbol{\xi}^*$. Based on the above idea, the linear SVR can be formulated as follows:

$$\begin{aligned}\min\ & \frac{1}{2}||\boldsymbol{w}||^2+C\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{\xi}+\boldsymbol{\xi}^*)\\ \text{s.t.}\ & \boldsymbol{Y}-(\boldsymbol{Aw}+b\boldsymbol{e})\le\varepsilon\boldsymbol{e}+\boldsymbol{\xi},\ \ \boldsymbol{\xi}\ge\boldsymbol{0},\\ & (\boldsymbol{Aw}+b\boldsymbol{e})-\boldsymbol{Y}\le\varepsilon\boldsymbol{e}+\boldsymbol{\xi}^*,\ \ \boldsymbol{\xi}^*\ge\boldsymbol{0},\end{aligned} \tag{2}$$

where C>0 is the penalty parameter.

For the nonlinear SVR, similar to nonlinear SVM, a nonlinear map φ: $\mathbb{R}^n\rightarrow\chi$ is introduced, where χ is the feature space. According to Mercer's theorem, $K(\boldsymbol{x}^{\mathrm{T}}, y)=<\varphi(\boldsymbol{x}), \varphi(y)>$ can be used to represent the inner product in χ by introducing some kernel $K(\boldsymbol{x}^{\mathrm{T}}, y)$. Therefore, the linear regression function in the χ space can be expressed as follows:

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x})+b. \tag{3}$$

And the nonlinear SVR formulation is

$$\begin{aligned}&\min\ \frac{1}{2}\|\boldsymbol{w}\|^2+C\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{\xi}+\boldsymbol{\xi}^*)\\&\text{s.t.}\ \ \boldsymbol{Y}-(\varphi(\boldsymbol{A})\boldsymbol{w}+b\boldsymbol{e})\le\varepsilon\boldsymbol{e}+\boldsymbol{\xi},\ \ \boldsymbol{\xi}\ge\boldsymbol{0},\\&\qquad(\varphi(\boldsymbol{A})\boldsymbol{w}+b\boldsymbol{e})-\boldsymbol{Y}\le\varepsilon\boldsymbol{e}+\boldsymbol{\xi}^*,\ \ \boldsymbol{\xi}^*\ge\boldsymbol{0},\end{aligned}\tag{4}$$

where $\varphi(\boldsymbol{A})=(\varphi(\boldsymbol{A}_1), \varphi(\boldsymbol{A}_2), \ldots, \varphi(\boldsymbol{A}_l))$. An intuitive geometric interpretation for nonlinear SVR is shown in Fig. 1.

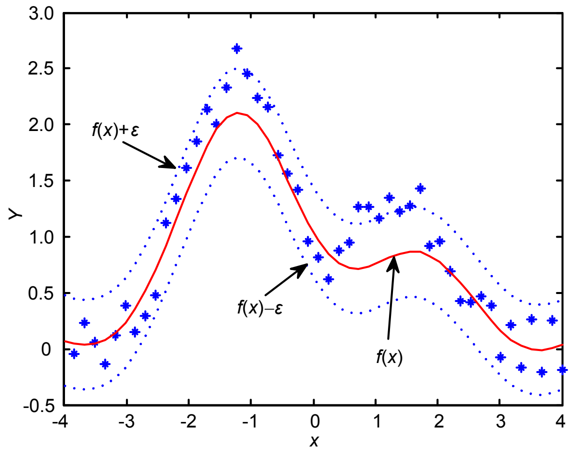

Fig. 1 The geometric interpretation for kernel support vector regression (SVR)

Note that in Eqs. (2) and (4) there are two groups of constraints. Generally they are solved in the dual space. For example, the dual QPP of Eq. (4) is

$$\begin{aligned}&\min\left(\varepsilon\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*)-\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)\right.\\&\qquad\left.+\frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^{\mathrm{T}}K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)\right)\\&\text{s.t.}\ \ \boldsymbol{e}^{\mathrm{T}}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)=0,\ \ \boldsymbol{0}\le\boldsymbol{\alpha},\boldsymbol{\alpha}^*\le C\boldsymbol{e},\end{aligned}\tag{5}$$

where $\boldsymbol{\alpha}^*=(\alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*)^{\mathrm{T}}\in\mathbb{R}^l$ is the Lagrangian multiplier vector. According to the Mercer theorem, we can achieve nonlinear regression using the kernel function $K(\boldsymbol{x}, \boldsymbol{y})$. Solving the QPP (5), we obtain the nonlinear SVR regression function:

$$f(\boldsymbol{x})=\sum_{i=0}^{l}(\alpha_i^*-\alpha_i)K(\boldsymbol{x}_i,\boldsymbol{x})+b.\tag{6}$$

2.2 Twin support vector regression

Similar to TWSVM, TSVR generates two nonparallel functions around the data points.

For the linear case, TSVR aims to find two nonparallel functions:

$$f_1(\boldsymbol{x})=\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}+b_1,\tag{7}$$

$$f_2(\boldsymbol{x})=\boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{x}+b_2.\tag{8}$$

The two functions are obtained by solving the following QPPs:

$$\begin{aligned}&\min\ \frac{1}{2}\left\|\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}-(\boldsymbol{A}\boldsymbol{w}_1+b_1\boldsymbol{e})\right\|^2+C_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}\\&\text{s.t.}\ \ \boldsymbol{Y}-(\boldsymbol{A}\boldsymbol{w}_1+b_1\boldsymbol{e})\ge\varepsilon_1\boldsymbol{e}-\boldsymbol{\xi},\ \boldsymbol{\xi}\ge\boldsymbol{0},\end{aligned}\tag{9}$$

$$\begin{aligned}&\min\ \frac{1}{2}\left\|\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}-(\boldsymbol{A}\boldsymbol{w}_2+b_2\boldsymbol{e})\right\|^2+C_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta}\\&\text{s.t.}\ \ (\boldsymbol{A}\boldsymbol{w}_2+b_2\boldsymbol{e})-\boldsymbol{Y}\ge\varepsilon_2\boldsymbol{e}-\boldsymbol{\eta},\ \boldsymbol{\eta}\ge\boldsymbol{0},\end{aligned}\tag{10}$$

where C_1, C_2>0, ε_1, ε_2>0 are the parameters, and $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ are the slack vectors.

Introducing the Lagrangian multiplier vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, the dual QPPs of (9) and (10) can be obtained as follows:

$$\begin{aligned}&\max\ -\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{G}(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\alpha}\\&\qquad\text{s.t.}\ \ \boldsymbol{0}\le\boldsymbol{\alpha}\le C_1\boldsymbol{e},\end{aligned}\tag{11}$$

$$\begin{aligned}&\max\ -\frac{1}{2}\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{G}(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\gamma}-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{G}(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\gamma}+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\gamma}\\&\qquad\text{s.t.}\ \ \boldsymbol{0}\le\boldsymbol{\gamma}\le C_2\boldsymbol{e},\end{aligned}\tag{12}$$

where $\boldsymbol{G}=[\boldsymbol{A}\ \boldsymbol{e}]$, $\boldsymbol{f}=\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}$, and $\boldsymbol{h}=\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}$. Then we obtain the regression function of TSVR:

$$f(\boldsymbol{x})=\frac{1}{2}(f_1(\boldsymbol{x})+f_2(\boldsymbol{x}))=\frac{1}{2}(\boldsymbol{w}_1+\boldsymbol{w}_2)^{\mathrm{T}}\boldsymbol{x}+\frac{1}{2}(b_1+b_2),\tag{13}$$

where

$$\begin{bmatrix}\boldsymbol{w}_1\\b_1\end{bmatrix}=(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{f}-\boldsymbol{\alpha}),\ \ \begin{bmatrix}\boldsymbol{w}_2\\b_2\end{bmatrix}=(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{h}+\boldsymbol{\gamma}).$$

For the nonlinear case, TSVR considers the following kernel-generated functions:

$$\begin{cases}f_1(\boldsymbol{x})=K(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_1+b_1,\\f_2(\boldsymbol{x})=K(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_2+b_2.\end{cases}\tag{14}$$

Similarly, solution to Eq. (14) can be obtained by dealing with the following QPPs:

$$\begin{aligned} \min\ & \frac{1}{2}\left\|\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_1+b_1\boldsymbol{e})\right\|^2+C_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi} \\ \text{s.t.}\ & \boldsymbol{Y}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_1+b_1\boldsymbol{e})\geq\varepsilon_1\boldsymbol{e}-\boldsymbol{\xi},\ \boldsymbol{\xi}\geq\boldsymbol{0}, \end{aligned} \tag{15}$$

$$\begin{aligned} \min\ & \frac{1}{2}\left\|\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_2+b_2\boldsymbol{e})\right\|^2+C_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta} \\ \text{s.t.}\ & (K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_2+b_2\boldsymbol{e})-\boldsymbol{Y}\geq\varepsilon_2\boldsymbol{e}-\boldsymbol{\eta},\ \boldsymbol{\eta}\geq\boldsymbol{0}. \end{aligned} \tag{16}$$

According to the KKT conditions, the dual problems of (15) and (16) are as follows:

$$\begin{aligned} \max\ & -\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\alpha}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\alpha} \\ & \text{s.t.}\ \boldsymbol{0}\leq\boldsymbol{\alpha}\leq C_1\boldsymbol{e}, \end{aligned} \tag{17}$$

$$\begin{aligned} \max\ & -\frac{1}{2}\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\gamma}-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\gamma}+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\gamma} \\ & \text{s.t.}\ \boldsymbol{0}\leq\boldsymbol{\gamma}\leq C_2\boldsymbol{e}, \end{aligned} \tag{18}$$

where $\boldsymbol{H}=[K(\boldsymbol{A},\ \boldsymbol{A}^{\mathrm{T}})\ \boldsymbol{e}]$. After solving Eqs. (17) and (18), we obtain

$$\begin{cases} [\boldsymbol{w}_1\ b_1]^{\mathrm{T}}=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{f}-\boldsymbol{\alpha}), \\ [\boldsymbol{w}_2\ b_2]^{\mathrm{T}}=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{h}+\boldsymbol{\gamma}). \end{cases} \tag{19}$$

Then the regression function of nonlinear TSVR is constructed as follows:

$$\begin{aligned} f(\boldsymbol{x}) &= \frac{1}{2}(f_1(\boldsymbol{x})+f_2(\boldsymbol{x})) \\ &= \frac{1}{2}K(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A})(\boldsymbol{w}_1+\boldsymbol{w}_2)+\frac{1}{2}(b_1+b_2). \end{aligned} \tag{20}$$

An intuitive geometric interpretation for nonlinear TSVR is shown in Fig. 2.

In short, compared with SVR, TSVR is composed of two smaller QPPs. This strategy makes TSVR approximately four times faster than SVR.

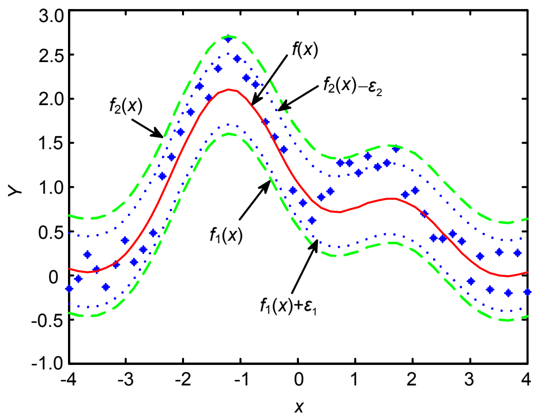

Fig. 2 The geometric interpretation for kernel twin support vector regression (TSVR)

3 Primal least squares twin support vector regression

In this section, in the spirit of LSTSVM (Zhong *et al.*, 2012), introducing the least squares method, the inequality constraints of conventional TSVR are modified to equality constraints. Furthermore, we directly solve the primal QPPs rather than the dual QPPs. This strategy would reduce the computational complexity of TSVR.

For the linear case, using the least squares method, the inequality constraints of Eqs. (9) and (10) are replaced with equality constraints as follows:

$$\begin{aligned} \min\ & \frac{1}{2}\left\|\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}-(\boldsymbol{A}\boldsymbol{w}_1+b_1\boldsymbol{e})\right\|^2+\frac{1}{2}C_1\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi} \\ & \text{s.t.}\ \boldsymbol{Y}-(\boldsymbol{A}\boldsymbol{w}_1+b_1\boldsymbol{e})=\varepsilon_1\boldsymbol{e}-\boldsymbol{\xi}, \end{aligned} \tag{21}$$

$$\begin{aligned} \min\ & \frac{1}{2}\left\|\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}-(\boldsymbol{A}\boldsymbol{w}_2+b_2\boldsymbol{e})\right\|^2+\frac{1}{2}C_2\boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\eta} \\ & \text{s.t.}\ (\boldsymbol{A}\boldsymbol{w}_2+b_2\boldsymbol{e})-\boldsymbol{Y}=\varepsilon_2\boldsymbol{e}-\boldsymbol{\eta}. \end{aligned} \tag{22}$$

Note that in QPP (21) the square of 2-norm of slack variables $\boldsymbol{\xi}$ with weight $C_1/2$ is used, instead of 1-norm of $\boldsymbol{\xi}$ with weight C_1 as used in Eq. (9), which makes the constraint $\boldsymbol{\xi}\geq\boldsymbol{0}$ redundant. We can write the solution of QPP (21) as a solution of a simultaneous system of linear equations through this simple modification. Substituting the equality constraints into the objective function, QPP (21) becomes

$$\begin{aligned} L(\boldsymbol{w}_1,b_1,\boldsymbol{\xi}) = \min\Bigg(& \frac{1}{2}\left\|\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}-(\boldsymbol{A}\boldsymbol{w}_1+b_1\boldsymbol{e})\right\|^2 \\ & +\frac{1}{2}C_1\left\|(\boldsymbol{A}\boldsymbol{w}_1+b_1\boldsymbol{e})+\varepsilon_1\boldsymbol{e}-\boldsymbol{Y}\right\|^2\Bigg). \end{aligned} \tag{23}$$

Setting the gradient of Eq. (23) with respect to $\boldsymbol{w}_1$ and b_1 to zero, gives

$$\begin{aligned}\frac{\partial L(\boldsymbol{w}_1,b_1,\boldsymbol{\xi})}{\partial \boldsymbol{w}_1}&=-\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{w}_1-b_1\boldsymbol{e}-\varepsilon_1\boldsymbol{e})\\&\quad+C_1\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{w}_1+b_1\boldsymbol{e}+\varepsilon_1\boldsymbol{e}-\boldsymbol{Y})=\boldsymbol{0},\end{aligned} \tag{24}$$

$$\begin{aligned}\frac{\partial L(\boldsymbol{w}_1,b_1,\boldsymbol{\xi})}{\partial b_1}&=-\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{w}_1-b_1\boldsymbol{e}-\varepsilon_1\boldsymbol{e})\\&\quad+C_1\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{w}_1+b_1\boldsymbol{e}+\varepsilon_1\boldsymbol{e}-\boldsymbol{Y})=0.\end{aligned} \tag{25}$$

Arranging Eqs. (24) and (25) in matrix form and solving for $\boldsymbol{w}_1$ and b_1, we have

$$\begin{aligned}&-\begin{bmatrix}\boldsymbol{A}\\\boldsymbol{e}\end{bmatrix}^{\mathrm{T}}\left((\boldsymbol{Y}-\varepsilon_1\boldsymbol{e})-[\boldsymbol{A}\ \ \boldsymbol{e}]\begin{bmatrix}\boldsymbol{w}_1\\b_1\end{bmatrix}\right)\\&+C_1\begin{bmatrix}\boldsymbol{A}^{\mathrm{T}}\\\boldsymbol{e}^{\mathrm{T}}\end{bmatrix}\left([\boldsymbol{A}\ \ \boldsymbol{e}]\begin{bmatrix}\boldsymbol{w}_1\\b_1\end{bmatrix}-(\boldsymbol{Y}-\varepsilon_1\boldsymbol{e})\right)=0.\end{aligned} \tag{26}$$

Defining $\boldsymbol{G}=[\boldsymbol{A}\ \boldsymbol{e}]$, $\boldsymbol{f}=\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}$, and $\boldsymbol{u}_1=[\boldsymbol{w}_1\ b_1]^{\mathrm{T}}$, we can obtain

$$-\boldsymbol{G}^{\mathrm{T}}\boldsymbol{f}+\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{u}_1+C_1\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{u}_1-C_1\boldsymbol{G}^{\mathrm{T}}\boldsymbol{f}=0. \tag{27}$$

Then we have

$$\boldsymbol{u}_1=\begin{bmatrix}\boldsymbol{w}_1\\b_1\end{bmatrix}=(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{f}. \tag{28}$$

Notice that $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$ is always positive semi-definite. To overcome this problem, a regularization term $\omega\boldsymbol{I}$ is introduced, where ω is a very small positive number, e.g., $\omega=10^{-5}$. Therefore, Eq. (28) can be rewritten as follows:

$$\boldsymbol{u}_1=(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\omega\boldsymbol{I})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{f}. \tag{29}$$

In a similar way, QPP (22) can be rewritten as

$$\begin{aligned}L(\boldsymbol{w}_2,b_2,\boldsymbol{\eta})=\min\bigg(&\frac{1}{2}\|\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}-(\boldsymbol{A}\boldsymbol{w}_2+b_2\boldsymbol{e})\|^2\\&+\frac{1}{2}C_2\|\boldsymbol{Y}-(\boldsymbol{A}\boldsymbol{w}_2+b_2\boldsymbol{e})+\varepsilon_2\boldsymbol{e}\|^2\bigg).\end{aligned} \tag{30}$$

Defining $\boldsymbol{h}=\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}$, the solution of Eq. (30) is obtained as follows:

$$\boldsymbol{u}_2=\begin{bmatrix}\boldsymbol{w}_2\\b_2\end{bmatrix}=(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{h}. \tag{31}$$

To overcome the ill-conditioning case, a regularization term $\omega\boldsymbol{I}$ is introduced. Then Eq. (31) can be modified to

$$\boldsymbol{u}_2=\begin{bmatrix}\boldsymbol{w}_2\\b_2\end{bmatrix}=(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\omega\boldsymbol{I})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{h}. \tag{32}$$

Once $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are obtained from Eqs. (29) and (32), respectively, the two up- and down-bound functions $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are obtained. Then the estimated regressor is constructed as follows:

$$\begin{aligned}f(\boldsymbol{x})&=\frac{1}{2}(f_1(\boldsymbol{x})+f_2(\boldsymbol{x}))\\&=\frac{1}{2}(\boldsymbol{w}_1+\boldsymbol{w}_2)\boldsymbol{x}^{\mathrm{T}}+\frac{1}{2}(b_1+b_2).\end{aligned} \tag{33}$$

For clarity, we list our linear PLSTSVR algorithm as follows:

Algorithm 1 Linear primal least squares twin support vector regression

Input: the datasets.

Output: the results of linear primal least squares support vector regression.

Step 1: Define $\boldsymbol{G}=[\boldsymbol{A}\ \boldsymbol{e}]$, $\boldsymbol{f}=\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}$, and $\boldsymbol{h}=\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}$.

Step 2: Select suitable up- and down-bound parameters ε_1, ε_2 and regularization term ω.

Step 3: Determine parameters $\boldsymbol{w}_1$, b_1, $\boldsymbol{w}_2$, b_2 of two non-parallel functions using Eqs. (29) and (32).

Step 4: Obtain the estimated regressor $f(\boldsymbol{x})$ using Eq. (33).

Note that in Eqs. (29) and (32) the linear PLSTSVR solves the problem with two matrix inverses of much smaller dimensional matrix of order $(n+1)\times(n+1)$ where $n<<l$. Furthermore, compared with the dual QPPs of TSVR (Eqs. (11) and (12)), in PLSTSVR there are not any constraint conditions (Eqs. (29) and (32)). This means that the learning speed of PLSTSVR is higher than that of TSVR, especially when dealing with large data sets.

For the nonlinear case, according to a similar idea, we consider the following functions with kernel:

$$\begin{cases}f_1(\boldsymbol{x})=K(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_1+b_1,\\f_2(\boldsymbol{x})=K(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_2+b_2,\end{cases} \tag{34}$$

where $K(\boldsymbol{x}^{\mathrm{T}}, \boldsymbol{A}^{\mathrm{T}})$ is any arbitrary kernel. Using the square of 2-norm of slack variables $\boldsymbol{\xi}$ with weight

$C_1/2$ instead of 1-norm of $\boldsymbol{\xi}$ with weight C_1, the inequality constraints are replaced with equality constraints as follows:

$$\begin{aligned}\min\ &\frac{1}{2}\left\|\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_1+b_1\boldsymbol{e})\right\|^2+\frac{1}{2}C_1\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}\\ \text{s.t. }&\boldsymbol{Y}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_1+b_1\boldsymbol{e})=\varepsilon_1\boldsymbol{e}-\boldsymbol{\xi},\end{aligned}\tag{35}$$

$$\begin{aligned}\min\ &\frac{1}{2}\left\|\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_2+b_2\boldsymbol{e})\right\|^2+\frac{1}{2}C_2\boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\eta}\\ \text{s.t. }&(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_2+b_2\boldsymbol{e})-\boldsymbol{Y}=\varepsilon_2\boldsymbol{e}-\boldsymbol{\eta}.\end{aligned}\tag{36}$$

Substituting the equality constraints into the objective function, Eqs. (35) and (36) become

$$\begin{aligned}\min\Bigg(&\frac{1}{2}\left\|\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_1+b_1\boldsymbol{e})\right\|^2\\ &+\frac{1}{2}C_1\left\|K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_1+b_1\boldsymbol{e}+\varepsilon_1\boldsymbol{e}-\boldsymbol{Y}\right\|^2\Bigg),\end{aligned}\tag{37}$$

$$\begin{aligned}\min\Bigg(&\frac{1}{2}\left\|\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_2+b_2\boldsymbol{e})\right\|^2\\ &+\frac{1}{2}C_2\left\|\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}-(K(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}})\boldsymbol{w}_2+b_2\boldsymbol{e})\right\|^2\Bigg).\end{aligned}\tag{38}$$

Similar to linear PLSTSVR, we can obtain the solutions of Eqs. (37) and (38) as follows:

$$\begin{bmatrix}\boldsymbol{w}_1\\ b_1\end{bmatrix}=(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E})^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{f},\tag{39}$$

$$\begin{bmatrix}\boldsymbol{w}_2\\ b_2\end{bmatrix}=(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E})^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{h}.\tag{40}$$

If $\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}$ is irreversible, we can introduce a regularization term to overcome this problem. Then Eqs. (39) and (40) can be derived to

$$\begin{bmatrix}\boldsymbol{w}_1\\ b_1\end{bmatrix}=(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}+\omega\boldsymbol{I})^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{f},\tag{41}$$

$$\begin{bmatrix}\boldsymbol{w}_2\\ b_2\end{bmatrix}=(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}+\omega\boldsymbol{I})^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{h},\tag{42}$$

where $\boldsymbol{E}=[K(\boldsymbol{A},\ \boldsymbol{A}^{\mathrm{T}})\ \boldsymbol{e}]$, and ω is a small positive number. Once the parameters $\boldsymbol{w}_1$, b_1, $\boldsymbol{w}_2$, b_2 of two non-parallel functions are obtained, the regression function $f(\boldsymbol{x})$ can be determined. We now give the steps of the nonlinear PLSTSVR as follows:

Algorithm 2 Nonlinear primal least squares twin support vector regression

Input: the datasets.

Output: the results of nonlinear primal least squares support vector regression.

Step 1: Select a suitable kernel function K.

Step 2: Define $\boldsymbol{E}=[K(\boldsymbol{A},\ \boldsymbol{A}^{\mathrm{T}})\ \boldsymbol{e}]$, $\boldsymbol{f}=\boldsymbol{Y}-\varepsilon_1\boldsymbol{e}$, and $\boldsymbol{h}=\boldsymbol{Y}+\varepsilon_2\boldsymbol{e}$.

Step 3: Choose the suitable up- and down-bound parameters ε_1, ε_2 and regularization term ω.

Step 4: Determine parameters $\boldsymbol{w}_1$, b_1, $\boldsymbol{w}_2$, b_2 of two non-parallel functions using Eqs. (41) and (42).

Step 5: Obtain the estimated regressor $f(\boldsymbol{x})$.

Compared with linear PLSTSVR, the nonlinear PLSTSVR solves the problem with two matrix inverses of order $(l+1)\times(l+1)$, where l is the number of training samples. However, compared with the QPPs of nonlinear TSVR, in nonlinear PLSTSVR there are not any constraint conditions (Eqs. (37) and (38)). This means that the learning speed of nonlinear PLSTSVR is higher than that of the nonlinear TSVR.

4 Experiments and discussion

To check the performance of PLSTSVR, we compared it with SVR, LS-SVR, and TSVR on several datasets, including one group of artificial datasets and 10 UCI datasets. To further verify its validity, PLSTSVR was used to predict the opening price of stock in a final experiment. All the regression algorithms were implemented in the MATLAB 7.11 (R2010b) environment on Windows XP running on a PC with 1 GB of RAM. In this study, we considered only the nonlinear case with the Gaussian kernel for all datasets. Like other machine learning methods, the learning performances of these algorithms are very sensitive to the choice of parameters. In all our experiments, we set $C_1=C_2$ and $\varepsilon_1=\varepsilon_2$ in TSVR to reduce the computational complexity of parameter selection. The parameters for all algorithms were selected over the range $\{2^i|i=-7, -6, \ldots, 6, 7\}$. We evaluated the regression accuracy using 10-fold cross-validation.

We give the evaluation criteria as follows. Define l as the number of training samples, and m as the number of testing samples. Let y_i and $\hat{y}_i$ be the real value and predicted value of sample x_i, respectively, and $\bar{y}=\frac{1}{m}\sum_{i=1}^{m}y_i$. We use the following criteria for algorithm evaluation:

SSE: sum squared error of testing, which is defined as $\text{SSE}=\sum_{i=1}^{m}(y_i-\hat{y}_i)^2$. In most cases, the smaller is the SSE, the fitter is the estimation.

SST: sum squared deviation of testing samples, which is defined as $\text{SST}=\sum_{i=1}^{m}(y_i-\overline{y})^2$. SST represents the underlying variance of the testing samples.

SSR: sum squared deviation, defined as $\text{SSR}=\sum_{i=1}^{m}(\hat{y}_i-\overline{y})^2$. SSR reflects the explanation ability of the regression algorithm.

SSE/SST: ratio between SSE and SST, defined as $\text{SSE}/\text{SST}=\sum_{i=1}^{m}(\hat{y}_i-y_i)^2 \Big/ \sum_{i=1}^{m}(y_i-\overline{y})^2$. The smaller is the SSE/SST, the better is the agreement between estimations and real values.

SSR/SST: ratio between SSR and SST, defined as $\text{SSR/SST}=\sum_{i=1}^{m}(\hat{y}_i-\overline{y})^2 \Big/ \sum_{i=1}^{m}(y_i-\overline{y})^2$.

Obtaining a smaller SSE/SST is usually accompanied by an increase in SSR/SST (Peng, 2010a).

4.1 Artificial dataset

The sin$c$ function is usually used to test the performance of the regression methods:

$$y=\sin c(x)=\frac{\sin(\pi x)}{\pi x},\ \ x\in[-4,\ 4].$$

To effectively check the performance of PLSTSVR, training data points were perturbed by some different types of noises, including the Gaussian noises and the uniformly distributed noises. The four types of the sin$c$ function with noises are listed as follows:

Type 1: $y_i=\frac{\sin(\pi x_i)}{\pi x_i}+\xi_i, x\sim U[-4,4], \xi_i\sim N(0,0.1^2)$,

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

Type 3: $y_i=\frac{\sin(\pi x_i)}{\pi x_i}+\xi_i, x\sim U[-4,4], \xi_i\sim U[-0.1,0.1]$,

Type 4: $y_i=\frac{\sin(\pi x_i)}{\pi x_i}+\xi_i, x\sim U[-4,4], \xi_i\sim U[-0.2,0.2]$,

where $U[a,\ b]$ represents the uniformly random variable in $[a, b]$ and $N(c, d^2)$ represents the Gaussian random variable with mean c and variance d^2.

To improve the reliability of results comparison, for each type of noise, 15 independent groups of noisy samples were randomly generated. Each group of noisy samples contained 500 training samples and 500 test samples. The test points were uniformly sampled from the sin$c$ function without any noise. Table 1 lists the average results of SVR, LS-SVR, TSVR, and PLSTSVR with 15 independent runs. Figs. 3a–3d show the one-run simulation results of SVR, LS-SVR, TSVR, and PLSTSVR for four kinds of noise.

Table 1 shows that TSVR and PLSTSVR obtained smaller SSE, SSE/SST, and larger SSR/SST than SVR and LS-SVR. PLSTSVR derived the smallest SSE, SSE/SST for types 1, 2, and 4 of noise. Furthermore, the learning CPU time of PLSTSVR was less than that of TSVR. This indicates that PLSTSVR obtained a higher learning speed but its regression performance was not reduced compared with TSVR. Fig. 3 shows that the fitting capacity of PLSTSVR is perfect.

4.2 Benchmark datasets

For further evaluation, 10 benchmark datasets were tested. These datasets were usually used to validate the performances of regression methods. For these 10 datasets, the regression results were obtained using 10-fold cross-validation. Table 2 shows the average results of SVR, LS-SVR, TSVR, and PLSTSVR with 15 independent runs on 10 benchmark datasets.

Table 2 shows that TSVR and PLSTSVR had better SSE, SSE/SST, and SSR/SST values than the other two algorithms. For most datasets, the results of PLSTSVR were close to those of TSVR; for some datasets the results of PLSTSVR were even better than those of TSVR. The most important point is that the learning speed of PLSTSVR was much higher than that of TSVR. Specifically, for large data sets, the advantage of PLSTSVR was more obvious. These experimental results indicated that PLSTSVR can obtain not only satisfactory solutions but also good learning speed.

Table 1 Result comparisons of SVR, LS-SVR, TSVR, and PLSTSVR on sin *c* datasets with different types of noise*

Noise	Algorithm	SSE	SSE/SST	SSR/SST	Time (s)
Type 1	SVR	0.3327±0.1524	0.0061±0.0031	0.9528±0.0321	24.561
	LS-SVR	0.2851±0.1321	0.0057±0.0052	0.9534±0.1242	0.034
	TSVR	0.2554±0.0347	0.0055±0.0024	0.9715±0.0347	6.824
	PLSTSVR	0.2456±0.1102	0.0051±0.0013	0.9734±0.0568	0.019
Type 2	SVR	1.1952±0.3462	0.0221±0.0038	0.9428±0.0221	35.678
	LS-SVR	0.9984±0.5241	0.0249±0.0056	0.9214±0.0131	0.061
	TSVR	0.9637±0.1473	0.0175±0.0034	0.9484±0.0352	9.031
	PLSTSVR	0.8754±0.0351	0.0154±0.0021	0.9521±0.0432	0.032
Type 3	SVR	0.0856±0.0474	0.0016±0.0003	0.9989±0.0154	18.350
	LS-SVR	0.1235±0.0571	0.0018±0.0171	0.9874±0.0234	0.098
	TSVR	0.0693±0.0143	0.0013±0.0223	1.0023±0.0062	4.026
	PLSTSVR	0.0734±0.0257	0.0014±0.1501	1.0015±0.0035	0.073
Type 4	SVR	0.5848±0.1356	0.0157±0.0025	0.9438±0.0123	19.240
	LS-SVR	0.5764±0.0478	0.0142±0.0041	0.9524±0.0347	0.083
	TSVR	0.5245±0.1361	0.0098±0.0012	0.9526±0.0512	2.741
	PLSTSVR	0.5221±0.0397	0.0096±0.1351	0.9527±0.0068	0.065

* Average results of 15 independent runs

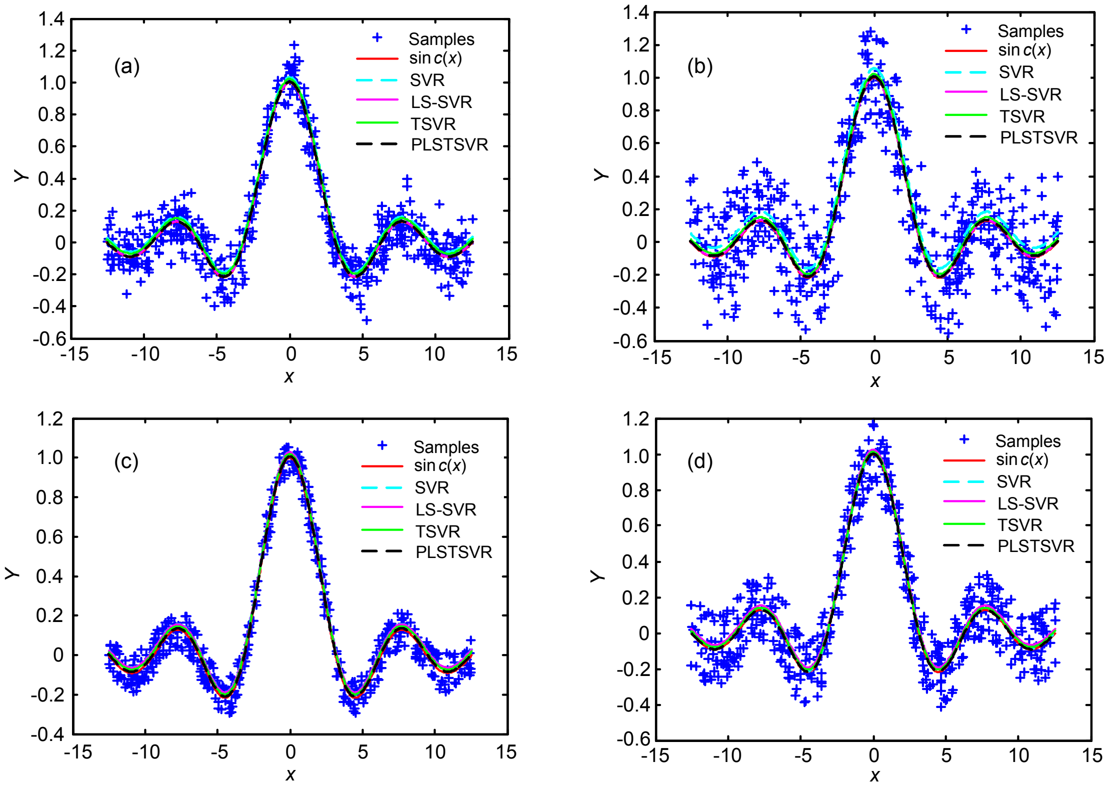

Fig. 3 Predictions of SVR, LS-SVR, TSVR, and PLSTSVR on the sin *c* function with noise $N(0, 0.1^2)$ (a), $N(0, 0.2^2)$ (b), $U[-0.1, 0.1]$ (c), or $U[-0.2, 0.2]$ (d)

References to color refer to the online version of this figure

Table 2 Result comparisons of SVR, LS-SVR, TSVR, and PLSTSVR on 10 benchmark datasets*

Dataset	Algorithm	SSE	SSE/SST	SSR/SST	Time (s)
Diabetes (43×3)	SVR	0.4507±0.0521	0.5192±0.5673	0.6014±0.0174	3.474
	LS-SVR	0.4516±0.1572	0.5271±0.4956	0.5896±0.1176	0.018
	TSVR	0.3688±0.1983	0.4165±0.5801	0.6108±0.0515	0.982
	PLSTSVR	0.3469±0.0024	0.4037±0.2016	0.6224±0.0242	0.017
Boston Housing (506×14)	SVR	0.4056±0.0174	0.1287±0.0352	0.9056±0.1547	69.842
	LS-SVR	0.4051±0.1554	0.1298±0.0648	0.9158±0.1245	0.095
	TSVR	0.3974±0.2547	0.1235±0.0362	1.0257±0.1025	20.712
	PLSTSVR	0.3689±0.4521	0.1201±0.0057	1.0147±0.0356	0.083
Auto MPG (392×8)	SVR	0.1247±0.1578	0.1141±0.0314	0.9875±0.0145	29.241
	LS-SVR	0.1287±0.2412	0.1063±0.0527	0.9964±0.0574	0.045
	TSVR	0.0867±0.0234	0.1125±0.0423	1.0108±0.0012	7.254
	PLSTSVR	0.0825±0.0178	0.1053±0.0402	1.0251±0.0213	0.032
Machine CPU (209×9)	SVR	0.1024±0.1742	0.1049±0.0741	0.9678±0.0052	16.522
	LS-SVR	0.0985±0.5644	0.1019±0.0121	0.9660±0.0044	0.145
	TSVR	0.0854±0.5214	0.1112±0.0741	0.9801±0.0024	3.987
	PLSTSVR	0.0785±0.1458	0.1027±0.0754	0.9763±0.0042	0.107
Servo (167×4)	SVR	0.2542±0.0547	0.1415±0.0089	0.9542±0.0085	12.549
	LS-SVR	0.2684±0.1201	0.1442±0.0078	0.9616±0.0046	0.098
	TSVR	0.2248±0.0145	0.1284±0.0021	0.9748±0.0012	3.821
	PLSTSVR	0.2274±0.0147	0.1257±0.0027	0.9945±0.0011	0.084
Concrete Cs (1030×9)	SVR	0.1985±0.1457	0.1245±0.0052	0.9452±0.2454	215.254
	LS-SVR	0.1904±0.0124	0.1221±0.0078	0.9362±0.2452	1.252
	TSVR	0.1720±0.1701	0.1204±0.0041	0.9541±0.2103	51.362
	PLSTSVR	0.1717±0.0124	0.1198±0.0052	0.9502±0.1024	1.163
Auto price (159× 16)	SVR	0.2104±0.4556	0.1289±0.0056	0.9894±0.0145	45.260
	LS-SVR	0.2015±0.4178	0.1282±0.0034	0.9970±0.1212	0.087
	TSVR	0.1824±0.2014	0.1224±0.0004	1.0024±0.0412	10.850
	PLSTSVR	0.1987±0.4781	0.1240±0.0012	0.9987±0.0451	0.085
Bodyfat (252×15)	SVR	0.0147±0.0142	0.0462±0.0451	0.9634±0.0854	36.242
	LS-SVR	0.0124±0.0589	0.0478±0.0742	0.9952±0.0784	1.058
	TSVR	0.0098±0.0412	0.0451±0.0322	0.9985±0.0542	8.923
	PLSTSVR	0.0089±0.0128	0.0456±0.0363	0.9878±0.0821	0.985
Triazines (186×61)	SVR	0.2471±0.1472	0.3601±0.0522	0.9421±0.0854	8.524
	LS-SVR	0.2247±0.1240	0.3568±0.0451	0.9326±0.0742	0.074
	TSVR	0.2081±0.1041	0.3245±0.0152	0.9685±0.0717	2.129
	PLSTSVR	0.2012±0.1204	0.3018±0.0745	0.9701±0.0745	0.072
Chwirut1 (214×2)	SVR	0.0145±0.0741	0.0784±0.0521	0.9821±0.0142	22.725
	LS-SVR	0.0142±0.0521	0.0741±0.0747	0.9745±0.0121	0.103
	TSVR	0.0138±0.0458	0.0727±0.0544	0.9954±0.0452	5.582
	PLSTSVR	0.0135±0.0147	0.0705±0.0575	0.9924±0.0521	0.998

* Average results of 15 independent runs

4.3 Application on predicting the opening price of stock

In the stock market, the effective prediction of the market index can provide strong information for us to observe the overall change in the stock market. In this subsection, the stock market index data was obtained from Great Wisdom Stock Software (http://www.gw.com.cn), which includes 5279 trading days data from 1990.12.19 to 2012.07.31. The stock market index data contains six indicators, which are the opening price, the highest index value, the minimum index value, closing index, trading volume, and trading turnover. Among them, the opening price data of 5279 trading days is as shown in Fig. 4.

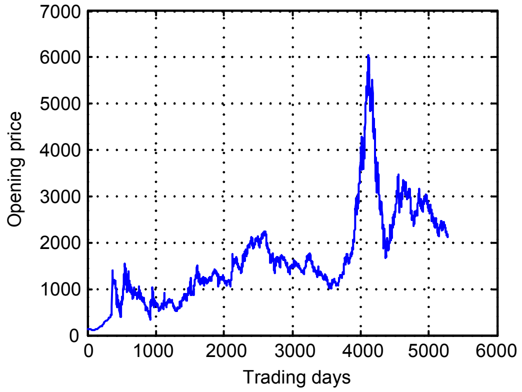

Fig. 4 The opening price during 1990.12.19–2012.07.31

To predict the opening price, we first establish the regression model based on the following assumption: The opening price data of the day is related to the highest index value, the minimum index value, closing index, trading volume, and trading turnover of the day before; i.e., the opening price of the day is the dependent variable and the other five factors are the independent variables. Table 3 shows the average results of SVR, LS-SVR, TSVR, and PLSTSVR with 15 independent runs on the stock market index data. Figs. 5a and 5b show the fitting results of the opening price using TSVR and PLSTSVR, respectively.

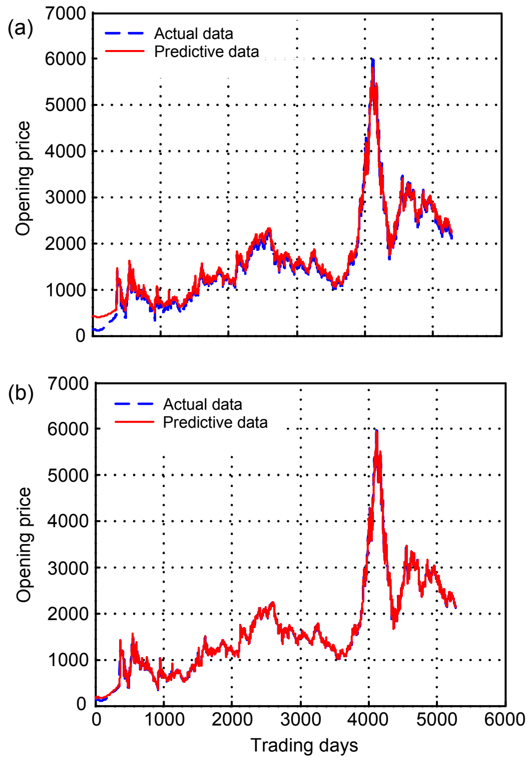

Fig. 5 The fitting results of TSVR (a) and PLSTSVR (b) on stock opening price during 1990.12.19–2012.07.31

Table 3 shows that PLSTSVR has lower SSE, SSE/SST values and higher SSR/SST values than other algorithms. Furthermore, in terms of learning, PLSTSVR is faster than the other three algorithms. This indicates that our algorithm is an effective method for predicting the stock opening price. Fig. 5 shows that the fitting result of PLSTSVR is better than that of TSVR.

5 Conclusions

The classical TSVR obtains its solutions by solving a pair of QPPs with inequality constraints in the dual space, which may reduce the generalization ability especially when dealing with large datasets.

Table 3 Result comparisons of SVR, LS-SVR, TSVR, and PLSTSVR on stock opening price data*

Algorithm	SSE	SSE/SST	SSR/SST	Time (s)
SVR	0.0045±0.174	0.0182±0.0215	0.9829±0.0112	6.251
LS-SVR	0.0043±0.189	0.0187±0.0546	0.9821±0.0195	0.062
TSVR	0.0031±0.189	0.0164±0.0236	0.9904±0.0257	1.540
PLSTSVR	0.0029±0.189	0.0159±0.0124	1.0021±0.0041	0.048

* Average results of 15 independent runs

In this paper, to improve the learning speed of TSVR, we directly solve the QPPs of TSVR in the primal space. By introducing the least squares method, the inequality constraints of TSVR are transformed into equality constraints. Furthermore, we substitute the equality constraints into the objective function of TSVR, and obtain two optimization problems without constraints. Finally, we directly solve them in the primal space. Based on this idea, we propose a new algorithm called primal least squares TSVR (PLSTSVR). In terms of learning speed, the experimental results on the artificial dataset and 10 UCI benchmark datasets have shown that PLSTSVR compares favorably with SVR, LS-SVR, and TSVR, while PLSTSVR also has a good generalization ability. Finally, PLSTSVR is used to predict the opening price of stock. Experimental results show that PLSTSVR is an effective method in stock prediction.

References

Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A Training Algorithm for Optimal Margin Classifiers. Proc. 5th Annual Workshop on Computational Learning Theory, p.144-152. [doi:10.1145/130385.130401]

Chen, Z.Y., Fan, Z.P., 2012. Distributed customer behavior prediction using multiplex data: a collaborative MK-SVM approach. *Knowl.-Based Syst.*, **35**:111-119. [doi:10.1016/j.knosys.2012.04.023]

Cong, H.H., Yang, C.F., Pu, X.R., 2008. Efficient Speaker Recognition Based on Multi-class Twin Support Vector Machines and GMMs. IEEE Conf. on Robotics, Automation and Mechatronics, p.348-352. [doi:10.1109/RAMECH.2008.4681433]

Ding, S.F., Qi, B.J., 2012. Research of granular support vector machine. *Artif. Intell. Rev.*, **38**(1):1-7. [doi:10.1007/s10462-011-9235-9]

Ding, S.F., Su, C.Y., Yu, J.Z., 2011. An optimizing BP neural network algorithm based on genetic algorithm. *Artif. Intell. Rev.*, **36**(2):153-162. [doi:10.1007/s10462-011-9208-z]

Huang, H.J., Ding, S.F., 2012. A novel granular support vector machine based on mixed kernel function. *Int. J. Dig. Cont. Technol. Its Appl.*, **6**(20):484-492. [doi:10.4156/jdcta.vol6.issue20.52]

Jayadeva, Khemchandani, R., Chandra, S., 2007. Twin support vector machines for pattern classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**(5):905-910. [doi:10.1109/TPAMI.2007.1068]

Liu, Y., Yang, J., Li, L., Wu, W., 2012. Negative effects of sufficiently small initial weights on back-propagation neural networks. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **13**(8):585-592. [doi:10.1631/jzus.C1200008]

Mangasarian, O.L., Wild, E.W., 2006. Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(1):69-74. [doi:10.1109/TPAMI.2006.17]

Moraes, R., Valiati, J.F., Neto, W.P.G., 2013. Document-level sentiment classification: an empirical comparison between SVM and ANN. *Exp. Syst. Appl.*, **40**(2):621-633. [doi:10.1016/j.eswa.2012.07.059]

Osuna, E., Freund, R., Girosi, F., 1997. An Improved Training Algorithm for Support Vector Machines. Proc. IEEE Workshop on Neural Networks for Signal Processing, p.276-285. [doi:10.1109/NNSP.1997.622408]

Pan, H., Zhu, Y.P., Xia, L.Z., 2013. Efficient and accurate face detection using heterogeneous feature descriptors and feature selection. *Comput. Vis. Image Understand.*, **117**(1):12-28. [doi:10.1016/j.cviu.2012.09.003]

Peng, X.J., 2010a. Primal twin support vector regression and its sparse approximation. *Neurocomputing*, **73**(16-18):2846-2858. [doi:10.1016/j.neucom.2010.08.013]

Peng, X.J., 2010b. TSVR: an efficient twin support vector machine for regression. *Neur. Networks*, **23**(3):365-372. [doi:10.1016/j.neunet.2009.07.002]

Platt, J.C., 1999. Using Analytic QP and Sparseness to Speed Training of Support Vector Machines. *In*: Kearns, M., Solla, S., Cohn, D. (Eds.), Advances in Neural Information Processing Systems 11. MIT Press, Cambridge, MA, p.557-563.

Suykens, J.A.K., van de Walle, J., 2001. Optimal control by least squares support vector machines. *Neur. Networks*, **14**(1):23-35. [doi:10.1016/S0893-6080(00)00077-0]

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York. [doi:10.1007/978-1-4757-2440-0]

Wu, J.X., 2012. Efficient HIK SVM learning for image classification. *IEEE Trans. Image Process.*, **21**(10):4442-4453. [doi:10.1109/TIP.2012.2207392]

Xu, X.Z., Ding, S.F., Shi, Z.Z., Zhu, H., 2012. A novel optimizing method for RBF neural network based on rough set and AP clustering algorithm. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **13**(2):131-138. [doi:10.1631/jzus.C1100176]

Zhang, X.S., Gao, X.B., Wang, Y., 2009. Twin support vector machines for MCs detection. *J. Electron. (China)*, **26**(3):318-325. [doi:10.1007/s11767-007-0211-0]

Zhong, P., Xu, Y.T., Zhao, Y.H., 2012. Training twin support vector regression via linear programming. *Neur. Comput. Appl.*, **21**(2):399-407. [doi:10.1007/s00521-011-0525-6]