An Efficient Twin Projection Support Vector Machine for Regression

Xinyu Ouyang, Nannan Zhao, Chuang Gao, and Lidong Wang

Abstract—**Taking motivation from ε-insensitive twin support vector regression (ε-TSVR) and the projection idea, this paper proposes a novel ε-twin projection support vector regression models, called ε-TPSVR. The proposed ε-TPSVR, which is based on ε-TSVR, determines the regression function through a pair of nonparallel hyperplanes solved by two smaller sized quadratic programming problems. Different from ε-TSVR, a projection axis is sought for each optimization problem of ε-TPSVR such that the variance of the projected points is minimized. Therefore, the empirical correlation coefficient between each hyperplane and the projected inputs can be optimized. The experimental results indicate that the proposed ε-TPSVR obtains the better prediction performance than TSVR and ε-TSVR methods that were widely adopted.**

Index Terms—**Support vector machine (SVM), Regression analysis, Projection algorithms, Benchmark testing, ε-twin support vector regression (ε-TSVR)**

I. Introduction

Support vector machine (SVM) [1-3] has become a powerful tool in pattern classification and regression field for its excellent generalization performance, and has been successfully applied to various real-world problems [4-6]. As for support vector regression model (SVR), which is a standard tool in regression tasks, there are some classical methods, such as ε-support vector regression (ε-SVR) [2], the sequential minimal optimization (SMO) [7], smooth SVR [8], the parametric insensitive SVR [9], PSO-SVR [10], and ν-support vector classification (ν-SVC) [11].

One of the main challenges for the SVR is the high cost of training, i.e. $O(m^3)$, where m is the number of training samples. Motivated by the researches on twin support vector machine (TSVM) [12-15], Peng [16] proposed a twin support vector regression (TSVR), in which the complexity of algorithm is reduced to $O(2\times(m/2)^3)$, that means the TSVR is four times faster than the usual SVR in theory. Different from SVR, the TSVR method generates two nonparallel functions by solving a pair of smaller sized quadratic programming problems (QPPs) instead of a large size one, so the algorithm TSVR is much faster than a usual SVR. Consequently, many kinds of twin-type SVR have been studied extensively. For example, Shao *et al.* [17] proposed an ε-insensitive twin support vector regression (ε-TSVR), in which the structural risk minimization principle is implemented by introducing the regularization term in the primal problems. R. Rastogi *et al.* [18] presented another twin model for regression termed as ν-TSVR, which extended the ε-TSVR techniques by automatically adjusting the value $\varepsilon1$ and $\varepsilon2$ via user defined parameters $\nu1$ and $\nu2$, so that the fraction of errors and support vectors can be controlled.

However, the previous algorithms and models are very suitable for the assumption that the noise level is uniform on training data, or at least. For the heteroscedastic noise structure, in which the amount of noise depends on location or input, the assumption of a uniform noise level is not satisfied. Recently, motivated by the par-ν-SVR [19], Peng [20] presented a novel parametric- insensitive SVR model for data regression, termed as twin parametric insensitive SVR (TPISVR). Although both par-ν-SVR and TPISVR are more suitable for the case that the noise is more heteroscedastic than the classical SVR, the difference between them is that the strategy TPISVR is to solve two smaller sized QPPs rather than one large QPP, which makes it learn faster than par-v-SVR and SVR. Later, another efficient twin projection support vector regression (TPSVR) algorithm proposed by Peng [21] improved TPISVR through seeking a suitable projection axis such that the empirical variance of projected points is minimized.

In addition, it is interesting to note that, in the ε-TSVR model, the regularization term $(w_k^T w_k + b_k^2)/2, k = 1,2$ is added to the primal optimization problems because of the structural risk minimization principle. The regularization terms in the TPSVR model, however, is $w_k^T w_k/2, k = 1,2$. Because minimizing the projection zone of input points is unrelated to the offset b_k, b_k^2 has to be removed from the regularization term in order to getting the solutions of optimization problems correctly. This implies that the method of solving b_k has to be deduced separately, which will increase the complexity of the model. In fact, what we expect is that the augmented vector $u_k = [w_k^T \quad b_k]^T$ is directly solved, not directly obtained w_k. To this end, inspired by ε-TSVR and the idea of projection, an ε-twin projection support vector machine for regression is proposed in this paper, which is termed as ε-TPSVR. Some of the major features of the proposed ε-TPSVR are as follows:

Manuscript received June 4, 2018; revised November 3. This work was supported by the National Natural Science Foundation of China under Grants 61304002 and 61403177.

Xinyu Oyang is with School of International Finance & Banking, University of Science and Technology Liaoning, Anshan, CO 114051 China. (corresponding author: e-mail: ouyangxinyu@ ustl.edu.cn).

Nannan Zhao is with School of Electronics and Information Engineering, University of Science and Technology Liaoning, Anshan, CO 114051 China. (e-mail: zhaonannan@ustl.edu.cn).

Chuang Gao is with School of Electronics and Information Engineering, University of Science and Technology Liaoning, Anshan, CO 114051 China. (e-mail: gaoyouwen@ustl.edu.cn).

Lidong Wang is with School of Electronics and Information Engineering, University of Science and Technology Liaoning, Anshan, CO 114051 China. (e-mail: wangld5600@ustl.edu.cn).

(1) The proposed ε-TPSVR model aims to seek a suitable projection axis such that the empirical variance of projected points is minimized. It is more suitable for the assumption that the noise level is heteroscedastic. In addition, because the propose method is based on the ε-TSVR model, the structural information of data is then embedded into the regression model.

(2) The Introduction of matrixes J and K makes it possible to solve the augmented vector $u_k = [w_k^T \quad b_k]^T$. It also reduces computational complexity. For the definition of matrixes J and K, see section III for details.

(3) In terms of generalization performance, the results of the experiment have indicated that the performance obtained by the proposed ε-TPSVR is superior to that of other classical twin-type SVR algorithms for the artificial dataset with the heteroscedastic error structure and UCI datasets.

The rest texts of paper are organized as follows. Section II introduces notations used in this paper and briefly describes ε-TSVR and the projection axis. Section III proposes the linear ε-TPSVR model and its extension for the non-linear case. Numerical experiments have been done on both synthetic and real-world datasets and their results have been compared with TSVR, ε-TSVR and ν-TSVR in Section IV, and section V contains concluding remarks.

II. Background

In this section, we first give a brief description of ε-twin support vector regression (ε-TSVR) [17] and then introduce the concept of projection axis [21]. Without loss of generality, given a training set $T = \{(x_i, y_i)\},\ i = 1,2,\ldots,l,$ where $x_i \in R^n$ and $y_i \in R$. Also let $A = (A_1; A_2; \ldots; A_l)$ be the input training sample, and $Y = (A_1; y_2; \ldots; y_l)$ be the response of the training samples, where A is a $l \times n$ matrix, the i-th row A_i represents the i-th training sample, Y is a $l \times 1$ vector, and y_i represents the i-th response.

A. ε-Twin Support Vector Regression

Following the idea of TWSVM and TSVR, Shao *et al.* [15] proposed an approach termed as ε-twin support vector regression (ε-TSVR). Just like any other twin-type SVR, it also finds two insensitive proximal linear function: $f_k(x) = \mathrm{w}_k^T x + b_k,\ k = 1,2$. By introducing the regularization terms $(w_k^T w_k + b_k^2)/2$ and the slack variables $\xi,\ \xi^*,\ \eta$ and η^*, the primal problems can be expressed as

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

and

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

where $c_1, c_2, c_3, c_4,\ \varepsilon_1$ and ε_2 are positive parameters, e is column vector of 'ones' of appropriate dimension. The main difference between TSVR and ε-TSVR is an extra regularization term $(w_k^T w_k + b_k^2)/2, k = 1,2,$ in (1) and (2), thus the structural risk minimization principle is implemented [17].

In order to get the solutions of problems (1) and (2), their dual problems need to be derived. By introducing the Lagrangian multiplies α and γ, the dual problems of the ε-TSVR are given by

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\alpha^T G(G^T G + c_3 I)^{-1} G^T \alpha - Y^T G(G^T G + c_3 I)^{-1} G^T \alpha + (e^T \varepsilon_1 + Y^T)\alpha, \\ &s.t. \quad 0 \leq \alpha \leq c_1 e, \end{aligned} \tag{3}$$

and

$$\begin{aligned} &\min_{\gamma} \frac{1}{2}\gamma^T G(G^T G + c_4 I)^{-1} G^T \gamma + Y^T G(G^T G + c_4 I)^{-1} G^T \gamma - (Y^T - e^T \varepsilon_2)\gamma, \\ &s.t. \quad 0 \leq \gamma \leq c_2 e, \end{aligned} \tag{4}$$

where $G = [A \quad e]$. Once the QPPs (3) and (4) are solved, u_1, u_2 and the final regressor function $f(x)$ can be obtained as follows

$$u_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = (G^T G + c_3 I)^{-1} G^T (Y - \alpha), \tag{5}$$

$$u_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (G^T G + c_4 I)^{-1} G^T (Y + \gamma), \tag{6}$$

$$f(x) = \frac{1}{2}\big(f_1(x) + f_2(x)\big) = \frac{1}{2}(w_1 + w_2)^T x + \frac{1}{2}(b_1 + b_2)\,. \tag{7}$$

B. Projection Axis

In the twin-type SVR, there is usually a pair of insensitive bound function $f_k(x) = \mathrm{w}_k^T x + b_k, k = 1,2$, i.e. the insensitive up- and down-bound functions by solving a pair of smaller sized QPPs. In order to obtain a suitable w_k, the variance of $w_k^T x_i + b_k - y_i$ are supposed to be as small as possible. In the other word, in each QPP, it finds a projection axis, denoted as $\widehat{w}_k = [w_k^T \quad -1]^T, k = 1,2,$ such that each points will be projected on the projection axis $\widehat{w}_k$, and the projected zone is expected to be minimized . It's important to note here that b_k does not contribute to the variance of $w_k^T x_i + b_k - y_i$ because b_k is a fixed variable. The projected zone for each bound function is minimized as

$$\begin{aligned} &\min_{w_k} \frac{1}{2l}\sum_{i=1}^{l} [w_k^T \quad -1] \begin{bmatrix} (x_i - \overline{x}) \\ y_i - \overline{y} \end{bmatrix} [(x_i - \overline{x})^T \quad y_i - \overline{y}] \begin{bmatrix} w_k \\ -1 \end{bmatrix} \\ &= \frac{1}{2}[w_k^T \quad -1] \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix} \begin{bmatrix} w_k \\ -1 \end{bmatrix} = \frac{1}{2} w_k^T \Sigma_x w_k - w_k^T \Sigma_{xy} + \frac{1}{2}\Sigma_y, k = 1,2, \end{aligned} \tag{8}$$

where Σ_x, Σ_y are the empirical covariance matrices of inputs, and responses, and Σ_{yx} is the empirical covariance matric of the inputs and responses. They are defined respectively as

$$\Sigma_x = \frac{1}{l}\sum_{i=1}^{n}(x_i - \overline{x})\,(x_i - \overline{x})^T = \frac{1}{l}\sum_{i=1}^{l} x_i x_i^{\,T} - \bar{x}\bar{x}^T,$$

$$\Sigma_y = \frac{1}{l}\sum_{i=1}^{n}(y_i - \overline{y})(y_i - \overline{y})^T = \frac{1}{l}\sum_{i=1}^{l} y_i^{\,2} - \bar{y}^2\,,$$

$$\Sigma_{xy} = \frac{1}{l}\sum_{i=1}^{l}(x_i y_i - \bar{x}\bar{y}), \qquad \Sigma_{xy} = \Sigma_{yx}^T,$$

where $\bar{x}$ and $\bar{y}$ are the centroid points of inputs and responses, respectively.

III. ε-Twin Projection Support Vector Regression

In this section, a novel ε-twin projection support vector machine for regression, termed as ε-TPSVR is presented, which is motivated by the ε-TSVR and the idea of projection. Using (8), the optimization problems of (1) and (2) can be rewritten as the following formulations,

$$\begin{aligned} &\min_{\substack{w_1,b_1,\xi,\\ \xi^*,\lambda_1}} \frac{1}{2}c_3(w_1^T w_1 + b_1^2) + \frac{1}{2}\xi^{*T}\xi^* + c_1 e^T \xi + \lambda_1\left(\frac{1}{2}w_1^T \Sigma_x w_1 - \Sigma_{xy}^T w_1\right), \\ &s.t. \quad Y - (Aw_1 + eb_1) = \xi^*, \\ &\qquad\ \ Y - (Aw_1 + eb_1) \geq -\varepsilon_1 e - \xi\ ,\ \ \xi \geq 0, \end{aligned} \tag{9}$$

and

$$\begin{aligned} &\min_{\substack{w_2,b_2,\\ \eta,\eta^*,\lambda_2}} \frac{1}{2}c_4(w_2^T w_2 + b_2^2) + \frac{1}{2}\eta^{*T}\eta^* + c_2 e^T \eta + \lambda_2\left(\frac{1}{2}w_2^T \Sigma_x w_2 - \Sigma_{xy}^T w_2\right), \\ &s.t. \quad (Aw_2 + eb_2) - Y = \eta^*, \\ &\qquad\ \ (Aw_2 + eb_2) - Y \geq -\varepsilon_2 e - \eta\ ,\ \ \eta \geq 0. \end{aligned} \tag{10}$$

To solve (9), the Lagrangian function is given by

$$\mathcal{L} = \frac{1}{2}c_3(w_1^T w_1 + b_1^2) + \frac{1}{2}\big(Y - (Aw_1 + eb_1)\big)^T\big(Y - (Aw_1 + eb_1)\big) + c_1 e^T \xi$$
$$+\lambda_1\left(\frac{1}{2}w_1^T \Sigma_x w_1 - \Sigma_{xy}^T w_1\right) - \alpha^T(Y - (Aw_1 + eb_1) + \epsilon_1 e + \xi) - \beta^T\xi, \quad (11)$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_l)$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_l)$ are the vectors of Lagrange multipliers. The K.T.T necessary and sufficient optimality conditions for (9) are given by

$$\frac{\partial \mathcal{L}}{\partial w_1} = -A^T\big(Y - (Aw_1 + eb_1)\big) + c_3 w_1 + A^T\alpha + \lambda_1 \Sigma_x w_1 - \lambda_1 \Sigma_{xy} = 0, \quad (12)$$

$$\frac{\partial \mathcal{L}}{\partial b_1} = -e^T\big(Y - (Aw_1 + eb_1)\big) + c_3 b_1 + e^T\alpha = 0, \quad (13)$$

$$\frac{\partial \mathcal{L}}{\partial \xi} = c_1 e - \alpha - \beta = 0, \quad (14)$$

$$Y - (Aw_1 + eb_1) \ge -\varepsilon_1 e - \xi, \ \xi \ge 0, \quad (15)$$

$$\alpha^T(Y - (Aw_1 + eb_1) + \epsilon_1 e + \xi) = 0, \ \beta^T\xi = 0, \quad (16)$$

$$\alpha \ge 0, \ \beta \ge 0. \quad (17)$$

Since $\beta \ge 0$, from (14), we have $0 \le \alpha \le c_1 e$. Next, combing (12) and (13) leads to

$$-\begin{bmatrix} A^T \\ e^T \end{bmatrix}\left(Y - [A \quad e]\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right) + c_3\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} A^T \\ e^T \end{bmatrix}\alpha$$
$$+\lambda_1\begin{bmatrix} \Sigma_x & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} - \lambda_1\begin{bmatrix} \Sigma_{xy} \\ 0 \end{bmatrix} = 0, \quad (18)$$

By defining

$$G = [A \quad e], \ u_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}, J = \begin{bmatrix} \Sigma_x & 0 \\ 0 & 0 \end{bmatrix}, K = \begin{bmatrix} \Sigma_{xy} \\ 0 \end{bmatrix}, \quad (19)$$

equation (18) can be rewritten as

$$-G^T(Y - Gu_1) + c_3 u_1 + G^T\alpha + \lambda_1 J u_1 - \lambda_1 K = 0, \quad (20)$$

Then the augmented vector u_1 is given by

$$u_1 = (G^T G + c_3 I + \lambda_1 J)^{-1}(G^T Y - G^T\alpha + \lambda_1 K). \quad (21)$$

Then putting (21) into (11) and using the above K.T.T. conditions, the dual problem of the problem (9) is obtained as

$$\max_{\alpha} -\frac{1}{2}\alpha^T D_1 \alpha + \alpha^T D_1 Y + \lambda_1 \alpha^T G(G^T G + c_3 I + \lambda_1 J)^{-1}K - \alpha^T f_1, \quad (22)$$
$$s.t. \quad 0 \le \alpha \le c_1 e,$$

where $D_1 = \mathrm{G}(G^T G + c_3 I + \lambda_1 J)^{-1}\mathrm{G}^{\mathrm{T}}$ and $f_1 = Y + \epsilon_1 e$ to simplify our expression. In the same way, the dual of the problem (10) can be given as

$$\max_{\alpha} -\frac{1}{2}\gamma^T D_2 \gamma - \gamma^T D_2 Y - \lambda_2 \gamma^T G\big(G^T G + c_4 I + \lambda_2 J\big)^{-1}K + \gamma^T f_2, \quad (23)$$
$$s.t. \quad 0 \le \gamma \le c_2 e,$$

where $D_2 = G\big(G^T G + c_4 I + \lambda_2 J\big)^{-1}G^T$ and $f_2 = Y - \epsilon_2 e$. The augmented vector u_2 is given by

$$u_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (G^T G + c_4 I + \lambda_2 J)^{-1}(G^T Y + G^T\gamma + \lambda_2 K), \quad (24)$$

Once the solutions w_1, b_1, w_2 and b_2 are obtained, the estimated regressor can be constructed as (7). For the case of nonlinear regressors, an appropriately chosen kernel function *K* should be introduced, and the more detailed derivation is similar to the references [17, 21].

IV. EXPERIMENTS

To check the performance of the proposed ε-TPSVR, we compare it with the popular TSVR, ε-TSVR and ν-TSVR in several artificial and benchmark [22] datasets. All the computations are carried out on Windows 7 OS Intel Core i5-4210U CPU(2.4GHz) with 4GB RAM and MATLAB R2014a environment. In order to decrease the computational complexity of parameter selection, we set $c_1 = c_2$, $c_3 = c_4$, $\varepsilon_1 = \varepsilon_1$and $\lambda_1 = \lambda_2$. And Gaussian kernel function defined by $K(a,b) = \exp(-\|a - b\|^2/\rho$ is used to process nonlinear data, where vectors $a, b \in R^n$, and the parameter $\rho \ge 0$. Some commonly-used evaluation criterions [16] shown in Table I are introduced before evaluating the performance of these methods.

To test the regression performance of our proposed ε-TPSVR, $f_1(x) = \sin(x)/x$, $x \in [-4\pi, 4\pi]$ and $f_2(x) = |x^{2/3}|$, $x \in [-4\pi, 4\pi]$ are introduced to generate all artificial datasets. The training data's observed values are polluted by the form $y = f(x) + err$, where the noise $err = (0.5 - |x|/8\pi)e$ depends on input. Variable e is the form of $u[a,b]$ or $N(\mu, \sigma^2)$, where $u[a,b]$ represents the uniformly random variable in $[a,b]$ and $N(\mu, \sigma^2)$ represents the Gaussian random variable with means μ and variance σ^2. It implies that these examples have the heteroscedastic error structure.

TABLE I
PERFORMANCE METRICS AND THEIR CALCULATION

Metric	Calculation
SSE	$\sum_{i=1}^{l}(y_i - \hat{y}_i)^2$
SSR	$\sum_{i=1}^{l}(\hat{y}_i - \bar{y})^2$
SST	$\sum_{i=1}^{l}(y_i - \bar{y})^2$
SSE/SST	$\sum_{i=1}^{l}(y_i - \hat{y}_i)^2/\sum_{i=1}^{l}(y_i - \bar{y})^2$
SSR/SST	$\sum_{i=1}^{l}(\hat{y}_i - \bar{y})^2/\sum_{i=1}^{l}(y_i - \bar{y})^2$

Fig. 1 shows the one run results of ε-TPSVR and our ε-TPSVR on $f_1(x)$ with Gaussian noise with mean zero and standard deviation 0.2, and Fig. 2 makes the same comparison with $f_2(x)$. In Fig. 1 and Fig. 2, the solid line is the noiseless test data, and heteroscedastic noise is added to the test data as training data, which is the cross in the figures. It can be seen from the figures that the heteroscedastic noise different from usual additive noise has the characteristic of depending on location or input, so it's harder to deal with. 260 noisy data are selected as training data, 500 data are selected as test data without noise, and the final regression function is trained by ε-TSVR and our ε-TPSVR. Horizontal coordinate represents the number of training points. In particular, in order to see the regression effect more clearly, the horizontal coordinates only show the range from 50 to 450 in Fig. 1, and the horizontal coordinates only show the range from 200 to 300 in Fig. 2. One of the dotted lines is the regression result of the proposed ε-TPSVR method, and the other dotted line is the regression result of the ε-TSVR method. The SSE values of ε-TSVR and ε-TPSVR algorithms in Fig. 1 are 0.4276 and 0.3674 in order, and the SSE values of them in Fig. 2 are 1.5449 and 1.0983 in order. It can be clearly observed from the figures that the result of ε-TPSVR is closer to the black solid line than the result of ε-TSVR. It is because that some points in the heteroscedastic zone are discarded, whereas ε-TSVR cannot filter out the possible noise points. It is also for this reason that data structure information is embedded in our ε-TPSVR learning process.

Next, the effectiveness of the proposed ε-TPSVR is further verified by comparing it with TSVR, ε-TSVR and ν-TSVR. To fairly compare with the performance of TSVR, ε-TSVR, ν-TSVR and ε-TPSVR, 20 independent groups' data on the two functions with different types of noise are generated randomly using Matlab toolbox, including 260 samples during training and 500 samples during testing for each function. Besides, testing data points are uniformly sampled from the objective function without any noise. The measure results are listed in Table Ⅱ and Table Ⅲ. It is easy to see that our method obtains the smaller SSE values and SSE/SST values than the other methods on these two problems, which indicates that the proposed ε-TPSVR gets better performance than the other algorithms and it is more suitable for the case that the noise is heteroscedastic.

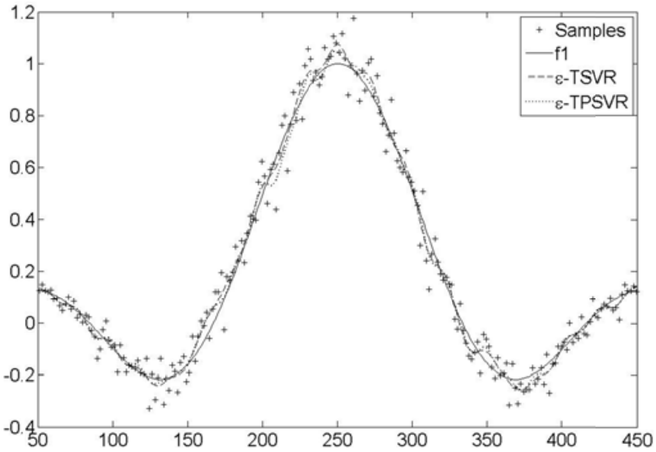

Fig. 1 Predictions of ε-TSVR and ε-TPSVR on $f_1(x)$ with err=$N(0, 0.2^2)$.

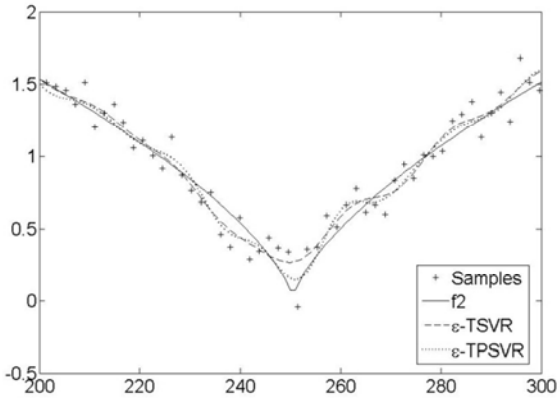

Fig. 2 Predictions of ε-TSVR and ε-TPSVR on $f_2(x)$ with err=$N(0, 0.2^2)$.

In addition, we test three UCI datasets: Servo, Wisconsin breast cancer datasets, Auto-Mpg, Concrete compressive strength, which are usually used in testing machine learning algorithms. To avoid biased comparisons, the standard ten-fold cross-validation is used to compute the optimal values. The results of the performance criteria are listed in Table IV. Also, it can be seen from Table IV that our method outperforms the other methods.

TABLE II
PERFORMANCE COMPARISON ON ARTIFICIAL DATASETS FOR U[-0.2,0.2]

DataSets	Regressor	SSE	SSE/SST	SSR/SST
f1	TSVR	0.1296± 0.0155	0.0020± 0.0002	1.0330± 0.0097
	ε-TSVR	0.1243± 0.0198	0.0019± 0.0003	1.0329± 0.0125
	v-TSVR	0.1240± 0.0196	0.0019± 0.0003	1.0329± 0.0124
	ε-TPSVR	0.1198± 0.0236	0.0019± 0.0004	0.9860± 0.0117
f2	TSVR	0.7261± 0.0757	0.0011± 0.0001	1.0228± 0.0060
	ε-TSVR	0.7549± 0.0421	0.0011± 0.0001	1.0209± 0.0069
	v-TSVR	0.7557± 0.0422	0.0011± 0.0001	1.0210± 0.0070
	ε-TPSVR	0.3254± 0.0572	0.0005± 0.0001	0.9985± 0.0062

TABLE III
PERFORMANCE COMPARISON ON ARTIFICIAL DATASETS FOR $N(0,0.1^2)$

DataSets	Regressor	SSE	SSE/SST	SSR/SST
f1	TSVR	0.1499± 0.0252	0.0023± 0.0004	1.0413± 0.0129
	ε-TSVR	0.1029± 0.0164	0.0016± 0.0003	1.0402± 0.0112
	v-TSVR	0.1031± 0.0165	0.0016± 0.0003	1.0401± 0.0112
	ε-TPSVR	0.0829± 0.0115	0.0013± 0.0002	0.9931± 0.0106
f2	TSVR	0.6911± 0.0573	0.0010± 0.0001	1.0213± 0.0041
	ε-TSVR	0.6511± 0.0556	0.0010± 0.0001	1.0228± 0.0047
	v-TSVR	0.6510± 0.0559	0.0010± 0.0001	1.0229± 0.0048
	ε-TPSVR	0.2485± 0.0393	0.0004± 0.0001	1.0000± 0.0042

V. CONCLUSION

In this paper, a novel ε-twin projection support vector regression models (ε-TPSVR) is proposed. The idea of proposed ε-TPSVR is based on ε-TSVR formulation. However, because the term of projection axis is introduced to maximize the empirical correlation coefficient between the up-or down-bound targets and the projected inputs, our proposed v-TPSVR derives better approximate than other ε-TSVR algorithm. The experimental results on several artificial and UCI datasets show that our proposed method gives similar or better generalization performance with TSVR, ε-TSVR and v-TSVR. Moreover, how to select the optimal hyper parameters and how to determine them rapidly are the difficult problems and should be studied in the following work.

TABLE IV
PERFORMANCE COMPARISON ON UCI DATASETS

DataSets	Regressor	SSE	SSE/SST	SSR/SST
Servo	TSVR	6.2364± 4.7243	0.2959± 0.4411	1.2939± 0.8266
	ε-TSVR	6.6304± 5.3403	0.3048± 0.4539	1.2746± 0.8466
	v-TSVR	6.6304± 5.3402	0.3048± 0.4539	1.2746± 0.8466
	ε-TPSVR	5.5540± 4.1309	0.2533± 0.3334	1.2992± 0.7315
Wisconsin B.C.	TSVR	4.2841± 1.5430	1.0006± 0.3366	0.3918± 0.2123
	ε-TSVR	4.0145± 1.6350	0.9162± 0.2299	0.1615± 0.0725
	v-TSVR	4.0171± 1.6316	0.9167± 0.2273	0.1582± 0.0717
	ε-TPSVR	3.9181± 1.5766	0.8953± 0.2552	0.1903± 0.0752
Auto Mpg	TSVR	1.8759± 0.6674	0.3001± 0.1524	0.8666± 0.1771
	ε-TSVR	1.6526± 0.6477	0.2636± 0.1474	0.7850± 0.1287
	v-TSVR	1.6528± 0.6473	0.2636± 0.1473	0.7850± 0.1287
	ε-TPSVR	1.5957± 0.6112	0.2547± 0.1398	0.7986± 0.1423

REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*. New York: Wiley-Interscience, 1998, pp.493-502.

[2] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, pp. 121-167, 1998.

[3] S. Wen, J.S. Wang; J. Gao. "Fault diagnosis strategy of polymerization kettle equipment based on support vector machine and cuckoo search algorithm", *Engineering Letters*, vol. 25, no. 4, pp. 474-482, 2017.
[4] E. Osuna, R. Freund and F. Girosi, "Training support vector machines: An application to face detection," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp.130-136.
[5] T. Joachims, C. Ndellec and C. Rouveriol, "Text categorization with support vector machines: learning with many relevant features," in European *Conference on Machine Learning No. 10*, Chemnitz, Germany, 1998, pp.137-142.
[6] M.P.S. Brown, W.N. Grundy, D. Lin, et al., "Knowledge-based analysis of microarray gene expression data by using support vector machine,"*Proc. Natl. Acad. Sci. USA,* vol.97, no.1, pp.262–267, 2000.
[7] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, et al., "Improvements to the SMO algorithm for SVM regression," *IEEE Trans. Neural Netw*, vol.11, no.5, pp.1188–1193, 2000.
[8] Y. J. Lee, W-F. Hsieh and C-M. Huang, "ε-SSVR: a smooth support vector machine for ϵ-insensitive regression," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 5, pp. 678-685, 2005.
[9] P. Y. Hao, "New support vector algorithms with parametric insensitive / margin model," *Neural Netw*., vol. 23, pp. 60-73, 2010.
[10] Z.H. Zhong, D.C. Pi, "Forecasting satellite attitude volatility using Support Vector Regression with Particle Swarm Optimization", *IAENG International Journal of Computer Science*, vol.41, no.3, pp. 153-162, 2014.
[11] B. Scholkopf, A. J. Smola, R.C. Williamson and P. L. Bartlett, "New support vector algorithms," *Neural Comput.*, vol. 12, no. 5, pp.1207-1245, 2000.
[12] R. Khemchandani and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905-910, 2007.
[13] Y. H. Shao, C. H. Zhang, X. B. Wang and N. Y. Deng, "Improvements on twin support vector machines," *IEEE Trans. Neural Netw.,* vol. 22, no. 6, pp.962-968, 2011.
[14] S. Mehrkanoon, X. L. Huang and J. A. K. Suykens, "Non-parallel support vector classifiers with different loss functions," *Neuro computing,* vol. 143, pp.294-301, 2014.
[15] Z. Q. Qi, Y. J. Tian and Y. Shi, "Robust twin support vector machine for pattern classification," *Pattern Recognit.,* vol. 46, no.1, pp. 305-316, 2013.
[16] X. J. Peng, "TSVR: an efficient twin support vector machine for regression," *Neural Netw.*, vol. 23, no. 3, pp. 365-372, 2010.
[17] Y. H. Shao, C. H. Zhang, Z. M. Yang, et al., "An ε-twin support vector machine for regression," *Neural Comput. Appl.*, vol. 23, pp.175-185, 2013.
[18] R. Rastogi, P. Anand and S. Chandra, "A *v*-twin support vector machine based regression with automatic accuracy control," *Appl. Intell.,* vol. 46, pp.670-683, 2017.
[19] P. Y. Hao, "New support vector algorithms with parametric insensitive margin model," Neural *Networks,* vol. 23, no. 1, pp. 60-73, 2010.
[20] X. J. Peng, "Efficient twin parametric insensitive support vector regression model," *Neuro computing*, vol.79, pp.26-38, 2012.
[21] X. J. Peng, D. Xu and J. D. Shen, "A twin projection support vector machine for data regression," *Neuro computing*, vol. 138, pp. 131-141, 2014.
[22] C. L. Blake and C. J. Merz. (1998). UCI Repository for Machine Learning Databases, Department of Information and Computer Sciences, University of California, Irvine. [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository. html

XINYU OUYANG was born in Hunan Province, China, in 1974. He received the B.S. and M.S. degrees from Northeastern University and University of Science and Technology Liaoning, China in 1998 and 2003 respectively, and he received Ph.D. degree in Control Theory and Control Engineering from Dalian University of Technology in 2014.

Since 1998 he has been with University of Science and Technology Liaoning, and he is currently a Professor of Automatic Control. He published over 30 scientific papers in diverse fields, some of which were indexed by SCI and EI. His research interests include artificial intelligence, machine learning and digital image processing.

NANNAN ZHAO was born in Liaoning Province, China, in 1975, received the B.S. and M.S. degrees from Northeastern University and University of Science and Technology Liaoning, China in 1998 and 2003 respectively, and she received Ph.D. degree in intelligent control and expert system from Northeastern University in 2006.

She is currently an Associate Professor in the Communication Engineering with University of Science and Technology Liaoning. She has authored over 20 research papers. Her research interests include pattern recognition, machine learning and image processing.

CHUANG GAO was born in Liaoning Province, China, in 1982, received the B.S. and M.S. degrees from Warwick University and King's College London, UK in 2005 and 2007 respectively.

He is currently a Ph.D. candidate in University of Science and Technology Liaoning, China. He has authored over 10 research papers indexed by SCI and EI. His research interests include nonlinear system control, machine learning and intelligent control.

LIDONG WANG was born in Liaoning province, China in 1971, received the B.S. and M.S. degrees from Harbin Institute of Technology and University of Science and Technology Liaoning, China in 1993 and 2004 respectively, and he received Ph.D. degree from Gyeongsang National University, Korea in 2008.

Since 1999 he has been with University of Science and Technology Liaoning, and he is currently an associate professor in Communication Engineering with University of Science and Technology Liaoning. He has authored over 30 research papers. His research interests include pattern recognition, machine learning and digital signal processing.