Appl Intell (2017) 46:670–683
DOI 10.1007/s10489-016-0860-5

A ν-twin support vector machine based regression with automatic accuracy control

Reshma Rastogi[1] · **Pritam Anand**[1] · **Suresh Chandra**[2]

Published online: 29 October 2016

Abstract This paper presents an efficient ν-Twin Support Vector Machine Based Regression Model with Automatic Accuracy Control (ν-TWSVR). This ν-TWSVR model is motivated by the celebrated ν-SVR model (Schlkoff et al. 1998) and recently introduced ϵ-TSVR model (Shao et al., Neural Comput Applic 23(1):175–185, 2013). The ν-TSVR model can automatically optimize the parameters ϵ_1 and ϵ_2 according to the structure of the data such that at most certain specified fraction ν_1(respectively ν_2) of data points contribute to the errors in up (respectively down) bound regressor. The ν-TWSVR formulation constructs a pair of optimization problems which are mathematically derived from a related ν-TWSVM formulation (Peng, Neural Netw 23(3):365–372, 2010) and making use of an important result of Bi and Bennett (Neurocomputing 55(1):79–108, 2003). The experimental results on artificial and UCI benchmark datasets show the efficacy of the proposed model in practice.

✉ Reshma Rastogi
reshma.khemchandani@sau.ac.in

Pritam Anand
ltpritamanand@gmail.com

Suresh Chandra
sureshiitdelhi@gmail.com

[1] Department of Mathematics and Computer Science, South Asian University, New Delhi 110021, India

[2] Department of Mathematics, Indian Institute of Technology Delhi, New Delhi 110016, India

Keywords Support vector machine · Regression · Twin support vector machine · Twin support vector regression · Support vectors

1 Introduction

The last decade has witnessed the evolution of Support Vector Machines (SVMs) as a powerful paradigm for pattern classification and regression [1–4, 15]. SVMs emerged from the research in statistical learning theory on how to regulate the trade off between structural complexity and empirical risk. SVM classifiers attempt to reduce generalization error by maximizing the margin between two separating hyperplanes [1–4].

Support Vector Regression (SVR) is a technique for handling regression problem which is similar in principle to SVM. The standard ϵ-SVR is an ϵ-insensitive model which sets an epsilon tube around the data points. The data points outside the epsilon tube contribute to the errors which are penalized in the objective function via a user specified parameter. Therefore, an appropriate choice of ϵ should be supplied beforehand in order to meet the desired accuracy. Bi and Bennett [5] have developed a geometric framework for SVR showing that it can be related to an appropriate SVM problem. This result of Bi and Bennett [5] is very significant as it allows to explore the possibilities of deriving new regression models corresponding to existing classification models.

Twin Support Vector Machine(TWSVM) [6] is a novel binary classification method that determines two non-parallel hyperplanes, each of them are proximal to its own class and has at least unit distance from the opposite class. TWSVM is faster than SVM as it solves two related

SVM-type problems each of which is smaller than the conventional SVM. The ν-TWSVM [11] model modifies the TWSVM formulation. It finds a pair of the non-parallel hyperplanes, each of them are proximal to its own class and is as far as possible from the points of opposite class.

In an attempt to extend twin methodology to the regression problem, Peng [7] proposed a novel algorithm which he termed as Twin Support Vector Machine for Regression (TSVR). However in a recent paper, Khemchandani et al. [8] observed that the model of Peng [7] is not in the true spirit of twin methodology. They [8] presented a new twin support vector machine based regression and termed it as TWSVR model. It is interesting to note that the TWSVR model can be derived mathematically using an important result of Bi and Bennett [5] together with the TWSVM formulation. Later, Shao et al. [9] presented another twin model for regression termed as ϵ-TSVR. The formulation of the ϵ-TSVR [9] is different from that of Peng [7] and is justified on the basis of different empirical risk functionals. Later, Xu et al. [14] proposed the K-nearest neighbor-based weighted twin support vector regression by using the local information present in the samples for improving the prediction accuracy.

In this paper, we have proposed a novel ν-TWSVM based regression formulation termed as ν-TWSVR. Some of the important features of ν-TWSVR are as follows.

(i) The proposed ν-TWSVR model automatically adjusts the values of ϵ_1 and ϵ_2 in order to achieve the desired accuracy level to the data in hand.

(ii) It provides an efficient way to control the fraction of errors and support vectors by adjusting the values of ϵ_1 and ϵ_2 via the user specified parameters ν_1 and ν_2.

(iii) The proposed ν-TWSVR formulation is in the true spirit of ν-TWSVM [11]. Thus, the two QPPs in the formulation of ν-TWSVR can be mathematically derived using an important result of Bi and Bennett [5].

(iv) As a consequence of our derivation, it can be shown that the ϵ-TSVR formulation of Shao et al. [9] is also in the true spirit of twin methodology and can be derived using Bi and Bennett approach [5].

(v) The experimental results on artificial and UCI-benchmark real world datasets show that ν-TWSVR outperforms ϵ-TSVR model in practice.

The rest of this paper is organized as follows. Section 2 introduces notations used in the paper and briefly describes ν-SVR model [10]. Section 3 discusses the linear ν-TWSVM [11] while Section 4 briefly describes ϵ-TSVR [9]. Section 5 proposes the linear ν-Twin Support Vector Machine Based Regression(ν-TWSVR) and its extension for the non-linear case. Section 6 describes the experimental results while Section 7 is devoted to the conclusions.

2 Support vector regression models

2.1 ϵ-support vector regression

Let the training samples be denoted by a set of l row vectors $A = (A_1, A_2, ..., A_l)$ where the i^{th} sample $A_i = (A_{i1}, A_{i2}, ..., A_{in})$ is in the n-dimensional real space $\mathbb{R}^n$. Also let $Y = (y_1, y_2, ..., y_l)$ denote the response vector of training samples where $y_i \in \mathbb{R}$. Let $\xi = (\xi_1, \xi_2, .., \xi_l)$ and $\xi^* = (\xi_1^*, \xi_2^*, .., \xi_l^*)$ be l dimensional vector which will be used to denote the errors and e is column vector of 'ones' of appropriate dimension.

Linear ϵ-SVR finds a linear function $f(x) = w^T x + b$, where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$. It minimizes the ϵ-insensitive loss function with a regularization term $\frac{1}{2}\|w\|^2$ which makes the function $f(x)$ as flat as possible. The ϵ-insensitive loss function ignores the error up to ϵ. The Support Vector Regression(SVR) solves following optimization problem

$$\begin{aligned}
&\min_{w,b,\xi,\xi^*} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)\\
&\text{subject to,}\\
&y_i - (A_i w + b) \le \epsilon + \xi_i, \ (i = 1, 2, .., l),\\
&(A_i w + b) - y_i \le \epsilon + \xi_i^*, \ (i = 1, 2, .., l),\\
&\xi_i \ge 0, \ \xi_i^* \ge 0 \ , (i = 1, 2, .., l),
\end{aligned} \tag{1}$$

where $C > 0$ is the user specified parameter that balances the trade off between the fitting error and the flatness of the function.

2.2 Support vector regression with automatic accuracy control

In ϵ-SVR, a good choice of ϵ should be specified beforehand in order to meet desired level of accuracy. A bad choice of ϵ can lead to poor accuracy. In [10] author proposes Support Vector Regression with Automatic Accuracy Control ($\nu-$SVR) which automatically minimizes the size of ϵ-tube and adjusts the accuracy level according to the data in hand. For the linear case ν-SVR solves the following optimization problem

$$\begin{aligned}
&\min_{w,b,\epsilon,\xi_1,\xi_2} \quad \frac{1}{2}\|w\|^2 + C\left(\nu.\epsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right)\\
&\text{subject to,}\\
&y_i - (A_i w + b) \le \epsilon + \xi_i, \ (i = 1, 2, .., l),\\
&(A_i w + b) - y_i \le \epsilon + \xi_i^*, \ (i = 1, 2, .., l),\\
&\xi_i \ge 0, \ \xi_i^* \ge 0 \ , (i = 1, 2, .., l).
\end{aligned} \tag{2}$$

The tube size ϵ is traded off against the model complexity and slack variables via a constant $\nu \ge 0$. Using the K.K.T.

optimality conditions, the Wolfe dual of the optimization problem (2) is obtained as follows

$$\min_{\alpha,\alpha^*} \ \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)A_i^T A_j - \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i)$$

subject to,

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0\,, \quad \sum_{i=1}^{l}(\alpha_i^* + \alpha_i) \le C\nu,$$

$$0 \le \alpha_i \le \frac{C}{l}\,, \quad 0 \le \alpha_i^* \le \frac{C}{l}. \tag{3}$$

After finding the optimal values of α and α^*, the estimated regressor for the test point x is given by

$$f(x) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)A_i^T x + b.$$

3 A ν-twin support vector machine for classification

Peng [11] proposed a modification of TWSVM, termed as ν-TWSVM and introduced the two new parameters ν_1 and ν_2 instead of the trade-off parameters c_1 and c_2 of TWSVM. Similar to the ν-SVM [12], the parameters ν_1 and ν_2 in the ν-TWSVM controls the bounds on the number of support vectors and the margin errors. $\nu-$Twin Support Vector Machine (ν-TWSVM) modifies the optimization problem of TWSVM as follows

(ν-TWSVM-1)

$$\min_{w_1,b_1,\rho_+,\xi_1} \ \frac{1}{2}||Aw_1 + e_1 b_1||_2 - \nu_1\rho_+ + \frac{1}{l_2}e_2^T\xi_1$$

subject to,

$$-(Bw_1 + e_2 b_1) \ge e_2\rho_+ - \xi_1,$$

$$\xi_1 \ge 0,\ \rho_+ \ge 0, \tag{4}$$

and

(ν-TWSVM-2)

$$\min_{w_2,b_2,\rho_-,\xi_2} \ \frac{1}{2}||Bw_2 + e_2 b_2||_2 - \nu_2\rho_- + \frac{1}{l_1}e_1^T\xi_2$$

subject to,

$$(Aw_1 + e_1 b_1) \ge e_1\rho_- - \xi_2,$$

$$\xi_2 \ge 0,\ \rho_- \ge 0, \tag{5}$$

where A and B are $(l_1 \times n)$ and $(l_2 \times n)$ matrices representing data points belonging to class +1 (positive) and class -1 (negative) respectively. Here ξ_1 and ξ_2 are l-dimensional error vector. The introduction of the variables ρ_+ and ρ_- provides an efficient way to control the number of support vectors in TWSVM. In practice rather than solving the primal optimization problems (4) and (5), we solve their respective Wolfe duals by using the K.K.T. optimality conditions.

4 An ϵ-twin support vector machine for regression

On the lines of TWSVM, ϵ-Twin Support Vector Machine for Regression(ϵ-TSVR) [9] also finds two ϵ-insensitive proximal functions $f_1(x) = w_1^T x + b_1$ and $f_2(x) = w_2^T x + b_2$, which are obtained by solving following pairs of optimization problems

(ϵ-TSVR-1)

$$\min_{w_1,b_1,\xi} \ \tfrac{c_3}{2}\left(||w_1||^2 + b_1^2\right) + \tfrac{1}{2}||(Y - (Aw_1 + eb_1)||^2 + c_1 e^T\xi$$

subject to,

$$Y - (Aw_1 + eb_1) \ge -\epsilon_1 e - \xi,\ \ \xi \ge 0, \tag{6}$$

and

(ϵ-TSVR-2)

$$\min_{w_2,b_2,\eta} \ \tfrac{c_4}{2}\left(||w_2||^2 + b_2^2\right) + \tfrac{1}{2}||(Y - (Aw_2 + eb_2)||^2 + c_2 e^T\eta$$

subject to,

$$(Aw_2 + eb_2) - Y \ge -\epsilon_2 e - \eta,\ \ \eta \ge 0, \tag{7}$$

where parameters ϵ_1, ϵ_2, c_1, c_2, c_3 and c_4 are user supplied positive numbers. Defining $G = \begin{bmatrix} A, & e \end{bmatrix}$ and using the K.K.T optimality conditions, the Wolfe dual of (ϵ-TSVR-1) can be obtained as follows

(Dual ϵ-TSVR-1)

$$\min_{\alpha} \ \frac{1}{2}\alpha^T G(c_3 I + G^T G)^{-1}G^T\alpha - Y^T G(G^T G + c_3 I)^{-1}G^T\alpha + (e^T\epsilon_1 + Y^T)\alpha$$

$$\text{subject to,}\quad 0 \le \alpha \le c_1 e, \tag{8}$$

where α is vector of Lagrangian multipliers.

The Wolfe dual of the optimization problem (ϵ-TSVR-2) is obtained as

(Dual ϵ-TSVR-2)

$$\min_{\gamma} \ \frac{1}{2}\gamma^T G(c_4 I + G^T G)^{-1}G^T\gamma + Y^T G(G^T G + c_4 I)^{-1}G^T\gamma - (Y^T - e^T\epsilon_2)\gamma$$

$$\text{subject to ,}\quad 0 \le \gamma \le c_2 e, \tag{9}$$

where γ is vector of Lagrange multipliers.

After solving the optimization problems (8) and (9), the augmented vector $v_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$ and $v_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$ are obtained as

$$v_1 = (c_3 I + G^T G)^{-1} G^T (Y - \alpha), \tag{10}$$

and

$$v_2 = (c_4 I + G^T G)^{-1} G^T (Y + \lambda). \tag{11}$$

The final regressor is then obtained by

$$f(x) = \frac{1}{2}(f_1(x) + f_2(x)) = \frac{1}{2}(w_1 + w_2)^T x + \frac{1}{2}(b_1 + b_2). \tag{12}$$

For the non-linear case, the ϵ-TSVR considers following kernel generated surfaces

$$f_1(x) = K(x^T, A^T)u_1 + b_1, \tag{13}$$

and

$$f_2(x) = K(x^T, A^T)u_2 + b_2,$$

which are obtained by solving following pairs of QPPs

(Kernel $\epsilon-$TSVR-1)

$$\min_{u_1, b_1, \xi} \ \tfrac{c_3}{2}\left(||u_1||^2 + b_1^2\right) + \tfrac{1}{2}||(Y - (K(A, A^T)u_1 + eb_1)||^2$$
$$+ c_1 e^T \xi$$

subject to,

$$Y - (K(A, A^T)u_1 + eb_1) \geq -\epsilon_1 e - \xi, \quad \xi \geq 0, \tag{14}$$

and

(Kernel $\epsilon-$TSVR-2)

$$\min_{u_2, b_2, \eta} \ \tfrac{c_4}{2}\left(||u_2||^2 + b_2^2\right) + \tfrac{1}{2}||(Y - (K(A, A^T)u_2 + eb_2)||^2$$
$$+ c_2 e^T \eta$$

subject to,

$$(K(A, A^T)u_2 + eb_2) - Y \geq -\epsilon_2 e - \eta, \quad \eta \geq 0. \tag{15}$$

Similar to the linear case, the (Kernel $\epsilon-$TSVR-1) and (Kernel $\epsilon-$TSVR-2) are solved in their dual forms using the K.K.T. optimality conditions.

5 ν-TWSVM based regression

Similar to ϵ-SVR model, in TWSVR model also, a good choice of ϵ_1 and ϵ_2 should be supplied beforehand in order to meet the desired accuracy. A bad choice of ϵ_1 and ϵ_2 could lead to poor results. The ν-TWSVM based Regression (ν-TWSVR) finds the optimal choices of ϵ_1 and ϵ_2 by trading off the values of ϵ_1 and ϵ_2 in the two respective optimization problems via the user specified parameters ν_1 and ν_2. The value of ν_1 (respectively ν_2) specifies an upper bound on the fraction of points allowed to contribute to the error ξ (respectively η) and specifies a lower bound on the number of support vectors for up (respectively down) bound function.

5.1 Linear ν-TWSVM based regression

Similar to the TSVR model, ν-TWSVR also solves a pair of optimization problems which are as follows

(ν-TWSVR-1)

$$\min_{w_1, b_1, \epsilon_1, \xi} \ \tfrac{c_1}{2}\left(||w_1||^2 + b_1^2\right) + \tfrac{1}{2}||(Y - (Aw_1 + eb_1)||^2$$
$$+ c_2(\nu_1 \epsilon_1 + \tfrac{1}{l} e^T \xi)$$

subject to,

$$Y - (Aw_1 + eb_1) \geq -\epsilon_1 e - \xi,$$
$$\xi \geq 0, \epsilon_1 \geq 0, \tag{16}$$

and

(ν-TWSVR-2)

$$\min_{w_2, b_2, \epsilon_2, \eta} \ \tfrac{c_3}{2}\left(||w_2||^2 + b_2^2\right) + \tfrac{1}{2}||(Y - (Aw_2 + eb_2)||^2$$
$$+ c_4\left(\nu_2 \epsilon_2 + \tfrac{1}{l} e^T \eta\right)$$

subject to,

$$(Aw_2 + eb_2) - Y \geq -\epsilon_2 e - \eta,$$
$$\eta \geq 0, \epsilon_2 \geq 0, \tag{17}$$

where ν_1, ν_2, c_1, c_2, c_3 and c_4 are user specified positive parameters.

In the above optimization problems, the size of ϵ_1 (respectively ϵ_2) tubes is traded off against the model complexity and other terms via positive parameters ν_1 (respectively ν_2) which allows them to automatically adjust according to the structure of data. Preposition-1 gives a theoretical interpretations of parameters ν_1 and ν_2. The derivation of the above formulation of ν-TWSVR from an appropriately constructed ν-TWSVM classification problem is given in Appendix B. This derivation is based on the work of Bi and Bennett [5] and Khemchandani et al. [8].

In order to get the solution of optimization problems (16) and (17), we derive their dual problems. We first consider (ν-TWSVR-1) and its corresponding Lagrangian function

$$L(w_1, b_1, \epsilon_1, \xi, \alpha, \beta, \gamma) = \frac{c_1}{2}\left(||w_1||^2 + b_1^2\right)$$
$$+ \frac{1}{2}||(Y - (Aw_1 + eb_1)||^2 + c_2\left(\nu_1 \epsilon_1 + \frac{1}{l} e^T \xi\right)$$
$$+ \alpha^T(-Y + (Aw_1 + eb_1) - e\epsilon_1 - \xi) - \beta^T \xi - \gamma \epsilon_1 \ , \tag{18}$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_l)$, $\beta = (\beta_1, \beta_2, ..., \beta_l)$ and γ are Lagrange multipliers. The K.K.T. optimality conditions are given by

$$\frac{\partial L}{\partial w_1} = -A^T(Y-(Aw_1+eb_1))+c_1w_1+A^T\alpha = 0, \tag{19}$$

$$\frac{\partial L}{\partial b_1} = -e^T(Y-(Aw_1+eb_1))+c_1b+e^T\alpha = 0, \tag{20}$$

$$\frac{\partial L}{\partial \xi} = \frac{c_2}{l}e - \alpha - \beta = 0, \tag{21}$$

$$\frac{\partial L}{\partial \epsilon_1} = c_2\nu_1 - e^T\alpha - \gamma = 0, \tag{22}$$

$$Y - (Aw_1+eb_1) \geq -e\epsilon_1 - \xi \ , \xi \geq 0, \epsilon_1 \geq 0, \tag{23}$$

$$\alpha^T(Y-(Aw_1+eb_1)+e\epsilon_1+\xi) = 0, \tag{24}$$

$$\beta^T\xi = 0, \gamma\epsilon_1 = 0, \tag{25}$$

$$\alpha \geq 0, \beta \geq 0, \gamma \geq 0. \tag{26}$$

Since $\beta \geq 0$, therefore from (21) and (26) we have

$$0 \leq \alpha \leq \frac{c_2}{l}e. \tag{27}$$

Also, since $\gamma \geq 0$, so (22) would lead to

$$e^T\alpha \leq c_2\nu_1. \tag{28}$$

Consider $G = \begin{bmatrix} A, & e \end{bmatrix}$ and $v_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$. Then combining (19) and (20) would result in following equation

$$-G^TY + (G^TG + c_1I)v_1 + G^T\alpha = 0, \tag{29}$$

or

$$v_1 = (c_1I + G^TG)^{-1}G^T(Y-\alpha). \tag{30}$$

After substituting the value of v_1 in (18) and using the above K.K.T optimality conditions the dual problem of the (ν-TWSVR-1) can be obtained as

(Dual ν-TWSVR-1)

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\alpha^TG(c_1I+G^TG)^{-1}G^T\alpha \\ &-Y^TG(c_1I+G^TG)^{-1}G^T\alpha + Y^T\alpha \\ &\text{subject to,} \\ &0 \leq \alpha \leq \frac{c_2}{l}e, \\ &e^T\alpha \leq c_2\nu_1. \end{aligned} \tag{31}$$

In a similar way, the dual of the problem (ν-TWSVR-2) can be obtained as

(Dual ν-TWSVR-2)

$$\begin{aligned} &\min_{\lambda} \frac{1}{2}\lambda^TG(c_3I+G^TG)^{-1}G^T\lambda \\ &+Y^TG(c_3I+G^TG)^{-1}G^T\lambda - Y^T\lambda \\ &\text{subject to.} \\ &0 \leq \lambda \leq \frac{c_4}{l}e, \\ &e^T\lambda \leq c_4\nu_2, \end{aligned} \tag{32}$$

where λ is the vector of the Lagrange multipliers. The augmented vector $v_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$ is given by

$$v_2 = (c_3I + G^TG)^{-1}G^T(Y+\lambda). \tag{33}$$

After obtaining the solution of the dual problems (31) and (32), the decision variables v_1 and v_2 can be obtained by using (30) and (33) respectively. For the given $x \in \mathbb{R}^n$, the estimated regressor is obtained as follows

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

5.2 ν-TWSVM based regression for non-linear case

For finding the non-linear regressor, the ν-TWSVR considers the following kernel generated functions

$$f_1(x) = K(x^T, A^T)u_1 + b_1,$$

and $\ f_2(x) = K(x^T, A^T)u_2 + b_2,$

where K is an appropriately chosen kernel function. For the non-linear case, the Kernel ν-TWSVR-1 and Kernel ν-TWSVR-2 can be formulated as

(Kernel ν-TWSVR-1)

$$\begin{aligned} &\min_{u_1,b_1,\epsilon_1,\xi} \frac{c_1}{2}(||u_1||^2+b_1^2) + \frac{1}{2}||(Y-(K(A,A^T)u_1+eb_1)||^2 \\ &+c_2(\nu_1\epsilon_1 + \frac{1}{l}e^T\xi) \\ &\text{subject to,} \\ &Y-(K(A,A^T)u_1+eb_1) \geq -\epsilon_1e - \xi, \\ &\xi \geq 0\ ,\ \epsilon_1 \geq 0, \end{aligned} \tag{34}$$

and

Table 1 Performance metrics and their calculations

Metrics	Calculation
RMSE	$= \sqrt{\frac{1}{l}\sum_{i=1}^{l}(y_i - y_i').^2}$
SSE	$= \sum_{i=1}^{l}(y_i - y_i')^2$
SST	$= \sum_{i=1}^{l}(y_i - \overline{y})^2$
SSR	$= \sum_{i=1}^{l}(y_i' - \overline{y})^2$

(Kernel ν-TWSVR-2)

$$\min_{u_2,b_2,\epsilon_2,\eta} \ \frac{c_3}{2}(||u_2||^2+b_2^2)+\frac{1}{2}||(Y-(K(A,A^T)u_2+eb_2)||^2$$
$$+c_4\left(\nu_2\epsilon_2+\frac{1}{l}e^T\eta\right)$$
subject to,
$$(K(A,A^T)u_2+eb_2)-Y \geq -\epsilon_2 e-\eta$$
$$\eta \geq 0\,,\ \ \epsilon_2 \geq 0. \tag{35}$$

Working on the lines similar to the linear case, the dual problems corresponding to the primal optimization problems (Kernel-ν-TWSVR-1) and (Kernel-ν-TWSVR-2) can be obtained as

$$\min_{\alpha} \frac{1}{2}\alpha^T H(c_1I+H^TH)^{-1}H^T\alpha$$
$$-Y^TH(c_1I+H^TH)^{-1}H^T\alpha+Y^T\alpha$$
subject to,
$$0 \leq \alpha \leq \frac{c_2}{l}e,$$
$$e^T\alpha \leq c_2\nu_1, \tag{36}$$

and

$$\min_{\lambda} \frac{1}{2}\lambda^T H(c_3I+H^TH)^{-1}H^T\lambda$$
$$+Y^TH(c_3I+H^TH)^{-1}H^T\lambda-Y^T\lambda$$
subject to,
$$0 \leq \lambda \leq \frac{c_4}{l}e,$$
$$e^T\lambda \leq c_4\nu_2, \tag{37}$$

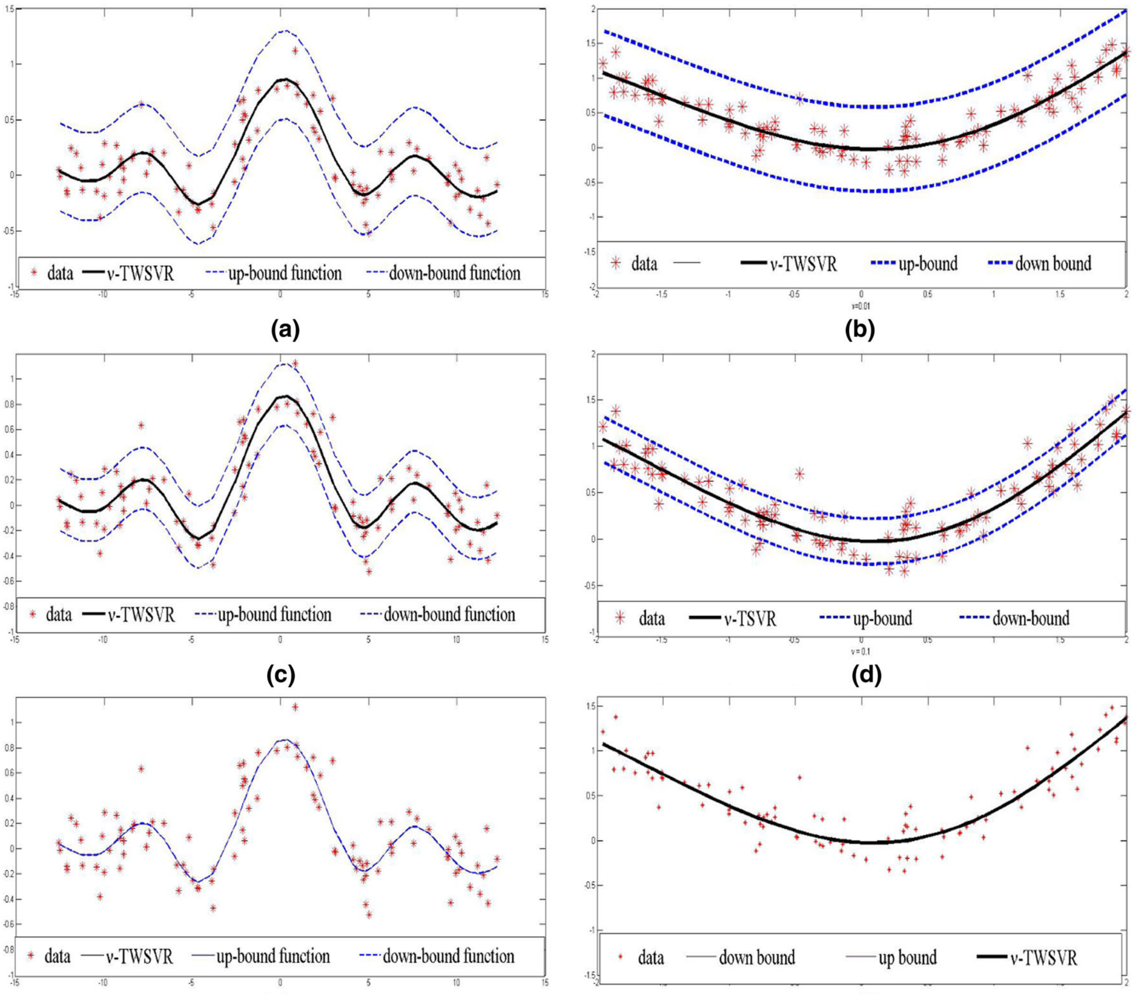
respectively.

Here H= $\left[\ K(A,A^T),\ e\ \right]$ and the augmented vectors $v_1=\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$ and $v_2=\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$ are given as

$$v_1=(c_1I+H^TH)^{-1}H^T(Y-\alpha), \tag{38}$$

and

$$v_2=(c_3I+H^TH)^{-1}H^T(Y+\lambda). \tag{39}$$



(a) (b) (c) (d) (e) (f)

Fig. 1 Performance of ν-TWSVR (a) for ν = 0.01 on Type 1 dataset (b) for ν = 0.01 on Type 2 dataset (c) for ν = 0.1 on Type 1 dataset (d) for ν = 0.1 on Type 2 dataset (e) for ν = 0.9 on Type 1 dataset and (f) for ν = 0.9 on Type 2 dataset

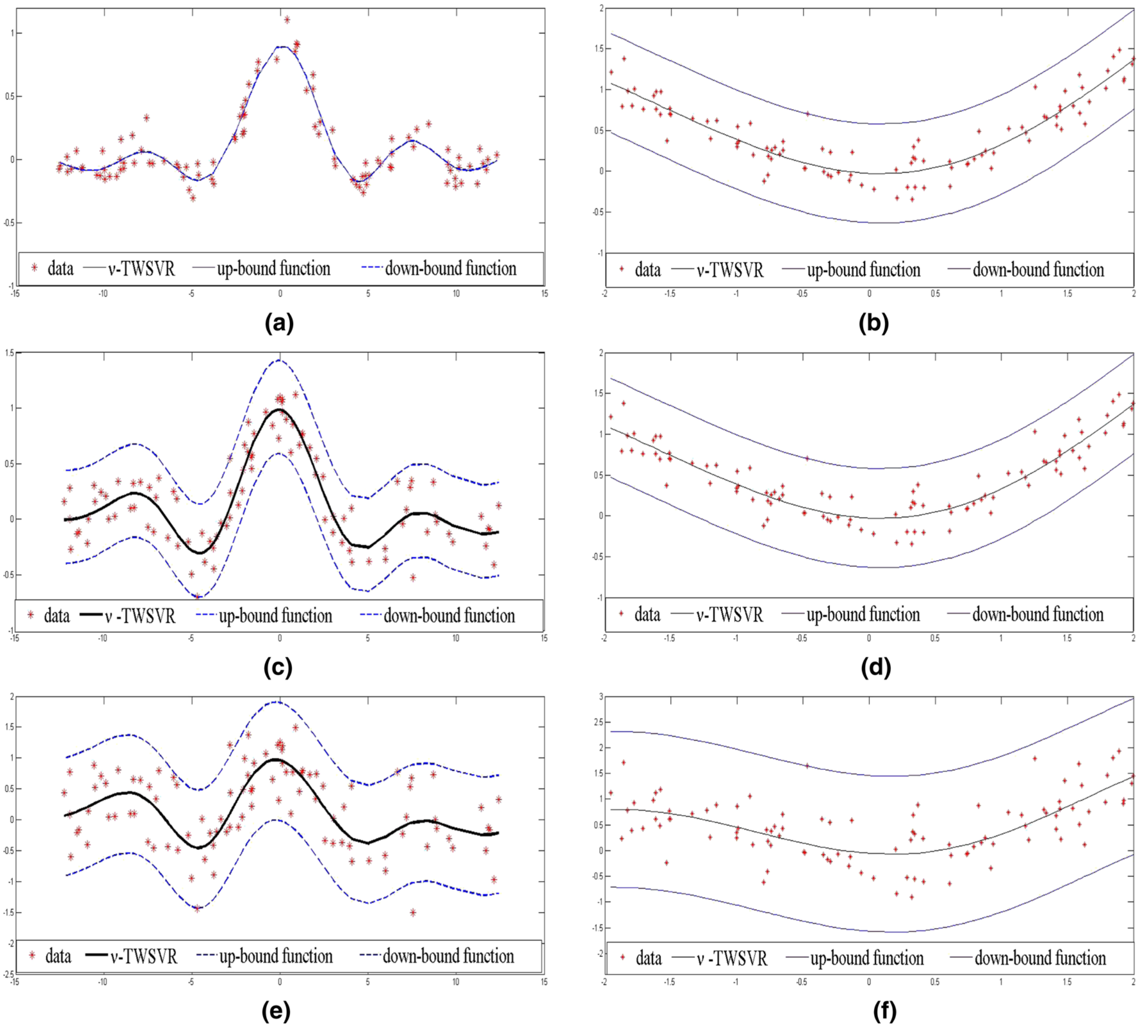


Fig. 2 Performance of ν-TWSVR (a) for σ = 0.01 on Type 1 dataset (b) for σ = 0.01 on Type 2 dataset (c) for σ = 0.2 on Type 1 dataset (d) for σ = 0.2 on Type 2 dataset (e) for σ = 0.5 on Type 1 dataset and (f) for σ = 0.5 on Type 2 dataset for fixed value of ν = 0.01

For a given value $x \in \mathbb{R}^n$, the estimated regressor is obtained as

$$f(x) = \frac{1}{2}(f_1(x) + f_2(x))$$
$$= \frac{1}{2}(u_1 + u_2)^T K(A, x) + \frac{1}{2}(b_1 + b_2).$$

Proposition 1 *Suppose ν-TWSVR is applied on a dataset which results ϵ_1(respectively ϵ_2) > 0, then following statements hold.*

(a) *ν_1(respectively ν_2) is an upper bound on fraction of error ξ(respectively η).*

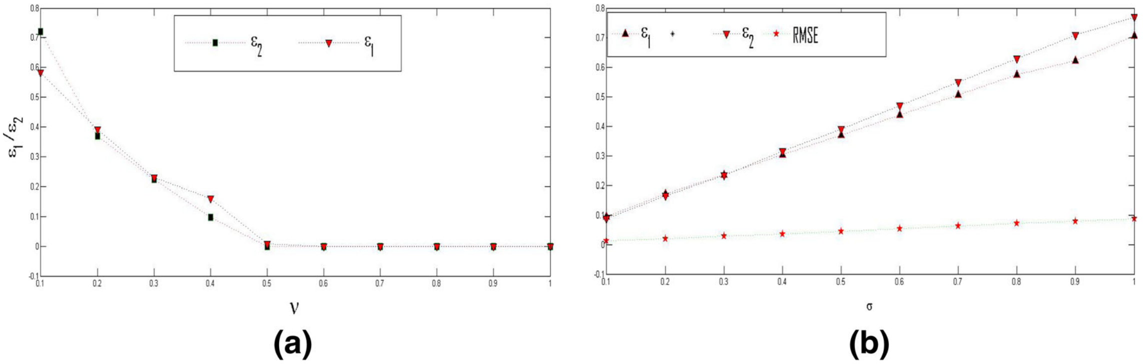


Fig. 3 **a** ν-TWSVR for different value of ν, **b** ν-TWSVR for different value of σ on Type 1 dataset

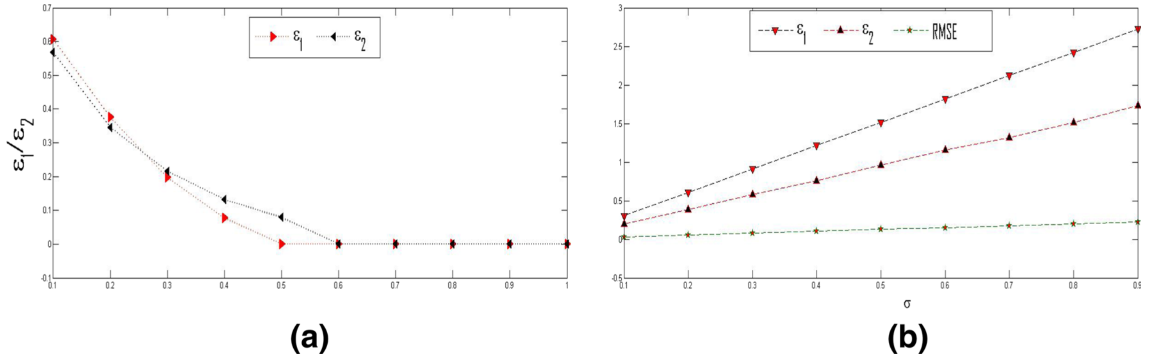


Fig. 4 **a** ν-TWSVR for different value of ν, **b** ν-TWSVR for different value of σ on Type 2 dataset

(b) *ν_1(respectively ν_2) is a lower bound on fraction of support vectors for up bound (respectively down bound) regressor.*

Proof The proof of the proposition is described in Appendix A. □

Remark 1 ν_1(respectively ν_2) can be used to control the number of points allowed to contribute to errors. Furthermore, it has been experimentally observed that if ν_1(respectively ν_2) > 1 then ϵ_1(respectively ϵ_2)= 0. For ν_1(respectively ν_2) < 1, it can still happen that ϵ_1(respectively ϵ_2)= 0 if the data is noise free.

6 Experimental results

This section experimentally verifies claims made in this paper. To show the efficacy of our proposed algorithm, we have considered artificial and UCI-benchmark datasets [13]. We have also compared the proposed algorithm with ϵ-TSVR and shown that the proposed algorithm outperforms it in practice. All the simulations have been performed in Matlab 12.0 environment (http://in.mathworks.com/) on Intel XEON processor with 16.0 GB RAM. Throughout these experiments we have used the RBF kernel $exp\left(\frac{-||x-y||^2}{q}\right)$ where q is the kernel parameter.

One of the issues with the ϵ-TSVR model is that it involves a large number of parameters which should be tunned in order to meet the desired accuracy level. To reduce the computational complexities of parameter selection involved in ϵ-TSVR, we have set $c_1 = c_2$, $c_3 = c_4$ and $\epsilon_1 = \epsilon_2$. We have also set $c_1 = c_3$, $c_2 = c_4$ and $\nu_1 = \nu_2$ for the proposed method for all kinds of experiments. For artificial datasets, we have considered two type of synthetic training dataset (x_i, y_i) for $i = 1, 2, .., l$ as follow

Type 1

$$y_i = \frac{sin(x_i)}{x_i} + \xi_i \quad \xi_i \sim N(0, \sigma) \;\;, \quad x_i \in U[-4\pi, 4\pi].$$

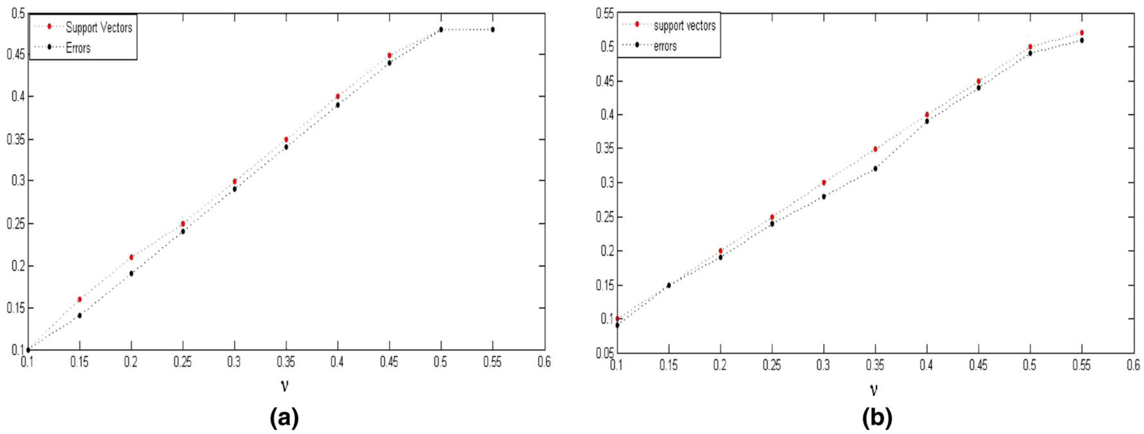


Fig. 5 Plot of ν versus fraction of support vectors and fraction of errors (**a**) for up bound regressor and (**b**) for down bound regressor on Type 1 dataset

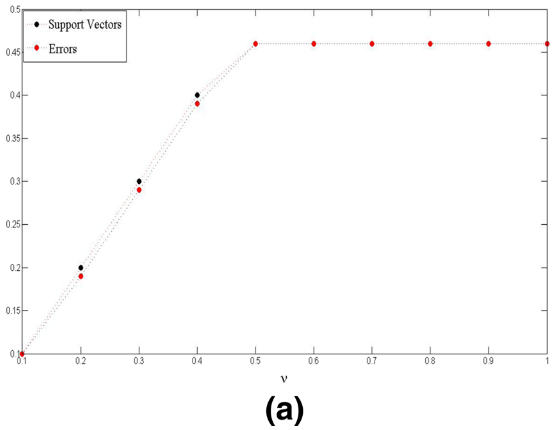


Fig. 6 Plot of ν versus fraction of support vectors and fraction of errors (**a**) for up bound regressor and (**b**) for down bound regressor on Type 2 dataset

Type 2

$$y_i = x_i^{\frac{2}{3}} + \xi_i \;\; \xi_i \sim N(0, \sigma) \;\;, \;\; x_i \in U[-2, 2].$$

Table 1 list the different performance metrics for a regression model where y_i' is the predicted value of y_i and $\overline{y}$ denotes the mean of y_i for i = 1 , 2 ,...l.

To avoid the biased comparison, ten independent groups of noisy samples are generated randomly using the Matlab toolbox for synthetic datasets which consists of 100 training samples and 655 non-noise testing samples. For artificial datasets, we have also solved the primal problem of ν-TWSVR as we have to retrieve the value of ϵ_1 and ϵ_2.

Figure 1 shows the performance of ν-TWSVR for different values of ν on Type 1 and Type 2 synthetic datasets. In the figures, the size of ϵ_1 and ϵ_2 tubes decreases as the value of ν increases. For ν=0.9, the ϵ-tubes almost vanish. Figure 2 shows the performance of ν-TWSVR on Type 1 and Type 2 synthetic datasets for the different noise coefficients $\sigma = 0.01$, $\sigma = 0.2$ and $\sigma = 0.5$. It can be easily seen that for the fixed value of $\nu = 0.01$, the tube width automatically adjust for the different value of σ. Figures 3a and 4a show the plot of ν versus ϵ_1 and ϵ_2 for the Type 1 and Type 2 synthetic datasets respectively. It is evident that as value of ν increases, the value of ϵ_1 and ϵ_2 decreases. In view of the proposition (1), it can be inferred that as the value of ν increases, the proposed method reduces the size of one sided ϵ-tube in order to control the number of points lying outside the ϵ-tubes. Figures 3b and 4b show that for a fixed value of ν, the size of one sided ϵ-tubes increases as more

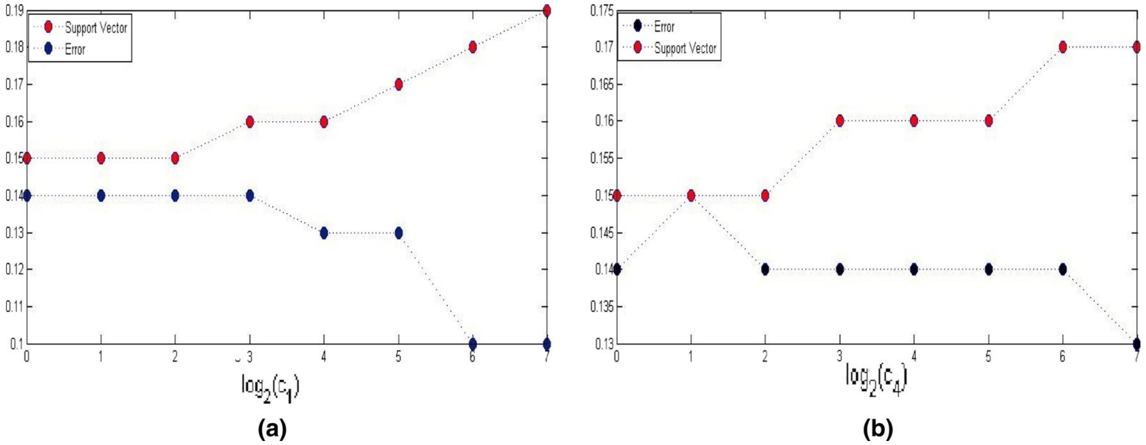


Fig. 7 **a** Plot of fraction of support vectors and fraction of errors versus log c_1 for up bound regressor. **b** Plot of fraction of support vectors and fraction of errors versus log c_2

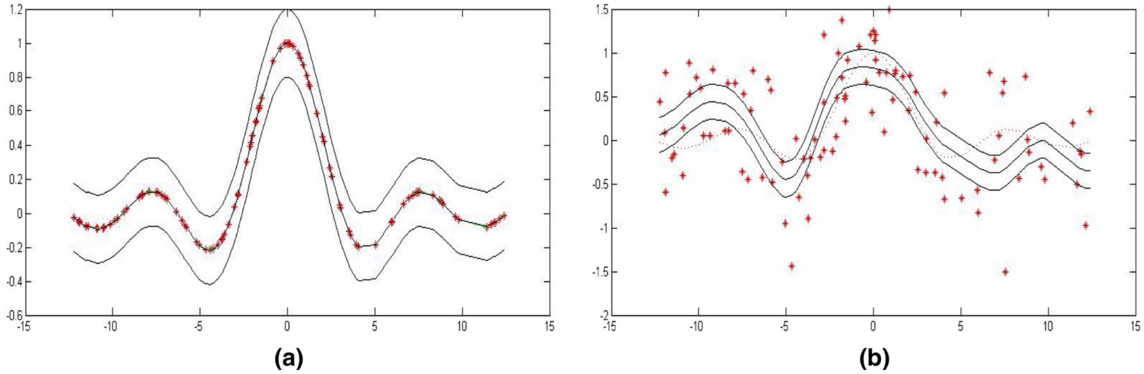


Fig. 8 Performance of ϵ-TSVR for (**a**) $\sigma = 0$ and (**b**) $\sigma = 0.5$ for $\epsilon = 0.2$

Table 2 Result on synthetic dataset type 1

$\nu_1 = \nu_2$	0.0100	0.0500	0.1000	0.1500	0.2000	0.2500	0.3000	0.3500	0.4000	0.4500
ϵ_1	0.7799	t0.6860	0.5333	0.4566	0.3849	0.2995	0.3815	0.1822	0.0581	0.0239
ϵ_2	1.1786	0.6396	0.5888	0.3957	0.3152	0.2520	0.1986	0.1639	0.1046	0.0603
SV1	1.0000	5.0000	10.0000	15.0000	20.0000	25.0000	30.0000	35.0000	40.0000	45.0000
SV2	1.0000	5.0000	10.0000	15.0000	20.0000	25.0000	30.0000	35.0000	40.0000	45.0000
Error1	0.0000	5.0000	10.0000	14.0000	19.0000	24.0000	29.0000	34.0000	39.0000	44.0000
Error2	1.0000	5.0000	9.0000	13.0000	18.0000	23.0000	27.0000	31.0000	38.0000	43.0000
SSE	17.3123	17.3394	17.2985	17.3210	17.2969	17.3728	17.3873	17.3737	17.3732	17.3577
SSE/SST	0.0914	0.0915	0.0913	0.0914	0.0913	0.0917	0.0918	0.0917	0.0917	0.0916
SSR/SST	1.0266	1.0269	1.0264	1.0261	1.0261	1.0265	1.0270	1.0269	1.0265	1.0275

Table 3 Result on synthetic dataset type 2

$\nu_1 = \nu_2$	0.01	0.05	0.1	0.15	0.2	0.25	0.35	0.45
ϵ_1	2.4021	1.5809	1.2009	1.0755	0.6968	0.6010	0.3094	0.0152
ϵ_2	1.8982	1.6023	1.2076	1.0851	0.6605	0.5442	0.3454	0.1576
SV1	1.0000	5.0000	10.0000	15.0000	20.0000	25.0000	35.0000	45.0000
SV2	1.0000	5.0000	10.0000	15.0000	20.0000	25.0000	35.0000	45.0000
Error1	0.0000	4.0000	9.0000	14.0000	19.0000	24.0000	34.0000	44.0000
Error2	0.0000	5.0000	9.0000	14.0000	20.0000	24.0000	34.0000	44.0000
SSE1	8.7497	8.7503	8.7502	8.7503	8.7493	8.7492	8.7493	8.7486
SSE/SST	0.0827	0.0827	0.0827	0.0827	0.0827	0.0827	0.0827	0.0827
SSR/SST	0.8568	0.8568	0.8568	0.8568	0.8568	0.8568	0.8568	0.8568

Table 4 Result on servo dataset

$\nu_1 = \nu_2$	0.0500	0.1000	0.1500	0.2000	0.2500	0.3000	0.3500	0.4000	0.4500	0.5
ϵ_1	0.5757	0.4572	0.3314	0.2389	0.1640	0.1154	0.0789	0.0449	0.0140	0.0124
ϵ_2	0.6430	0.4135	0.3232	0.2575	0.1937	0.1604	0.1093	0.0856	0.0558	0.0258
SV1	8.0000	15.4000	23.0000	30.3000	38.0000	45.3000	53.0000	60.5000	67.7000	69.3
SV2	8.0000	15.3000	23.2000	30.3000	38.2000	45.5000	53.3000	60.5000	68.2000	75.6
Error1	7.0000	14.3000	21.8000	29.6000	36.7000	44.3000	51.4000	59.0000	65.9000	67.9
Error2	7.0000	14.4000	21.8000	29.6000	36.6000	44.3000	51.6000	59.1000	66.7000	74.2
SSE	5.9472	5.9487	5.9479	5.9479	5.9480	5.9480	5.9479	5.9482	5.9481	5.9481
SSE/SST	0.1836	0.1836	0.1836	0.1836	0.1836	0.1835	0.1835	0.1836	0.1836	0.1835

Table 5 Result on machine CPU dataset

$\nu_1 = \nu_2$	0.0100	0.0500	0.1000	0.1500	0.2000	0.2500	0.3000	0.3500	0.4000	0.4500
ϵ_1	0.7799	0.6860	0.5333	0.4566	0.3849	0.2995	0.3815	0.1822	0.0581	0.0239
ϵ_2	1.1786	0.6396	0.5888	0.3957	0.3152	0.2520	0.1986	0.1639	0.1046	0.0603
SV1	1.0000	5.0000	10.0000	15.0000	20.0000	25.0000	30.0000	35.0000	40.0000	45.0000
SV2	1.0000	5.0000	10.0000	15.0000	20.0000	25.0000	30.0000	35.0000	40.0000	45.0000
Error1	0.0000	5.0000	10.0000	14.0000	19.0000	24.0000	29.0000	34.0000	39.0000	44.0000
Error2	1.0000	5.0000	9.0000	13.0000	18.0000	23.0000	27.0000	31.0000	38.0000	43.0000
SSE	17.3123	17.3394	17.2985	17.3210	17.2969	17.3728	17.3873	17.3737	17.3732	17.3577
SSE/SST	0.0914	0.0915	0.0913	0.0914	0.0913	0.0917	0.0918	0.0917	0.0917	0.0916
SSR/SST	1.0266	1.0269	1.0264	1.0261	1.0261	1.0265	1.0270	1.0269	1.0265	1.0275

noise is introduced in the data for Type 1 and Type 2 datasets respectively. It can be also seen that RMSE increases as the more noises is added in the data. Figure 5a and b show the plot of ν versus fraction of support vectors and errors for up and down bound regressor on the Type 1 synthetic dataset respectively. Figure 6a and b show the plot of ν versus fraction of support vectors and errors for up and down bound regressor on the Type 2 synthetic dataset respectively. It verifies claims made in Preposition-1. Figure 7a and b show the plot of fraction of support vectors and errors versus $log_2 c_2$ and $log_2 c_4$ for up bound and down bound regressor receptively for the Type 1 synthetic dataset. It can be observed that the bounds specified in Proposition 1 get loser as the value of c_1 and c_2 increases.

Figure 8 shows the performance of ϵ-TSVR for the fixed value of $\epsilon_1 = 0.2$ and $\epsilon_2 = 0.2$ for the noise coefficient $\sigma = 0$ and $\sigma = 0.5$. It can been seen that the ϵ-TSVR performs well for $\sigma = 0$ but as the noise increases it fails to predict accurately. So in the case of ϵ- TWSVR, a good choice of ϵ_1 and ϵ_2 should be supplied beforehand in order to have accurate result which is not the case of ν-TWSVR.

For further evaluation, we test six UCI benchmark datasets namely, Boston Housing, Auto-Price, Machine-CPU, Servo, Concrete CS. These datasets are commonly used in testing machine learning algorithms. For all datasets, feature vectors were normalized in the $[-1, 1]$. Tables 2 and 3 show the performance of the proposed algorithm on different values of ν on synthetic datasets Type 1 and Type 2 for the noise coefficient $\sigma = 0.5$. Tables 4 and 5 show the results of the proposed algorithm on Servo and Machine-CPU datasets for different value of ν. In these tables following abbreviations are used.

SV1 : Number of Support Vectors for up bound Regressor.

SV2 : Number of Support Vectors for down bound Regressor.

Error1 : Number of points lying outside of the one sided ϵ_1 tube for up bound regressor.

Error2 : Number of points lying outside of the ϵ_2 tube for down bound regressor.

The numerical results of the tables verify the claim made in the Preposition-1. These tables also show how the size of one sided ϵ-tubes decrease as the value of ν increases, which further affect the values of SSE, SSE/SST and SSR/SST.

To compare the proposed algorithm with ϵ-TSVR, we have downloaded the code of ϵ-TSVR from http://www.optimal-group.org/Resource. We have tunned the parameters of ϵ-TSVR and proposed method through the set of values $\{2^i | i = -9, -8, \dots, 10\}$ by tuning a set comprising of random 10 % of the dataset.

Table 6 lists the mean of SSE/SST and SSR/SST of the the proposed method with the ϵ-TSVR on five different UCI-benchmark datasets. For these datasets the ten-fold cross validation method was used to report the results. It can be easily observed that the proposed method outperform the ϵ-TSVR in practice. Table 7 lists our findings about the best value of parameters for the ϵ-TSVR and ν-TWSVR for the above mentioned UCI-datasets.

Table 6 Comparision of proposed ν-TWSVR with ϵ-TSVR on UCI datasets

Datasets	Regressor	SSE/SST	SSR/SST
Servo 167 × 4	ϵ-TSVR	0.2366	0.7394
	ν-TWSVR	0.1909	0.8937
Boston Housing 506 × 14	ϵ-TSVR	0.1224	0.8938
	ν-TWSVR	0.0991	0.9166
Machine CPU 209 × 8	ϵ-TSVR	0.0218	0.9802
	ν-TWSVR	0.0167	0.9733
AutoPrice 205 × 26	ϵ-TSVR	0.1641	0.9065
	ν-TWSVR	0.1468	0.9050
Concrete CS 1030 × 9	ϵ-TSVR	0.1485	0.8583
	ν-TWSVR	0.1404	0.8707

Table 7 Tunned parameters of ϵ-TSVR and ν-TWSVR

Dataset	ϵ-TSVR		–	–	ν-TWSVR	–	–	–
	q	$\epsilon_1 = \epsilon_2$	$c_1 = c_2$	$c_3 = c_4$	q	$\nu_1 = \nu_2$	$c_2 = c_4$	$c_1 = c_3$
Servo	0.5	0.002	128	2	0.5	0.1	0.5	0.1
Machine CPU	4	0.002	256	0.002	4	0.1	0.002	0.001
Boston Housing	4	64	0.5	0.002	2	0.001	1024	0.001
Auto Price	4	0.002	0.001	0.002	4	0.1	0.02	0.001
Concrete CS	2	32	0.002	0.002	2	0.1	0.002	0.002

7 Conclusions

In this paper, we have proposed a ν-Twin Support Vector Machine Based Regression Model (ν-TWSVR) which is capable of automatically optimizing the parameters ϵ_1 and ϵ_2 appearing in the recently proposed ϵ-TSVR model of Shao et al. [9]. It has been proved mathematically that the parameters ν_1 and ν_2 can be used for controlling the fraction of data points which contribute to the errors and support vectors. It has also been shown experimentally that ν-TWSVR outperforms ϵ-TSVR as it automatically adjusts the parameter $\epsilon_1(\epsilon_2)$ according to noise present in the data.

The optimization problems appearing in the ν-TWSVR formulation have been derived by employing an important result of Bi and Bennett [5] which connects a regression problem to an appropriately constructed classification problem. This development also gives a mathematical justification to the ϵ-TSVR formulation and thereby establishes that similar to TWSVR formulation [8], the ϵ-TSVR formulation is also in the true spirit of TWSVM methodology.

Acknowledgments The authors are extremely thankful to the learned referees whose valuable comments have helped to improve the content and presentation of the paper.

Appendix A

Proposition 1 *Suppose ν-TWSVR is applied on a dataset which results ϵ_1(respectively ϵ_2) > 0, then following statements hold.*

(a) *v_1(respectively v_2) is an upper bound on fraction of error ξ(respectively η).*

(b) *v_1(respectively v_2) is a lower bound on fraction of support vectors for up bound (respectively down bound) regressor.*

Proof

(a) Using the KKT conditions (21) and (25) for up bound regressor, we can find that for $\xi_i > 0$, $\beta_i = 0$ and $\alpha_i = \frac{c_2}{l}$.

Since from (22) and (26), $e^T\alpha \leq c_2 v_1$, so there may exist at most $l v_1$ points for which $\xi_i \neq 0$. In the similar way using the K.K.T. optimality conditions for down bound regressor we can prove that there are at most $l v_2$ points for which $\eta_i \neq 0$.

(b) Using the KKT conditions (22) and (25) for $\epsilon_1 \neq 0$ we find that $\gamma = 0$. This implies that $e^T\alpha = c_2 v_1$.

Since $0 \leq \alpha_i \leq \frac{c_2}{l}$ so there must be at least $l v_1$ points for which $\alpha_i \neq 0$. In similar way using the K.K.T conditions for down bound regressor we can prove that there are at least $l v_2$ points for which $\lambda_i \neq 0$.

□

Appendix B: ν-TWSVR via ν-TWSVM

Bi and Bennett [5] have shown the equivalence between a given regression problem and an appropriately constructed classification problem. They have shown that for a given regression training set (A,Y), a regressor $y = w^T x + b$ is an ϵ-insensitive regressor if and only if the set D^+ and D^- locate on different sides of $n+1$ dimensional hyperplane $w^T x - y + b = 0$ respectively where

$$D^+ = \{(A_i, y_i + \epsilon), i = 1, 2, ..., l\}$$
$$D^- = \{(A_i, y_i - \epsilon), i = 1, 2, ..., l\}.$$

In veiw of this result of Bi and Bennett [5], the regression problem is equivalent to the classification problem of sets D^+ and D^- in R^{n+1}. If we use the TWSVM methodology [6] for the classification of these two sets D^+ and D^- then we can find TWSVM based Regression [8]. It is relevant to mention here that the classification of set D^+ and D^- is a special case of classification where we have following privilege informations.

(a) D^+ and D^- classes are symmetric in nature and have equal number of sample points.

(b) Points in the class D^+ and D^- are separated by the distance 2ϵ.

These privileged informations must be exploited for the better classification as better classification of the set D^+ and D^- will eventually lead to better regressor. The classification of the set D^+ and D^- in R^{n+1} using ν-TWSVM results into following QPPs

$$\underset{(w_1,\eta_1,b_1,\rho_+)}{Min} \; \tfrac{c_1}{2}\left(||w_1||^2+\eta_1^2+b_1^2\right)+\tfrac{1}{2}||Aw_1+\eta_1(Y+\epsilon e)+eb_1||^2 -c_2v_1\rho_+$$

subject to,

$$(Aw_1+\eta_1(Y-\epsilon e)+eb_1)+\rho_+e \le 0, \quad \rho_+ \ge 0, \tag{40}$$

and

$$\underset{(w_2,\eta_2,b_2,\rho_-)}{Min} \; \frac{c_3}{2}\left(||w_2||^2+\eta_2^2+b_2^2\right)+\frac{1}{2}||Aw_2 +\eta_2(Y-\epsilon e)+eb_2||^2-c_4v_2\rho_-$$

subject to,

$$(Aw_2+\eta_2(Y+\epsilon e)+eb_2)-\rho_-e \ge 0, \quad \rho_- \ge 0. \tag{41}$$

Let us first consider the problem (40). Here we note that $\eta_1 \neq 0$ and therefore, without loss of generality, we can assume that $\eta_1 > 0$. The constraint of (40) can be rewriteen as

$$\left[-A\left(\frac{-w_1}{\eta_1}\right)+(Y-\epsilon e)-e\left(\frac{-b}{\eta_1}\right)\right]+\frac{\rho_+e}{\eta_1} \le 0, \;\; \rho_+ \ge 0$$

On replacing $w_1 := -w_1/\eta_1$, $b_1 := -b_1/\eta_1$ and noting that $\eta_1 \ge 0$, (40) reduces to

$$\underset{(w_1,b_1,\rho_+)}{Min} \; \frac{1}{2}||w_1||^2+b_1^2+||(Y+\epsilon e)-(Aw_1+eb_1)||^2-c_2v_1\rho_+$$

subject to,

$$Aw_1+eb_1+\epsilon e \ge Y+\tfrac{\rho_+e}{\eta_1}, \quad \rho_+ \ge 0. \tag{42}$$

Next, if we replace $eb_1 := eb_1 - \epsilon e$ in (42) then it reduces to

$$\underset{(w_1,b_1,\rho_+)}{Min} \; \frac{1}{2}||w_1||^2+b_1^2+||Y-(Aw_1+eb_1)||^2-c_2v_1\rho_+$$

subject to,

$$Aw_1+eb_1 \ge Y-\left(2\epsilon e-\tfrac{\rho_+e}{\eta_1}\right) \quad \rho_+ \ge 0. \tag{43}$$

Let $\left(2e\epsilon-\tfrac{\rho_+}{\eta_1}\right) := e\epsilon_1$ then it will reduce to

$$\underset{w_1,b_1,\epsilon_1}{Min} \; \tfrac{1}{2}||w_1||^2+b_1^2+\tfrac{1}{2}||(Y-(Aw_1+eb_1)||^2 +c_2v_1\epsilon_1$$

subject to, $$(Aw_1+eb_1)-Y \ge -\epsilon_1e. \tag{44}$$

In the similar manner, assuming $\eta_2 > 0$ and using the replacement $w_2 := -w_2/\eta_2$, $b_2 := -b_2/\eta_2$, problem (41) can be written as

$$\underset{(w_2,b_2,\rho_-)}{Min} \; \frac{1}{2}||w_2||^2+b_2^2+\frac{1}{2}||(Y-\epsilon e)-(Aw_2+eb_2)||^2-c_4v_2\rho_-$$

subject to,,

$$(Aw_2+eb_2)-\epsilon e \le Y-\tfrac{\rho_-}{\eta_2}e, \quad \rho_- \ge 0.$$

If we replace $eb_2 := eb_2+\epsilon e$ and $(2e\epsilon-\frac{\rho_-}{\eta_2}) := e\epsilon_2$ then problems reduces to

$$\underset{w_2,b_2,\epsilon_2}{Min} \; \tfrac{1}{2}||w_2||^2+b_2^2+\tfrac{1}{2}||(Y-(Aw_2+eb_2)||^2 +c_4v_1\epsilon_2$$

subject to, $$Y-(Aw_2+eb_2) \ge -\epsilon_2e. \tag{45}$$

Looking at problems (44) and (45) we observe that our approach is valid provided we can show that $\epsilon_1 = (2\epsilon - \frac{2\rho_+}{\eta_1}) \ge 0$ and $\epsilon_2 = (2\epsilon - \frac{2\rho_-}{\eta_2}) \ge 0$. We can prove this assertion as follow.

As the first hyperplane $w^Tx+\eta_1y+b_1=0$ is the least square fit for the class D^+ so there certainly exists an index j such that

$$\eta_1(y_j+\epsilon)+w_1^Tx_j+b_1 \ge 0. \tag{46}$$

Also from (40),

$$\eta_1(y_i-\epsilon)+w_1^Tx_j+b_1+\rho_+ \le 0, \quad \text{for all } i \tag{47}$$

In particular, taking (47) for j we get

$$\eta_1(y_j-\epsilon)+w_1^Tx_j+b_1+\rho_+ \le 0,$$
$$\text{i.e.} \quad -\eta_1(y_j-\epsilon)-w_1^Tx_j-b_1-\rho_+ \ge 0 \tag{48}$$

Adding (47) and (48) we get $\epsilon_1 = \left(2\epsilon - \frac{\rho_+}{\eta_1}\right) \ge 0.$

Similarly we can prove that $\epsilon_2 = \left(2\epsilon - \frac{\rho_-}{\eta_2}\right) \ge 0.$

Remark 2 The above proof can be appropriately modified to show that ϵ-TSVR formulation of Shao et al. [9] also follows from Bi and Bennett [5] results and TWSVM methodology.

References

1. Burges JC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167
2. Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20(3):273–297
3. Bradley P, Mangasarian OL (2000) Massive data discrimination via linear support vector machines. Optim Methods Softw 13(1):1–10
4. Cherkassky V, Mulier F (2007) Learning from data: concepts, theory and methods. Wiley, New York

5. Bi J, Bennett KP (2003) A geometric approach to support vector regression. Neurocomputing 55(1):79–108
6. Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
7. Peng X (2010) TSVR: an efficient twin support vector machine for regression. Neural Netw 23(3):365–372
8. Khemchandani R, Goyal K, Chandra S (2016) TWSVR: regression via twin support vector machine. Neural Netw 74:14–21
9. Shao YH, Zhang C, Yang Z, Deng N (2013) An ϵ-twin support vector machine for regression. Neural Comput & Applic 23(1):175–185
10. Schölkopf B, Bartlett P, Smola AJ, Williamson RC (1998) Support vector regression with automatic accuracy control. In: ICANN, vol 98. Springer, London, pp 111–116
11. Peng X (2010) A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms. Inf Sci 180(20):3863–3875
12. Schölkopf B, Smola AJ, Williamson RC, Bartlett PL (2000) New support vector algorithms. Neural Comput 12(5):1207–1245
13. Blake CI, Merz CJ (1998) UCI repository for machine learning databases, http://www.ics.uci.edu/*mlearn/MLRepository.html
14. Xu Y, Wang L (2014) K-nearest neighbor-based weighted twin support vector regression. Appl Intell 41(1):299–309
15. Vapnik V (1998) Statistical learning theory, vol 1. Wiley, New York