A novel twin support vector regression

Yitian Xu

College of science, China Agricultural University, Beijing 100083, China

Abstract

Twin support vector regression (TSVR), as an effective regression machine, solves a pair of smaller-sized quadratic programming problems (QPPs) rather than a single large one as in the classical support vector regression (SVR), which makes the learning speed of TSVR approximately 4 times faster than that of the SVR. However, the empirical risk minimization principle is implemented in TSVR, which reduces its generalization ability to a certain extent. In order to improve the prediction accuracy, we propose a novel TSVR for the regression problem by introducing a regularization term into the objective function, which ensures the new algorithm implements the structural risk minimization principle instead of the empirical risk minimization principle. Moreover, the up- and down-bound functions obtained in our algorithm are as parallel as possible. Thus it ensures that our proposed algorithm yields lower prediction error and lower standard deviation in theory. The experimental results on one artificial dataset and six benchmark datasets indicate the feasibility and validity of our novel TSVR.

Key words: SVR, TSVR, novel TSVR, up- and down-bound functions.

1. Introduction

The support vector machine (SVM), introduced by Vapnik [1], is a promising machine learning technique. Compared with other machine learning approaches like artificial neural networks, SVM has many advantages. First, SVM solves a quadratic programming problem, assuring that once an optimal solution is obtained, it is the unique (global) solution. Second, SVM derives a sparse and robust solution by maximizing the margin between two classes. Third, SVM implements the structural risk minimization principle rather than the empirical risk minimization principle, which minimizes the upper bound of the generalization error. The introduction of a kernel function cannot only extend the linear case to the nonlinear case, but can also effectively overcome the “curse of

Email address: xytshuxue@126.com, Tel: +86 1062737077. (Yitian Xu).
Because of its great generalization performance, SVM has been successfully applied in various aspects ranging from machine learning, data mining, and knowledge discovery.

Although SVM generates a more generalized performance than other machine learning methods such as artificial neural networks, the training cost of $O(l^3)$, where $l$ denotes the total size of training data, is expensive. To reduce the computational complexity, Jayadeva et al. proposed a twin support vector machine (TSVM) [2] for the binary classification data in the spirit of the proximal SVM [3–5]. TSVM generates two nonparallel hyper-planes by solving a pair of smaller-sized QPPs such that each hyper-plane is closer to one class and as far from the other as possible. The strategy of solving a pair of smaller-sized QPPs rather than a single large one, makes the learning speed of TSVM approximately four times faster than that of the standard SVM. TSVM has become one of the popular methods because of its low computational complexity. Many variants of TSVM have been proposed in recent years, such as $\nu$-TSVM [6,7], least squares TSVM [8], and TSVR [9,10].

With the introduction of an $\epsilon$-insensitive loss function, the SVM may be extended to support vector regression (SVR). There are two terms in the objective function of SVR, one of which $\frac{1}{2}||w||^2$ embodies the generalization ability of model, the other term measures the training error. So SVR implements the structural risk minimization principle and leads to great generalization performance. Moreover, the two bound functions are parallel. Although there are two terms in the objective function of TSVR, they all measure the training error. So TSVR implements the empirical risk minimization principle, which reduces its generalization ability to a certain degree. Moreover the two bound functions are nonparallel, which easily makes the prediction results fluctuated remarkably, and generates larger standard deviation. Motivated by the above studies, we propose a novel TSVR for the regression problem. Our proposed TSVR also resolves a pair of smaller-sized QPPs to improve the computational speed. While a regularization term $\frac{1}{2}||w_1||^2$ is introduced into the first objective function, then the structural risk minimization principle is implemented [12,13]. In addition, the term $\frac{1}{2}||w_1 - w_2||^2$ is introduced into the second objective function to make two bound functions as parallel as possible, which is significant different from TSVR. Therefore the new TSVR not only yields lower prediction error but also produces lowest standard deviation.

The effectiveness of our proposed algorithm is demonstrated by numerical experiments on one artificial dataset and six benchmark datasets. The experimental results on six benchmark datasets show that our proposed TSVR achieves significant performance in comparison with SVR and TSVR. Moreover, the novel TSVR requires nearly the same CPU time as TSVR, and far lower time than SVR.

The paper is organized as follows. Section 2 outlines the TSVR. A novel TSVR is proposed in Section 3, which includes both the linear and nonlinear cases. Section 4 performs experiments on one artificial dataset and six benchmark datasets to investigate the feasibility and validity of the proposed algorithm. The last section contains the conclusions.
2. Twin support vector regression

To improve the computational speed, Peng [9] proposed an efficient twin support vector regression for the regression problem, termed as TSVR. Given a training set \( T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \), where \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \). For the sake of conciseness, let matrix \( A = (x_1; x_2; \ldots; x_n) \) and matrix \( Y = (y_1; y_2; \ldots; y_n) \). The TSVR generates the \( \epsilon \)-insensitive down-bound function \( f_1(x) = w_1^T x + b_1 \) and \( \epsilon \)-insensitive up-bound function \( f_2(x) = w_2^T x + b_2 \). The TSVR is illustrated in Fig. 1.

![Fig. 1. Illustration of the TSVR](image)

The final regressor is decided by the mean of two nonparallel bound functions, i.e.,

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

(1)

TSVR is obtained by resolving the following pair of QPPs,

\[
 \begin{align*}
 \min_{w_1, b_1, \xi} & \quad \frac{1}{2}||Y - \epsilon e_1 - (Aw_1 + eb_1)||^2 + c_1 \epsilon^T \xi \\
 \text{s.t.} & \quad Y - (Aw_1 + eb_1) \geq \epsilon e_1 - \xi, \\
 & \quad \xi \geq 0 \epsilon,
\end{align*}
\]  

(2)

and

\[
 \begin{align*}
 \min_{w_2, b_2, \eta} & \quad \frac{1}{2}||Y + \epsilon e_2 - (Aw_2 + eb_2)||^2 + c_2 \epsilon^T \eta \\
 \text{s.t.} & \quad (Aw_2 + eb_2) - Y \geq \epsilon e_2 - \eta, \\
 & \quad \eta \geq 0 \epsilon,
\end{align*}
\]  

(3)

where \( c_1, c_2, \epsilon_1 \) and \( \epsilon_2 \) are parameters chosen a priori, \( \xi \) and \( \eta \) are the slack vectors. By introducing the Lagrangian multipliers \( \alpha \) and \( \beta \) into (2) and (3), respectively, we can derive their dual QPPs as follows,
\[
\begin{align*}
\max_{\alpha} & \quad \frac{1}{2} \alpha^T G(G^T G)^{-1} G^T \alpha + g_1^T G(G^T G)^{-1} G^T \alpha - g_1^T \alpha \\
\text{s.t.} & \quad 0 e \leq \alpha \leq c_1 e,
\end{align*}
\]

and

\[
\begin{align*}
\max_{\beta} & \quad \frac{1}{2} \beta^T G(G^T G)^{-1} G^T \beta - g_2^T G(G^T G)^{-1} G^T \beta + g_2^T \beta \\
\text{s.t.} & \quad 0 e \leq \beta \leq c_2 e,
\end{align*}
\]

where \( G = [A \ e], \ g_1 = Y - e\epsilon_1, \) and \( g_2 = Y + e\epsilon_2. \)

Once the dual QPPs (4) and (5) are resolved, we can get \( w_1, b_1 \) and \( w_2, b_2 \) in (1) as follows,

\[
[w_1 \ b_1]^T = (G^T G)^{-1} G^T (g_1 - \alpha),
\]

\[
[w_2 \ b_2]^T = (G^T G)^{-1} G^T (g_2 + \beta).
\]

For the nonlinear case, TSVR resolves the following pair of QPPs,

\[
\begin{align*}
\min_{w_1, b_1, \xi} & \quad \frac{1}{2} \|Y - e\epsilon_1 - (K(A, A^T) w_1 + eb_1)\|^2 + c_1 e^T \xi \\
\text{s.t.} & \quad Y - (K(A, A^T) w_1 + eb_1) \geq e\epsilon_1 - \xi, \\
& \quad \xi \geq 0 e,
\end{align*}
\]

and

\[
\begin{align*}
\min_{w_2, b_2, \eta} & \quad \frac{1}{2} \|Y + e\epsilon_2 - (K(A, A^T) w_2 + eb_2)\|^2 + c_2 e^T \eta \\
\text{s.t.} & \quad K(A, A^T) w_2 + eb_2 - Y \geq e\epsilon_2 - \eta, \\
& \quad \eta \geq 0 e.
\end{align*}
\]

Similarly, we can derive the dual problems of QPPs (8) and (9) as follows,

\[
\begin{align*}
\max_{\alpha} & \quad -\frac{1}{2} \alpha^T H(H^T H)^{-1} H^T \alpha + g_1^T H(H^T H)^{-1} H^T \alpha - g_1^T \alpha \\
\text{s.t.} & \quad 0 e \leq \alpha \leq c_1 e,
\end{align*}
\]

and

\[
\begin{align*}
\max_{\beta} & \quad -\frac{1}{2} \beta^T H(H^T H)^{-1} H^T \beta - g_2^T H(H^T H)^{-1} H^T \beta + g_2^T \beta \\
\text{s.t.} & \quad 0 e \leq \beta \leq c_2 e,
\end{align*}
\]

where \( H = [K(A, A^T) \ e], \ g_1 = Y - e\epsilon_1, \) and \( g_2 = Y + e\epsilon_2. \)

Note that TSVR is comprised of a pair of QPPs such that each QPP determines one of up- or down-bound function by using only a group of constraints instead of two group of constraints in the standard SVR. Hence, TSVR resolves a pair of smaller-sized QPPs rather than a single large one, which implies that TSVR is approximately four times faster than the standard SVR in theory. In addition, TSVR gets two nonparallel hyperplanes, they play a great influence on the end regression function.
3. A novel twin support vector regression

Although there are two terms in the objective function of (8) or (9), they all measure the training error. The first term in the objective function (8) is the sum of squared distances from the shifted function $f_1(x)$ to the training points. Minimizing it leads to the function $f_1(x)$ fit the $\epsilon_1$-insensitive down-bound function. The constraints require the estimated function $f_1(x)$ to be at a distance of at least $\epsilon_1$ from the training points. That is, the output values for the training points should be larger than the function $f_1(x)$ by at least $\epsilon_1$. The slack vector $\xi$ is introduced to measure the error wherever the distance is closer than $\epsilon_1$. The second term of the objective function minimizes the sum of error variables, thus attempting to over-fit the training points. So the empirical risk minimization principle is implemented in TSVR. It is well known that there are two terms in the objective function of the classical SVR, one of which $\frac{1}{2}||w||^2$ embodies the generalization ability of model, the other term measures the training error. So SVR implements the structural risk minimization principle, which makes SVR own greater generalization performance than other learning algorithms in theory. In addition, two bound functions obtained in TSVR are nonparallel, which plays a great influence on the end regression function, which easily makes the prediction results fluctuated remarkably.

For the two drawbacks of TSVR, we propose a novel TSVR in this section, where the structural risk is minimized by introducing a regularization term into the objective function. Different from TSVR, the up- and down-bound functions obtained in our proposed algorithm are as parallel as possible, which easily leads to lower standard deviation.

3.1. Linear case

In the first QPP of novel TSVR, a regularization term $\frac{1}{2}||w_1||^2$ is introduced to enhance the generalization performance. Like SVR, minimizing term $\frac{1}{2}||w_1||^2$ implies that the bound function obtained is as flat as possible, which results in high prediction accuracy. In the second QPP, minimizing term $\frac{1}{2}||w_1 - w_2||^2$ implies that the up- and down-bound functions are as parallel as possible. A novel TSVR is obtained by solving the following pair of QPPs,

$$\min_{w_1,b_1,\xi} \frac{1}{2}||w_1||^2 + \frac{1}{2}(Y - e\epsilon_1 - (Aw_1 + eb_1))^T(Y - e\epsilon_1 - (Aw_1 + eb_1)) + c_1e^T\xi$$

subject to $Y - (Aw_1 + eb_1) \geq e\epsilon_1 - \xi, \xi \geq 0 e$, \hspace{1cm} (12)

and

$$\min_{w_2,b_2,\eta} \frac{1}{2}||w_1 - w_2||^2 + \frac{1}{2}(Y + e\epsilon_2 - (Aw_2 + eb_2))^T(Y + e\epsilon_2 - (Aw_2 + eb_2)) + c_2e^T\eta$$

subject to $(Aw_2 + eb_2) - Y \geq e\epsilon_2 - \eta, \eta \geq 0 e$, \hspace{1cm} (13)

where parameters $c_1, c_2, \epsilon_1, \epsilon_2 > 0$ are chosen a priori, $\xi$ and $\eta$ are slack vectors, and $e$ is the vector of ones of appropriate dimensions.

To resolve the QPP (12), we first introduce the Lagrangian function,
\[ L_1 = \frac{1}{2} ||w_1||^2 + \frac{1}{2} (Y - \epsilon e_1 - (Aw_1 + eb_1))^T (Y - \epsilon e_1 - (Aw_1 + eb_1)) + c_1 e^T \xi \\
- \alpha^T (Y - \epsilon e_1 - (Aw_1 + eb_1) + \xi) - \beta^T \xi, \]  
(14)

where \( \alpha \) and \( \beta \) are Lagrangian multipliers. Differentiating the Lagrangian function \( L_1 \) in (14) with respect to \( w_1, b_1 \) and \( \xi \) yields the following Karush-Kuhn-Tucker (KKT) conditions:

\[
\frac{\partial L_1}{\partial w_1} = w_1 - A^T (Y - \epsilon e_1 - (Aw_1 + eb_1)) + A^T \alpha = 0, 
\]  
(15)

\[
\frac{\partial L_1}{\partial b_1} = -e^T (Y - \epsilon e_1 - (Aw_1 + eb_1)) + e^T \alpha = 0, 
\]  
(16)

\[
\frac{\partial L_1}{\partial \xi} = c_1 e - \alpha - \beta = 0, 
\]  
(17)

\[ Y - (Aw_1 + eb_1) \geq \epsilon e_1 - \xi, \quad \xi \geq 0, \]  
(18)

\[ \alpha^T (Y - \epsilon e_1 - (Aw_1 + eb_1) + \xi) = 0, \]  
(19)

\[ \beta^T \xi = 0, \quad \beta \geq 0 \epsilon. \]  
(20)

Combining Eqs. (15) and (16) leads to

\[
- \begin{bmatrix} A^T \\ e^T \end{bmatrix} \begin{pmatrix} (Y - \epsilon e_1) - [A e] \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \end{pmatrix} + \begin{bmatrix} A^T \\ e^T \end{bmatrix} \alpha + \begin{bmatrix} w_1 \\ 0 \end{bmatrix} = 0. 
\]  
(21)

Define \( G = [A e], \) \( g_1 = Y - \epsilon e_1, \) \( I_2 \) is an identity matrix of \( d \) orders, \( d \) denotes the number of dimensions of samples, and \( u_1 = [w_1^T \ b_1]^T, \) then (21) is simplified as follows,

\[
(G^T G + \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix}) u_1 = G^T (g_1 - \alpha), 
\]  
(22)

we can further get \( u_1 \) from (22),

\[ u_1 = [w_1^T \ b_1]^T = (G^T G + M)^{-1} G^T (g_1 - \alpha). \]  
(23)

where \( M = \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix}. \)

Finally, the dual formulation of the QPP (12) is derived as follows,

\[
\max_{\alpha} \frac{1}{2} \alpha^T G N^{-1} G^T \alpha - g_1^T (G N^{-1} G^T - I_3) \alpha - \frac{1}{2} g_1^T (G N^{-1} G^T - I_3) g_1 
\]  
(24)

s.t. \( 0 \epsilon \leq \alpha \leq c_1 \epsilon, \)

where \( N = G^T G + M, \) \( I_3 \) is an identity matrix of \( n \) orders, \( n \) denotes the number of training samples.
Similarly, we introduce the Lagrangian function of the QPP (13) as follows,

\[
L_2 = \frac{1}{2}||w_1 - w_2||^2 + \frac{1}{2}(Y + \epsilon e_2 - (Aw_2 + eb_2))^T(Y + \epsilon e_2 - (Aw_2 + eb_2)) + c_2 \gamma^T \eta - \gamma^T((Aw_2 + eb_2) - Y - \epsilon e_2 + \eta) - \delta^T \eta,
\]  

where \(\gamma\) and \(\delta\) are Lagrangian multipliers. After differentiating the Lagrangian function \(L_2\) in (25) with respect to \(w_2, b_2\) and \(\eta\), we can obtain the dual formulation of (13) as follows,

\[
\begin{align*}
\max_{\gamma} & \quad -\frac{1}{2} \gamma^T G^{-1} G^T \gamma - g_2^T M N^{-1} G^T \gamma - u_1^T M N^{-1} G^T g_2 \\
& \quad + \frac{1}{2} u_1^T M u_1 + \frac{1}{2} g_2^T g_2 - \frac{1}{2} u_1^T M N^{-1} M u_1 - \frac{1}{2} g_2^T G N^{-1} G^T g_2 \\
\text{s.t.} & \quad 0 \epsilon \leq \gamma \leq c_2 \epsilon,
\end{align*}
\]

where \(G = [A \ e]\), and \(g_2 = Y + \epsilon e_2\).

Once the solution \(\gamma\) to the QPP (26) is found, we can obtain

\[
u_2 = (M + G^T G)^{-1}[G^T(g_2 + \gamma) + Mu_1],
\]

where \(u_2 = [w_2^T \ b_2]^T\). Finally, the estimated regressor is as follows,

\[
f(x) = \frac{1}{2}(f_1(x) + f_2(x)) = \frac{1}{2}(w_1 + w_2)^T x + \frac{1}{2}(b_1 + b_2) = \frac{1}{2}(u_1 + u_2)^T (x - 1)^T.
\]

3.2. Nonlinear case

To extend our proposed algorithm to the nonlinear case, we consider the following kernel-generated functions,

\[
f_1(x) = K(A, x^T)w_1 + b_1, \quad \text{and} \quad f_2(x) = K(A, x^T)w_2 + b_2
\]

As in the linear case, we construct a pair of smaller-sized QPPs,

\[
\begin{align*}
\min_{w_1, b_1, \xi} & \quad \frac{1}{2}||w_1||^2 + \frac{1}{2}(Y - \epsilon e_1 - (K(A, A^T)w_1 + eb_1))^T(Y - \epsilon e_1 - (K(A, A^T)w_1 + eb_1)) + c_1 \epsilon^T \xi \\
\text{s.t.} & \quad Y - (K(A, A^T)w_1 + eb_1) \geq \epsilon e_1 - \xi, \\
& \quad \xi \geq 0 \epsilon,
\end{align*}
\]

and

\[
\begin{align*}
\min_{w_2, b_2, \eta} & \quad \frac{1}{2}||w_1 - w_2||^2 + \frac{1}{2}(Y + \epsilon e_2 - (K(A, A^T)w_2 + eb_2))^T(Y + \epsilon e_2 - (K(A, A^T)w_2 + eb_2)) + c_2 \epsilon^T \eta \\
\text{s.t.} & \quad (K(A, A^T)w_2 + eb_2) - Y \geq \epsilon e_2 - \eta, \\
& \quad \eta \geq 0 \epsilon,
\end{align*}
\]
where parameters $c_1, c_2, \epsilon_1, c_2 > 0$ are chosen a priori, and $\xi, \eta$ are slack vectors. To resolve the QPP (30), we also introduce the following Lagrangian function,

$$L_1 = \frac{1}{2}||w_1||^2 + \frac{1}{2}(Y - \epsilon \epsilon_1 - (K(A, AT)w_1 + \epsilon_1))^{T}(Y - \epsilon \epsilon_1 - (K(A, AT)w_1 + \epsilon_1)) + c_1 \epsilon^T \epsilon - \alpha^T(Y - \epsilon \epsilon_1 - (K(A, AT)w_1 + \epsilon_1) + \xi) - \beta^T \xi.$$  

(32)

After differentiating the Lagrangian function $L_1$ with respect to $w_1$, $b_1$ and $\xi$, we can derive the dual formulation of the QPP (30) as follows,

$$\max_{\alpha} -\frac{1}{2} \alpha^T G^{-1}G^T \alpha - g_1^T (G^{-1}G^T - I_3) \alpha - \frac{1}{2} g_1^T (G^{-1}G^T - I_3) g_1$$

subject to $0 \leq \alpha \leq c_1 \epsilon$,  

(33)

where $G = [K(A, AT) \epsilon]$, $N = G^T G + M$, and $M = \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix}$.

In an exactly similar way, we can derive the dual problem of (31) as follows,

$$\max_{\gamma} -\frac{1}{2} \gamma^T G^{-1}G^T \gamma - g_2^T M^{-1}G^T \gamma - u_1^T M^{-1}G^T \gamma + g_2^T \gamma - u_1^T M^{-1}G^T g_2 + \frac{1}{2} u_1^T M u_1 + u_2^T g_2 - \frac{1}{2} u_1^T M^{-1}M u_1 - \frac{1}{2} g_2^T G^{-1}G^T g_2$$

subject to $0 \leq \gamma \leq c_2 \epsilon$,  

(34)

Once the dual QPPs (33) and (34) are resolved, we can get vectors $u_1$ and $u_2$,

$$u_1 = [w_1^T b_1] = (G^T G + M)^{-1}G^T (g_1 - \alpha),$$

(35)

$$u_2 = [w_2^T b_2] = (M + G^T G)^{-1}G^T (g_2 + \gamma) + Mu_1.$$  

(36)

Finally, we can achieve the end regressor for the nonlinear case as follows,

$$f(x) = \frac{1}{2} (f_1(x) + f_2(x)) = \frac{1}{2} (w_1 + w_2)^T K(A, x^T) + \frac{1}{2} (b_1 + b_2)$$

$$= \frac{1}{2} (u_1 + u_2)^T (K(A, x^T) - I)^T.$$  

(37)

In (33), the third term $-\frac{1}{2} g_1^T (G^{-1}G^T - I_3) g_1$ is a constant corresponding to the variable $\alpha$, then discarding it does not affect the solution of (33). So the computational complexity of (33) is the same as that of (10). In (34), the term $-u_1^T M^{-1}G^T g_2 + \frac{1}{2} u_1^T M u_1 + \frac{1}{2} g_2^T g_2 - \frac{1}{2} u_1^T M^{-1}M u_1 - \frac{1}{2} g_2^T G^{-1}G^T g_2$ is also a constant corresponding to the variable $\gamma$ after $u_1$ is obtained, and it can be ignored while resolving the QPP (34). Therefore the computational complexity of (34) is the same as that of (11). Generally speaking, the computational complexity of our proposed algorithm is the same as that of TSVR in theory, which can be learned from the following experimental results.

4. Numerical Experiments

To demonstrate the validity of our proposed TSVR algorithm, we compare it with SVR and TSVR using seven datasets, including one artificial dataset and six benchmark
datasets. We test the validity of our proposed algorithm according to both its testing error and efficiency. All of the algorithms are implemented in Matlab 7.9 (R2009b).

### 4.1. Evaluation criteria

To evaluate the performance of our proposed algorithm, the evaluation criteria \[10,11\] are specified before presenting the experimental results. The total number of testing samples is denoted by \(m\), while \(y_i\) denotes the real-value of a sample \(x_i\), \(\hat{y}_i\) denotes the predicted value of \(x_i\), and \(\bar{y} = \frac{1}{m} \sum_{i=1}^{m} y_i\) is the mean of \(y_1, y_2, \cdots, y_m\). We use the following five criteria for algorithm evaluation.

- **MAE**: Mean absolute error, defined as \(\text{MAE} = \frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y}_i|\).

- **RMSE**: Root mean squared error, defined as \(\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2}\).

- **SSE/SST**: Ratio between sum of squared error and sum of squared deviation of testing samples, defined as \(\text{SSE/SST} = \frac{\sum_{i=1}^{m} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{m} (y_i - \bar{y})^2}\).

- **SSR/SSE**: Ratio between interpretable sum squared deviation and real sum of squared deviation of testing samples, defined as \(\text{SSR/SSE} = \frac{\sum_{i=1}^{m} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{m} (y_i - \bar{y})^2}\).

- **CPU Time**: The total training and testing time.

In most cases, small SSE/SST means there is good agreement between the estimates and the real values, and decreasing SSE/SST is usually accompanied by an increase in SSR/SST. However, an extremely small value for SSE/SST is in fact not good, for it probably means that the regressor is over-fitting the data.

### 4.2. Parameters selection

The performance of these algorithms depends heavily on the choices of parameters \[14,15\]. In our experiments, we only consider the Gaussian kernel function \(k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\gamma^2}\right)\) for these datasets as it is often employed and yields great generalization performance. We choose optimal values for the parameters by the grid search method. The optimal values for \(c\) in the three algorithms are chosen from the set \(\{2^i | i = -2, -1, \cdots, 8\}\), the Gaussian kernel parameter \(\gamma\) is selected from \(\{2^i | i = -4, -3, \cdots, 9\}\), and the optimal values for \(\epsilon\) are chosen from the set \(\{0.1, 0.2, \cdots, 0.9, 0.95\}\).

### 4.3. Experiment on artificial dataset

We first test the performance of our proposed algorithm on regression of the sinc function, which is defined as

\[
y = \text{Sinc}(x) = \frac{\sin(x)}{x}, \quad x \in [-10, 10].
\]
To demonstrate the performance of our proposed algorithm, 51 training points are perturbed by Gaussian noises with zero mean and variance $0.1^2$. Specially, we have the following training samples:

$$y_i = \frac{\sin x_i}{x_i} + \xi_i, \quad x_i \in [-10, 10], \quad \xi_i \sim N(0, 0.1^2).$$ (39)

A comparison of performance of three algorithms is summarized in Table 1.

| Algorithm       | $(c, \gamma, \nu (\epsilon))$ | MAE   | RMSE  | SSE/SST | SSR/SST | Time 
|-----------------|-------------------------------|-------|-------|---------|---------|-------
| SVR             | (256, 0.25, 0.1)              | 0.0221| 0.0529| 1.5579  | 0.9973  | 4.532 |
| TSVR            | (2, 1, 0.9)                   | 0.0545| 0.0650| 0.0290  | 0.9624  | 0.482 |
| Novel SVR       | (0.125, 0.5, 0.9)             | 0.0524| 0.0559| 0.0395  | 0.8526  | 0.468 |

The corresponding fitting curves obtained by three algorithms are illustrated in Fig. 2.

From Fig. 2 we can find that the fitting curves obtained by novel TSVR and TSVR are smoother than that obtained by SVR, which implies that SVR easily result in over-fitting problem, and reduces the generalization ability of SVR.
4.4. Experiments on benchmark datasets

The validity of our proposed algorithm is also demonstrated using six benchmark datasets obtained from the UCI machine learning repository\(^1\). In this study, we use Diabets, Con.s, Machine cpu, Triazines, Auto-price, and Auto-Mpg. Their backgrounds can be learned from above Web site. For instance, Diabets data set concerns the study of the factors affecting patterns of insulin-dependent diabetes mellitus in children. The objective is to investigate the dependence of the level of serum C-peptide on the various other factors in order to understand the patterns of residual insulin secretion. The response measurement is the logarithm of C-peptide concentration at the diagnosis, and the predictor measurements age and base deficit, a measure of acidity.

For the experiment on each dataset, we use 5-fold cross-validation to evaluate the performance of our novel TSVR, SVR, and TSVR. That is to say, the dataset is split randomly into five subsets, and one of those sets is reserved as a test set; this process is repeated five times.

The comparisons of experimental results are summarized in Table 2, where the error term denotes the mean value of five times testing results and plus or minus the standard deviation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Regressor</th>
<th>((c_1, c_2, \gamma, \nu(\epsilon_1), \epsilon_2))</th>
<th>MAE</th>
<th>RMSE</th>
<th>SSE/SST</th>
<th>SSR/SST</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diabets</td>
<td>SVR</td>
<td>(256, ~, 8, 0.1, ~)</td>
<td>0.458 ± 0.108</td>
<td>0.560 ± 0.081</td>
<td>0.800 ± 0.215</td>
<td>0.515 ± 0.241</td>
<td>0.142</td>
</tr>
<tr>
<td></td>
<td>TSVR</td>
<td>(4, ~, 125, 0.95, ~)</td>
<td>0.466 ± 0.093</td>
<td>0.553 ± 0.072</td>
<td>0.792 ± 0.277</td>
<td>0.650 ± 0.215</td>
<td>0.061</td>
</tr>
<tr>
<td></td>
<td>Novel TSVR</td>
<td>(64, 256, 0, 125, 0.1, 0.95)</td>
<td>0.480 ± 0.077</td>
<td>0.612 ±0.061</td>
<td>0.977 ± 0.332</td>
<td>0.516 ± 0.241</td>
<td>0.056</td>
</tr>
<tr>
<td>Con.s</td>
<td>SVR</td>
<td>(256, ~, 1.0, 1, ~)</td>
<td>0.216 ± 0.032</td>
<td>0.253 ± 0.031</td>
<td>0.781 ± 0.232</td>
<td>0.257 ± 0.100</td>
<td>1.218</td>
</tr>
<tr>
<td></td>
<td>TSVR</td>
<td>(0.125, ~, 64, 0.95, ~)</td>
<td>0.182 ± 0.027</td>
<td>0.221 ± 0.022</td>
<td>0.620 ± 0.210</td>
<td>0.500 ± 0.258</td>
<td>0.153</td>
</tr>
<tr>
<td></td>
<td>Novel TSVR</td>
<td>(128, 1, 125, 0.8, 0.95)</td>
<td>0.111 ± 0.010</td>
<td>0.147 ± 0.003</td>
<td>0.944 ± 0.188</td>
<td>0.546 ± 0.187</td>
<td>0.439</td>
</tr>
<tr>
<td>Machine cpu</td>
<td>SVR</td>
<td>(64, ~, 0.8, 32, ~)</td>
<td>78.45 ± 32.16</td>
<td>152.68 ± 74.42</td>
<td>1.133 ± 0.083</td>
<td>0.133 ± 0.083</td>
<td>19.34</td>
</tr>
<tr>
<td></td>
<td>TSVR</td>
<td>(16, ~, 8, 0.95, ~)</td>
<td>93.93 ± 17.47</td>
<td>146.78 ± 59.05</td>
<td>1.411 ± 0.922</td>
<td>0.488 ± 0.875</td>
<td>0.501</td>
</tr>
<tr>
<td></td>
<td>Novel TSVR</td>
<td>(128, 1, 125, 0.8, 0.95)</td>
<td>42.85 ±14.48</td>
<td>71.29 ±36.74</td>
<td>0.383 ± 0.365</td>
<td>1.017 ± 0.384</td>
<td>0.481</td>
</tr>
<tr>
<td>Triazines</td>
<td>SVR</td>
<td>(64, ~, 0.2, 0.25, ~)</td>
<td>0.112 ± 0.015</td>
<td>0.153 ± 0.013</td>
<td>0.998 ± 0.087</td>
<td>0.171 ± 0.126</td>
<td>8.345</td>
</tr>
<tr>
<td></td>
<td>TSVR</td>
<td>(0.125, ~, 256, 0.1, ~)</td>
<td>0.131 ± 0.008</td>
<td>0.144 ± 0.011</td>
<td>0.885 ± 0.064</td>
<td>0.193 ± 0.076</td>
<td>0.469</td>
</tr>
<tr>
<td></td>
<td>Novel TSVR</td>
<td>(64, 1, 256, 0.5, 0.125, 0.8, 0.09)</td>
<td>0.109 ± 0.005</td>
<td>0.148 ± 0.003</td>
<td>0.953 ± 0.157</td>
<td>0.444 ± 0.139</td>
<td>0.423</td>
</tr>
<tr>
<td>Auto-price</td>
<td>SVR</td>
<td>(256, ~, 0.6, 32, ~)</td>
<td>5103.0 ± 2147.9</td>
<td>6713.3 ±2534.6</td>
<td>1.679 ± 0.895</td>
<td>0.679 ± 0.895</td>
<td>5.971</td>
</tr>
<tr>
<td></td>
<td>TSVR</td>
<td>(0.125, ~, 256, 0.95, ~)</td>
<td>4333.0 ± 2098.0</td>
<td>6444.4 ±2892.1</td>
<td>2.281 ± 2.537</td>
<td>2.214 ± 2.237</td>
<td>0.288</td>
</tr>
<tr>
<td></td>
<td>Novel TSVR</td>
<td>(256, 1, 125, 0.95, 0.1)</td>
<td>2501.4 ±539.5</td>
<td>3567.6 ±997.4</td>
<td>0.481 ± 0.165</td>
<td>1.022 ± 0.827</td>
<td>0.295</td>
</tr>
<tr>
<td>Auto-Mpg</td>
<td>SVR</td>
<td>(0.5, ~, 1, 1, ~)</td>
<td>6.880 ± 0.705</td>
<td>8.081 ± 0.715</td>
<td>114.84 ±36.05</td>
<td>0.484 ± 0.531</td>
<td>596.6</td>
</tr>
<tr>
<td></td>
<td>TSVR</td>
<td>(2, ~, 512, 0.95, ~)</td>
<td>2.501 ± 0.480</td>
<td>3.635 ± 0.902</td>
<td>0.280 ± 0.072</td>
<td>0.943 ± 0.250</td>
<td>1.910</td>
</tr>
<tr>
<td></td>
<td>Novel TSVR</td>
<td>(0.125, 128, 0.125, 0.95, 0.09)</td>
<td>2.692 ± 0.462</td>
<td>3.445 ± 0.645</td>
<td>0.261 ± 0.082</td>
<td>0.936 ± 0.386</td>
<td>1.691</td>
</tr>
</tbody>
</table>

\(^1\) http://archive.ics.uci.edu/ml/datasets.html.
From Table 2, we know that our proposed TSVR outperform two other algorithms for most datasets. To further evaluate the performance of three algorithms, we count their average ranks on MAE and RMSE values, respectively. The average ranks of three algorithms on MAE values are shown in Table 3, and average ranks on RMSE values are shown in Table 4. From Table 3 we can find the average rank of our proposed TSVR is lower than that of SVR and TSVR, and the average ranks of two other algorithms are nearly the same. From Table 4 we can also find the average rank of our proposed TSVR is lower than that of SVR and TSVR. Certainly, the advantage of our proposed algorithm is not evident as that in Table 3.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVR</th>
<th>TSVR</th>
<th>Novel TSVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diabetes</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Con.s</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Machine cpu</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Triazines</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Auto-price</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Auto-mpg</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Average rank</strong></td>
<td><strong>2.33</strong></td>
<td><strong>2.16</strong></td>
<td><strong>1.5</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVR</th>
<th>TSVR</th>
<th>Novel TSVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diabetes</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Con.s</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Machine cpu</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Triazines</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Auto-price</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Auto-mpg</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>Average rank</strong></td>
<td><strong>2.83</strong></td>
<td><strong>1.66</strong></td>
<td><strong>1.5</strong></td>
</tr>
</tbody>
</table>

4.5. Result discussion

After implementing these experiments on six benchmark datasets, we can draw the following conclusions:

- From the perspective of prediction accuracy, our proposed algorithm yields the best generalization performance among three algorithms for most cases. Although our proposed algorithm yields lightly higher errors than TSVR on Diabetes and Auto-Mpg datasets, their standard deviations obtained by our novel TSVR are lower than those obtained by TSVR, which implies that our proposed algorithm is stable.
In terms of computation time, the novel TSVR requires nearly the same CPU time as that of TSVR, and they are far less than that of SVR. That is to say, the new introduced terms in our proposed algorithm do not increase the computational complexity.

In the case of standard deviation, our proposed algorithm produces lowest standard deviation on MAE and RMSE for all datasets. It implies the second modification that \( \frac{1}{2}||w_1 - w_2||^2 \) is introduced into the second objective function is effective.

5. Conclusion

In this paper, we propose a novel TSVR for the regression problem, where a regularization term \( \frac{1}{2}||w_1||^2 \) is introduced into the first objective function to enhance the generalization performance of the model, and the term \( \frac{1}{2}||w_1 - w_2||^2 \) is introduced into the second objective function to make the two bound functions as parallel as possible. Therefore the structural risk minimization principle is implemented rather than the empirical risk minimization principle in our proposed TSVR. Moreover, the two bound functions obtained by our proposed algorithm are as parallel as possible, thus it can further improve the generalization performance. The experimental results shows that our proposed algorithm not only yields lowest prediction error for most datasets, but also produces lowest testing standard deviation for all datasets. While it costs nearly the same CPU time as TSVR does. So the new introduced regularization terms does not increase the computational complexity. Therefore our proposed algorithm is both effective and feasible.

Acknowledgments The author gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation. This work was supported by National Natural Science Foundation of China (No. 61153003).

References