IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 25, No. 4, August, 2020, pp. 427-435
Journal homepage: http://iieta.org/journals/isi

Improving the Speed of Support Vector Regression Using Regularized Least Square Regression

Sanaz Sagha Pirmard[1], Yahya Forghani[2*]

[1] Computer Department, Imam Reza International University, Mashhad 9138833186, Iran
[2] Computer Department, Islamic Azad University, Mashhad Branch, Mashhad 9187147578, Iran

Corresponding Author Email: yforghani@mshdiau.ac.ir

https://doi.org/10.18280/isi.250404

Received: 20 May 2020
Accepted: 26 July 2020

Keywords:
ε-insensitive support vector regression (ε-SVR), regularized least square (RLS), runtime, function estimation

ABSTRACT

The Regularized Least Square (RLS) method is one of the fastest function estimation methods, but it is too sensitive to noise. Against this, ε-insensitive Support Vector Regression (ε-SVR) is robust to noise but doesn't have a good runtime. ε-SVR supposes that the noise level is at most ε. Therefore, the center of a tube with radius ε, which is used as the estimated function, is determined in a way that the training data are located in that tube. Therefore, this method is robust to such noisy data. In this paper, to improve the runtime of ε-SVR, first, an initial estimated function is obtained using the RLS method. Then, unlike the ε-SVR model, which uses all the data to determine the lower and upper limits of the tube, our proposed method uses the initial estimated function for determining the tube and the final estimated function. Strictly speaking, the data below and above the initial estimated function are used to estimate the upper and lower limits of the tube, respectively. Thus, the number of the model constraints and, consequently, the model runtime are reduced. The experiments carried out on 15 benchmark data sets confirm that our proposed method is faster than ε-SVR, ε-TSVR and pair v-SVR, and its accuracy are comparable with that of ε-SVR, ε-TSVR and pair v-SVR.

1. INTRODUCTION

One of the most commonly used eager learning method is least-square method (LS) [1], which first was proposed by Carl Friedesh Gauss. In the LS method, the best estimated function is a function that the sum of square of the difference between the observed values and the corresponding estimated values with the estimated function is minimum. Later, on the basis of the LS method, the Regulation Least Square (RLS) method was proposed [2, 3]. The RLS uses a regularization term to reduce the complexity of estimated function and to increase its generalization ability.

ε-insensitive Support Vector Regression (ε-SVR) [4] supposes that the noise in the output variable is at most ε. Therefore, the center of a tube with radius ε, which is used as the estimated function, is determined in a way that the training data are located in that tube. Therefore, this method is robust to such noisy data. In v-SVR [5], the maximum noise level or the tube radius is considered as a model variable, which is determined in the process of solving the model. In both ε-SVR and v-SVR, the maximum noise level or the radius of the tube is considered to be constant or independent in terms of the training data characteristics. To alleviate this shortcoming, Support Vector Interval Regression Networks (SVIRNs) was proposed [6]. In SVIRNs, the upper and lower bounds of the tube are automatically determined by two independent RBF networks. In this model, the radius of the tube can change by changing the features of the training data. Therefore, SVIRNs is robust to a noise whose maximum level depends on training data features. These two networks are initialized using ε-SVR. v-Support Vector Interval Regression Networks (v-SVIRN) [7] determines the upper and lower bounds of the tube by only one neural network or one model. In Twin SVR (TSVR) [8], estimated function is determined by solving two small Quadratic Programming Problems (QPPs), while ε-SVR solves one large QPP. Solving the TSVR model needs less time than solving ε-SVR model. Later, in ε-TSVR [9], a regularization term was used to reduce the complexity of estimated function and to increase its generalization ability. The regularization term of ε-TSVR is L2-norm of the estimated function weight vector, while in L1-TWSVR model [10], L1-norm of the estimate function weight vector was used as the regularization term. L1-TWSVR is a linear programming, while ε-TSVR is a QPP. As well, the estimated function of L1-TWSVR is sparser than that of ε-TSVR. Therefore, the testing time of L1-TWSVR model is less than that of ε-TSVR. Pair v-SVR is extended version of ε-TSVR which has additional advantage of using parameter v for controlling the bounds on fraction of support vectors and error [11].

All the mentioned SVR-based methods assume that the noise distribution is symmetrical, therefore, the center of a tube which contain noisy data is used as the estimated function. In other words, the estimated function is determined in such a way that approximately half of the training data are positioned above it, and the remaining data are positioned below it. In the asymmetric v-SVR method [12], the estimated function is determined in such a way that an arbitrary number of training data are positioned above it and the remaining data are positioned below it. The size of each of these two parts of data is determined based on the prior knowledge about the noise distribution. Next, twin version of asymmetric v-SVR, i.e.

asymmetric v-TSVR [13], was proposed to improve the runtime of the asymmetric v-SVR model.

Among the mentioned methods, the LS and RLS methods have the lowest runtime because each of them is an n + 1 variable problem with a close-form solution which is obtained by inverting an n × n matrix, but they are too sensitive to noisy data. Against this, SVR-based models have a good robustness against noise but do not have a good runtime because for example to solve ε-SVR problem, a quadratic programming problem with 2×n variables and linear constraints must be solved which is too time consuming. In this paper, to improve the runtime of ε-SVR, first, an initial estimated function is obtained using the RLS method. Then, unlike the ε-SVR model, which uses all training data to determine the lower and upper limits of the tube, our proposed method uses the initial estimated function for determining the tube and the final estimated function. In other words, the data below and above the initial estimated function are used to estimate the upper limit and the lower limit of the tube, respectively. Thus, the number of the model constraints and, consequently, the model runtime are reduced. The experiments carried out on 15 benchmark data sets confirm that our proposed method is faster than ε-SVR, ε-TSVR and pair v-SVR, and its accuracy are comparable with that of ε-SVR, ε-TSVR and pair v-SVR.

In continue, in Section 2, RLS and ε-SVR are introduced. Then, in Section 3, our proposed method is presented. In Section 4, the results of the experiments are presented and Section 5 provides conclusion.

2. BACKGROUND

2.1 RLS

Let $\{(x_1,y_1), (x_2,y_2), \dots, (x_n,y_n)\}$, be training data where $x \in R^d$ is input variable and $y \in R$ is output variable. Suppose $\varphi(.)$ is a function which transforms data from the input space into a high-dimensional feature space. The goal is to find the function $f(x)=w^T\boldsymbol{\varphi}(x)+b$ with the weight vector $w = \sum_{i=1}^{n} \alpha_i \varphi(x_i)$ and the bias *b* such that

$$\forall i: y_i \cong f(x_i). \tag{1}$$

For this purpose, the following model is solved:

$$\min_{\alpha} \frac{1}{2}(\|\alpha\|^2 + b^2) + \frac{C_1}{2}\sum_{i=1}^{n}(f(x_i) - y_i)^2, \tag{2}$$

where, $\alpha=(\alpha_1, \alpha_2, \dots, \alpha_n)^T$. The first term of model (2) is intended to prevent over-fitting and to increase the generalizability. The parameter C_1 controls the importance of the estimated function training error against generalization ability. Model (2) can be written as follows:

$$\min_{\alpha} \frac{1}{2}(\|\alpha\|^2 + b^2) + \frac{C_1}{2}\|K\alpha + b - y\|^2, \tag{3}$$

where, $y=(y_1, y_2, \dots, y_n)^T$ and K is a matrix of which the i -th row of j-th column, i.e. K_{ij}, is equal to $k(x_j,x_i)= \boldsymbol{\varphi}^T(x_j)\boldsymbol{\varphi}(x_i)$ where k(…) is a kernel function. The model (3) can be written as follows:

$$\min_{\tilde{\alpha}} L = \frac{1}{2}\|\tilde{\alpha}\|^2 + \frac{C_1}{2}\left\|\tilde{K}\tilde{\alpha} - y\right\|^2, \tag{4}$$

where,

$$\tilde{\alpha} = \begin{pmatrix}\alpha \\ b\end{pmatrix}, \tag{5}$$

$$\tilde{K} = (K \quad 1). \tag{6}$$

At the optimal point of the model (4), we have:

$$0 = \frac{\partial L}{\partial \tilde{\alpha}} = \tilde{\alpha} + C_1\tilde{K}^T\tilde{K}\tilde{\alpha} - C_1\tilde{K}^T y. \tag{7}$$

Therefore,

$$\tilde{\alpha} = C_1\left(I + C_1\tilde{K}^T\tilde{K}\right)^{-1}\tilde{K}^T y. \tag{8}$$

RLS is sensitive to noise because when y_i is noisy, the optimal function f(.) changes mistakenly in order to the objective function of RLS become minimum according to Eq. (2).

2.2 ε-SVR method

ε-SVR supposes that the noise level is at most ε. Therefore, the center of a tube with radius ε, which is used as the estimated function, is determined in a way that the training data are located in that tube. In other words, the function $f(x)=w^T\boldsymbol{\varphi}(x)+b$ is determined in such a way that as much as possible,

$$\forall i: -\varepsilon \le y_i - f(x_i) \le \varepsilon. \tag{9}$$

To do this, the following model is solved:

$$\begin{aligned} &\min_{w,b,\xi,\hat{\xi}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \hat{\xi}_i) \\ &\text{subject to} \begin{cases} y_i - f(x_i) \le \varepsilon + \xi_i\,, & i = 1,2,\dots,n; \\ f(x_i) - y_i \le \varepsilon + \hat{\xi}_i\,, & i = 1,2,\dots,n; \\ \xi_i, \hat{\xi}_i \ge 0\,, & i = 1,2,\dots,n; \end{cases} \end{aligned} \tag{10}$$

where, ξ and $\hat{\xi}$ allow outliers are not positioned outside the tube (see Figure 1). The number of outliers positioned outside the tube is controlled by the parameter C≥0.

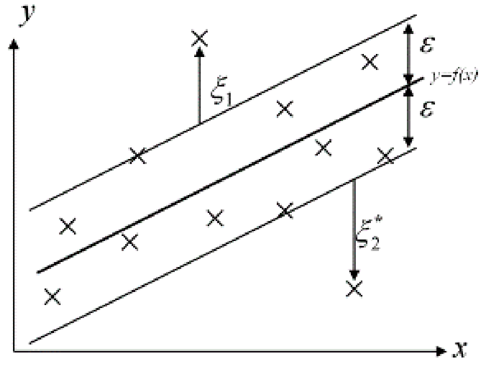


Figure 1. The estimated function of ε-SVR for the training data represented by x signs

The dual of the model (10) is as follows:

$$\max_{\alpha,\hat{\alpha}} \sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)y_i - \varepsilon\sum_{i=1}^{n}(\hat{\alpha}_i + \alpha_i)$$
$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j)k(x_i, x_j) \qquad (11)$$
$$subject\ to \begin{cases}\sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i) = 0\,; \\ 0 \le \alpha_i \le C,\ \ i = 1, \dots, n; \\ 0 \le \hat{\alpha}_i \le C,\ \ i = 1, \dots, n;\end{cases}$$

According to the Karush-Kuhn-Tucker (KKT) conditions [14], we have:

$$w = \sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)\varphi(x_i), \qquad (12)$$

Therefore, the estimated function is determined according to the following formula:

$$f(x) = w^T\varphi(x) + b = \sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)k(x_i, x) + b, \qquad (13)$$

where, based on the KKT conditions, the bias is determined by the following formula:

$$b = \begin{cases} y_i - \sum_{j=1}^{n}(\alpha_j - \hat{\alpha}_j)k(x_j, x_i) - \varepsilon, & 0 < \alpha_i < C. \\ y_i - \sum_{j=1}^{n}(\alpha_j - \hat{\alpha}_j)k(x_j, x_i) + \varepsilon, & 0 < \hat{\alpha}_i < C. \end{cases} \qquad (14)$$

3. OUR PROPOSED METHOD

The RLS have the good runtime, but it is sensitive to noisy data. Against this, ε-SVR model-based models are robust to noise but do not have a good runtime. In this paper, to improve the runtime of ε-SVR, first, an initial estimated function is obtained using the RLS method. Then, unlike the ε-SVR model, which uses all training data to determine the lower and upper limits of the tube, our proposed method uses the initial estimated function for determining the tube and the final estimated function. In other words, the data below and above the initial estimated function are used to estimate the upper and the lower limits of the tube, respectively. Thus, the number of the model constraints and, consequently, the model runtime are reduced.

Let $f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) + b$ be initial estimated function obtained by using the RLS model based on the training data $\{(x_1,y_1), (x_2,y_2), \dots, (x_n,y_n)\}$. If $f(x_i\ ) \le 0$ then (x_i,y_i) is a point below the function f(.); otherwise, (x_i,y_i) is a point above the function f(.). Let I_a be the index of the training data above the function f, and I_b be the index of the training data below the function f, and $|I_a|+|I_b|=n$. Our proposed model for determining a tube with the radius ε and the center $\tilde{f}(x) = w^T\varphi(x) + b$ is as follows:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C_2\sum_{i=1}^{n}\xi_i$$
$$\text{subject to} \begin{cases} y_i - \tilde{f}(x_i) \le \varepsilon + \xi_i\,,\ \ i \in I_a; \\ \tilde{f}(x_i) - y_i \le \varepsilon + \xi_i\,,\ \ i \in I_b; \\ \xi_i \ge 0\,,\ \ i = 1,2, \dots, n. \end{cases} \qquad (15)$$

where, ξ allows outliers are not positioned outside the tube. The number of outliers positioned outside the tube is controlled by the parameter C_2. Unlike ε-SVR which uses all training data to determine the lower and upper bounds of the tube, in our proposed model, only the upper half of the training data is used to determine the upper limit of the tube, and also only the lower half of the training data is used to determine the lower limit of the tube. The upper half and lower half of training data are determined based on the initial estimated function obtained by the RLS. The Lagrange function of the model (15) is as follows:

$$L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$-\sum_{i\in I_a}\alpha_i(\varepsilon + \xi_i - y_i + w^T\varphi(x_i) + b) \qquad (16)$$
$$-\sum_{i\in I_b}\alpha_i(\varepsilon + \xi_i + y_i - w^T\varphi(x_i) - b) - \sum_{i=1}^{n}\gamma_i\,\xi_i;$$

where, γ_i and α_i are Lagrange coefficients. At the optimal point of the model (15) we have:

$$\frac{\partial L}{\partial w} = 0 \to w = \sum_{i\in I_a}\alpha_i\varphi(x_i) - \sum_{i\in I_b}\alpha_i\varphi(x_i)\,; \qquad (17)$$

$$\frac{\partial L}{\partial b} = 0 \to \sum_{i\in I_a}\alpha_i - \sum_{i\in I_b}\alpha_i = 0; \qquad (18)$$

$$\frac{\partial L}{\partial \xi} = 0 \to \alpha_i = C_2 - \gamma_i\,, \qquad i = 1,2, \dots, n; \qquad (19)$$

$$y_i - w^T\varphi(x_i) - b \le \varepsilon + \xi_i\,, \qquad i \in I_a; \qquad (20)$$

$$w^T\varphi(x_i) + b - y_i \le \varepsilon + \xi_i\,, \qquad i \in I_b; \qquad (21)$$

$$\alpha_i(\varepsilon + \xi_i - y_i + w^T\varphi(x_i) + b) = 0, \qquad i \in I_a; \qquad (22)$$

$$\alpha_i(\varepsilon + \xi_i + y_i - w^T\varphi(x_i) - b) = 0, \qquad i \in I_b; \qquad (23)$$

$$\gamma_i\xi_i = 0, \qquad i = 1,2, \dots, n; \qquad (24)$$

$$\xi_i, \alpha_i, \gamma_i \ge 0, \qquad i = 1,2, \dots, n. \qquad (25)$$

By substituting Eq. (17) to Eq. (24) into the Lagrange function, this function is simplified as follows:

$$L = \sum_{i\in I_o}\alpha_i y_i - \sum_{i\in I_u}\alpha_i y_i - \varepsilon\sum_{i=1}^{n}\alpha_i$$
$$-\frac{1}{2}\left(\sum_{i\in I_o}\sum_{j\in I_o}\alpha_i\,\alpha_j k(x_i, x_j) + \sum_{i\in I_u}\sum_{j\in I_u}\alpha_i\,\alpha_j k(x_i, x_j)\right. \qquad (26)$$
$$\left. - 2\sum_{i\in I_o}\sum_{j\in I_u}\alpha_i\,\alpha_j k(x_i, x_j)\right)$$

In addition, according to Eq. (19), and since $\alpha_i,\gamma_i \geq 0$, we have:

$$0 \leq \alpha_i \leq C_2, \quad i = 1,2,\dots,n; \tag{27}$$

Therefore, the dual of the model (15) is as follows:

$$\begin{aligned} \max_{\alpha} & \sum_{i\in I_a} \alpha_i y_i - \sum_{i\in I_b} \alpha_i y_i - \varepsilon \sum_{i=1}^{n} \alpha_i \\ & -\frac{1}{2}\Bigg(\sum_{i\in I_a}\sum_{j\in I_a} \alpha_i\,\alpha_j k(x_i,x_j) + \sum_{i\in I_b}\sum_{j\in I_b} \alpha_i\,\alpha_j k(x_i,x_j) \\ & \qquad -2\sum_{i\in I_a}\sum_{j\in I_b} \alpha_i\,\alpha_j k(x_i,x_j)\Bigg) \\ \text{subject to} & \begin{cases} \sum_{i\in I_a} \alpha_i - \sum_{i\in I_b} \alpha_i = 0; \\ 0 \leq a_i \leq C_2, \ i = 1,2,\dots,n; \end{cases} \end{aligned} \tag{28}$$

If we define:

$$\beta_i = \begin{cases} \alpha_i, & i \in I_o; \\ -\alpha_i, & i \in I_u, \end{cases} \tag{29}$$

then, the model (28) can be written as follows:

$$\begin{aligned} \max_{\beta} & \sum_{i=1}^{n} \beta_i y_i - \varepsilon\left(\sum_{i\in I_a} \beta_i - \sum_{i\in I_b} \beta_i\right) \\ & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_i\,\beta_j k(x_i,x_j) \\ \text{subject to} & \begin{cases} \sum_{i=1}^{n} \beta_i = 0; \\ 0 \leq \beta_i \leq C_2, \ i \in I_a; \\ -C_2 \leq \beta_i \leq 0, \ i \in I_b; \end{cases} \end{aligned} \tag{30}$$

which is a convex quadratic programming problem. After solving the model (30) and obtaining the optimal value of β, the optimal values of α can be determined using Eq. (29). Given Eq. (22), for each $i \in I_a$, if $\alpha_i>0$ then

$$y_i - w^T\varphi(x_i) - b = \varepsilon + \xi_i; \tag{31}$$

and according to Eq. (19), if $\alpha_i<C_2$ then

$$\gamma_i > 0. \tag{32}$$

Then, according to Eq. (24), we have:

$$\xi_i = 0. \tag{33}$$

Thus, for each $i \in I_a$, if $0<\alpha_i<C_2$, the bias can be obtained from the following equation:

$$b = y_i - w^T\varphi(x_i) - \varepsilon. \tag{34}$$

According to Eq. (23), for each $i\in I_b$, if $\alpha_i>0$, then

$$-y_i + w^T\varphi(x_i) + b = \varepsilon + \xi_i; \tag{35}$$

and according to Eq. (19), if $\alpha_i<C_2$, then

$$\gamma_i > 0; \tag{36}$$

Then, according to Eq. (24), we have:

$$\xi_i = 0. \tag{37}$$

Thus, for each $i \in I_u$, if $0<\alpha_i<C_2$, the bias can be obtained from the following equation:

$$b = \varepsilon + y_i - w^T\varphi(x_i). \tag{38}$$

According to Eq. (17), the optimal hyperplane or estimated function is as follows:

$$\begin{aligned} f(x) &= w^T\varphi(x) + b \\ &= b + \sum_{i\in I_a} \alpha_i k(x,x_i) - \sum_{i\in I_b} \alpha_i k(x,x_i) \\ &= b + \sum_{i=1}^{n} \beta_i k(x,x_i) = b + \sum_{i\in SV} \beta_i k(x,x_i)\,; \end{aligned} \tag{39}$$

where, $SV = \{i|\beta_i \neq 0\}$ is called support vector set.

4. EXPERIMENTS

In this section, our proposed method is compared with ε-SVR [4], ε-TSVR [9] and pair v-SVR [11] using 15 benchmark data sets of the UCI repository. The characteristics of these data sets are in accordance with Table 1. The kernel function used in each regression method is Gaussian kernel function. For each data set, the optimal values of the parameters C_1,C_2,C_3,C_4 were selected from the set {0,0.01,…,100}, the optimal value of the parameter $\varepsilon_1=\varepsilon_2=\varepsilon$ was selected from the set {0,0.1}, and the optimal value of the parameter σ was selected from the set {0.1,0.2,…,100} by using the grid search mechanism. These optimal values were reported in Table 2. The RMSE and runtime of each method for the best values of their parameters are according to Table 3 and Table 4, respectively. To calculate RMSE, 10-fold cross validation was used. As it can be seen, the RMSE of our proposed method is competitive with the three other methods and the run time of our proposed method is much less than that of ε-SVR and less than that of ε-TSVR and pair v-SVR. The reason is that in our proposed method instead of solving the constrained quadratic programming problem (11) with 2×n variables, two smaller problems, i.e. the unconstrained quadratic programming problem (2) with n + 1 variables, and then the constrained quadratic programming problem (30) with n variables are solved. Also, in each of ε-TSVR and pair v-SVR, two constrained quadratic programming problem problems with n variables are solved. Solving the mentioned unconstrained quadratic problem is faster than the mentioned constrained quadratic problems of the same size. It should be noted that, each problem was solved using MATLAB2015 on a computer of 8GbRAM and 2.20GHz CPU. The quadprog function and the interior-point-convex algorithm were used to solve each constrained quadratic model.

Table 1. Characteristics of 15 UCI datasets

No.	Dataset	Application	#Instance	#Feature
1	Pyrimidine	Regression	74	27
2	Triazines	Regression	186	61
3	Bodyfat	Regression	252	14
4	Haberman	Classification	306	4
5	Yacht Hydrodynamics	Regression	308	7
6	ionosphere	Classification	351	34
7	Housing	Regression	506	14
8	Pima Indians Diabetes	Classification	768	8
9	Concrete Compressive Strength	Regression	1030	9
10	Mg	Regression	1385	7
11	banknote authentication	Classification	1372	5
12	Abalone	Regression	4177	8
13	Wine	Regression	4898	12
14	Wisconsin Breast Cancer	Classification	569	32
15	Forest Fires	Regression	517	13

Note that determination of appropriate values for the parameters of a model is a challenging issue which can be estimated using a validation set or based on prior knowledge which may not be available for each dataset. The Figures 2, 3 and 4 show the sensitivity of RMSE of our proposed method to the parameters C_1, C_2, and σ, respectively. According to Figure 2, for each dataset except "Concrete" dataset, the best RMSE can be obtained by a large value of C_1. Moreover, the sensitivity of RMSE of our proposed method to the parameter C_1 for "Concrete" dataset is small. According to Figure 3, the best RMSE can be obtained by a large value of C_2 for some datasets and the best RMSE can be obtained by a small value of C_2 for the others. But, as it can be seen, the RMSE is not too sensitive to a wide range of values of C_2. For example, the sensitivity of MSE of our proposed method to the parameter C_2 for "Triazines" dataset is about 0.004 for the range [0,100]. According to Figure 4, the RMSE is also not too sensitive to a wide range of the values of σ. For example, the sensitivity of MSE of our proposed method to the parameter σ for "Triazines" dataset is about 0.16 for its whole range, and about 0.01 for the range [20,100].

Table 2. The best parameters of our proposed method with the lowest RMSE

Dataset	σ	C_1	C_2
Pyrimidine	63.4	100	100
Triazines	28.3	100	100
Bodyfat	100	100	20
Haberman	100	100	100
Yacht Hydrodynamics	2	0.01	100
ionosphere	1.4	50	1
Housing	50	0.01	100
Pima Indians Diabetes	49.4	100	1
Concrete Compressive Strength	38	0.01	100
Mg	2	0.01	100
Banknote authentication	1.2	100	10
Abalone	100	100	100
Wine	1	100	100
Wisconsin Breast Cancer	20.1	20	1
Forest Fire	59.3	100	1

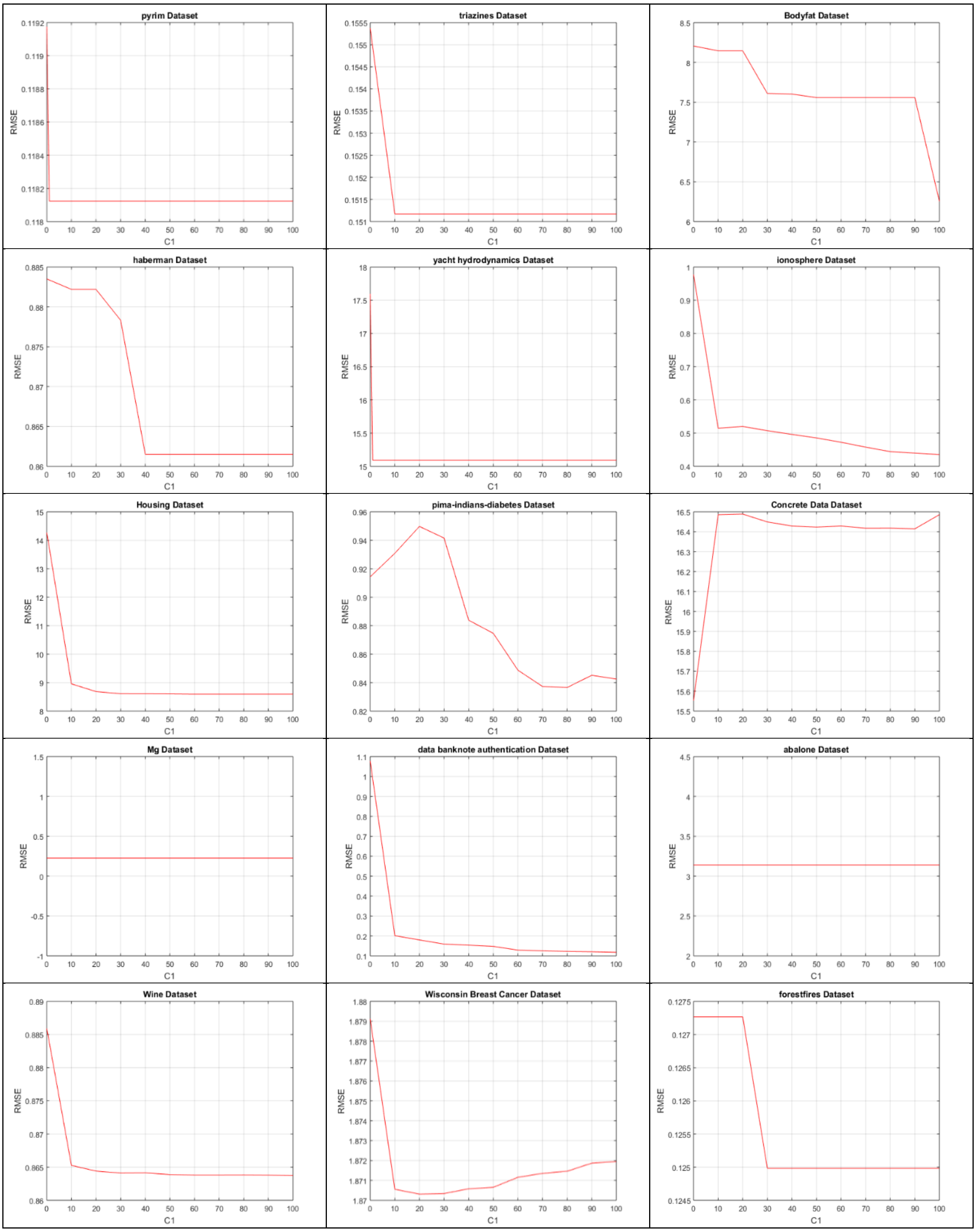
Table 3. The mean and standard deviation of RMSE of our proposed method, ε-SVR, ε-TSVR and pair v-SVR

Dataset	ε-SVR	ε-TSVR	pair v-SVR	proposed
Pyrimidine	0.0842± 0.0593	0.0656± 0.0471	0.0675±0 .0367	0.1180± 0.0615
Triazines	0.1503± 0.0338	0.1362± 0.0434	0.1456± 0.0457	0.1512± 0.0332
Bodyfat	4.7757± 0.8102	8.1441± 2.1848	7.6785± 1.5767	6.2215± 1.4876
Haberman	0.8664± 0.1086	0.8775± 0.1395	0.8452± 0.1065	0.8615± 0.1055
Yacht Hydrodynamics	10.9085± 1.2578	10.3360± 1.8207	10.5678± 1.5342	9.6190± 0.9907
Ionosphere	0.5148± 0.1096	0.5022± 0.1283	0.4322± 0.1164	0.4354± 0.1279
Housing	6.7392± 2.1573	8.9263± 3.5525	7.5678± 3.4234	8.5130± 3.3002
Pima Indians Diabetes	0.9112± 0.0640	0.9871± 0.0033	0.9454± 0.0197	0.8425± 0.0946
Concrete Compressive Strength	13.9188± 6.8642	14.6849± 7.1039	13.7545± 6.7433	14.6370± 5.2094
Mg	0.1014± 0.0123	0.1113± 0.0156	0.0945± 0.0154	0.0999± 0.0134
Banknote authentication	0.0857± 0.0184	0.0768± 0.0183	0.0956± 0.0224	0.1178± 0.0209
Abalone	2.0999± 0.6538	2.1265± 0.6173	2.0451± 0.9835	3.1384± 0.9881
Wine	0.8637± 0.0660	0.8685± 0.0576	0.8621± 0.0452	0.8601± 0.0355
Wisconsin Breast Cancer	1.8819± 0.5168	1.8847± 0.5224	1.8664± 0.4767	1.8703± 0.4350
Forest Fires	0.1974± 0.2502	0.1288± 0.2810	0.1198± 0.2576	0.1250± 0.2746

Table 4. The runtime of our proposed method, ε- SVR, ε-TSVR and pair v-SVR (seconds)

Dataset	ε-SVR	ε-TSVR	pair v-SVR	proposed
Pyrimidine	0.41	0.29	0.35	0.12
Triazines	3.44	2.12	2.87	1.71
Bodyfat	5.69	0.71	1.75	0.28
Haberman	8.34	0.78	2.76	0.43
Yacht Hydrodynamics	9.79	0.64	3.65	0.26

ionosphere	9.12	4.20	6.43	3.08
Housing	33.94	2.06	9.34	0.77
Pima Indians Diabetes	40.85	13.45	21.67	12.34
Concrete Compressive Strength	144.2	40.79	31.43	38.32
Mg	187.02	10.26	49.45	4.84
Banknote authentication	252.38	11.82	51.43	11.70
Abalone	14202.98	199.51	199.51	54.07
Wine	3906.94	1384.58	2232.32	920.94
Wisconsin Breast Cancer	2.87	1.92	2.02	1.53
Forest Fires	17.81	3.70	5.21	2.51



pyrim Dataset

triazines Dataset

Bodyfat Dataset

haberman Dataset

yacht hydrodynamics Dataset

ionosphere Dataset

Housing Dataset

pima-indians-diabetes Dataset

Concrete Data Dataset

Mg Dataset

data banknote authentication Dataset

abalone Dataset

Wine Dataset

Wisconsin Breast Cancer Dataset

forestfires Dataset

Figure 2. The sensitivity of our proposed method to the parameter C_1 for the best C_2 and σ reported in Table 2

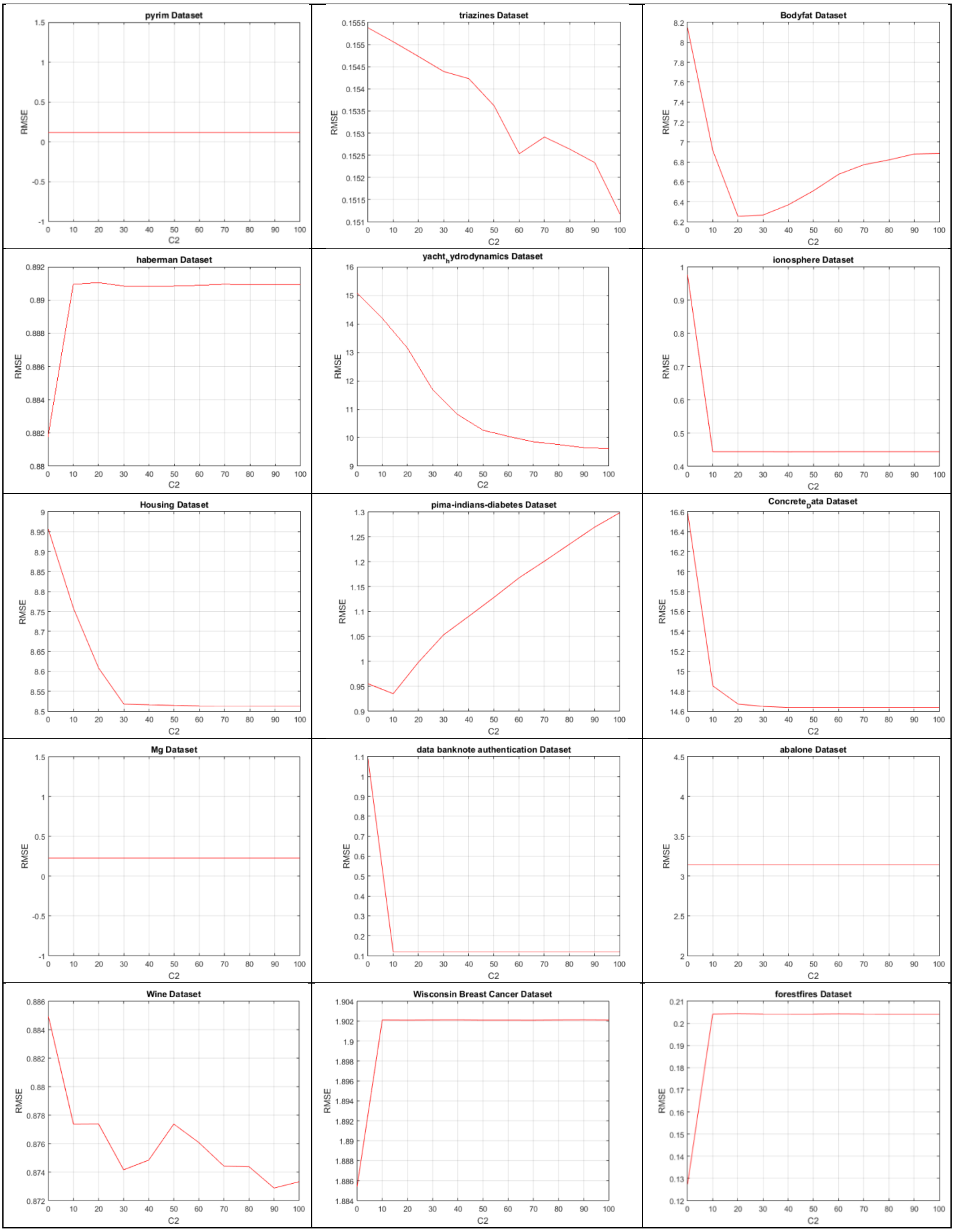
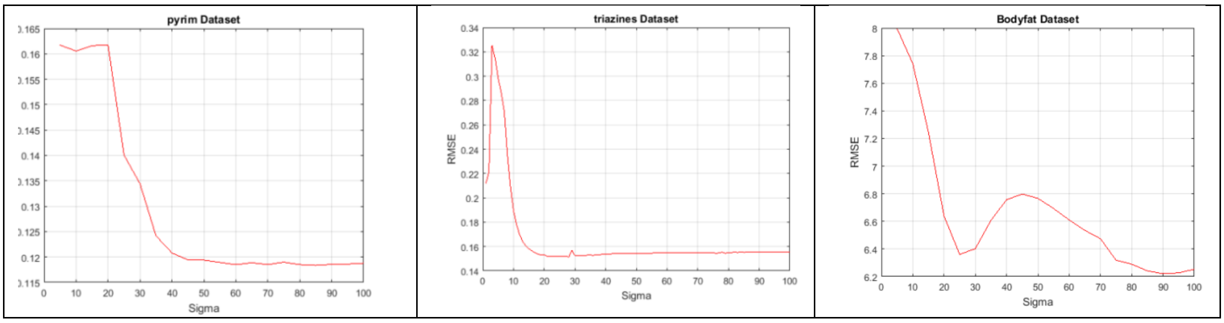


Figure 3. The sensitivity of our proposed method to the parameter C_2 for the best C_1 and σ reported in Table 2



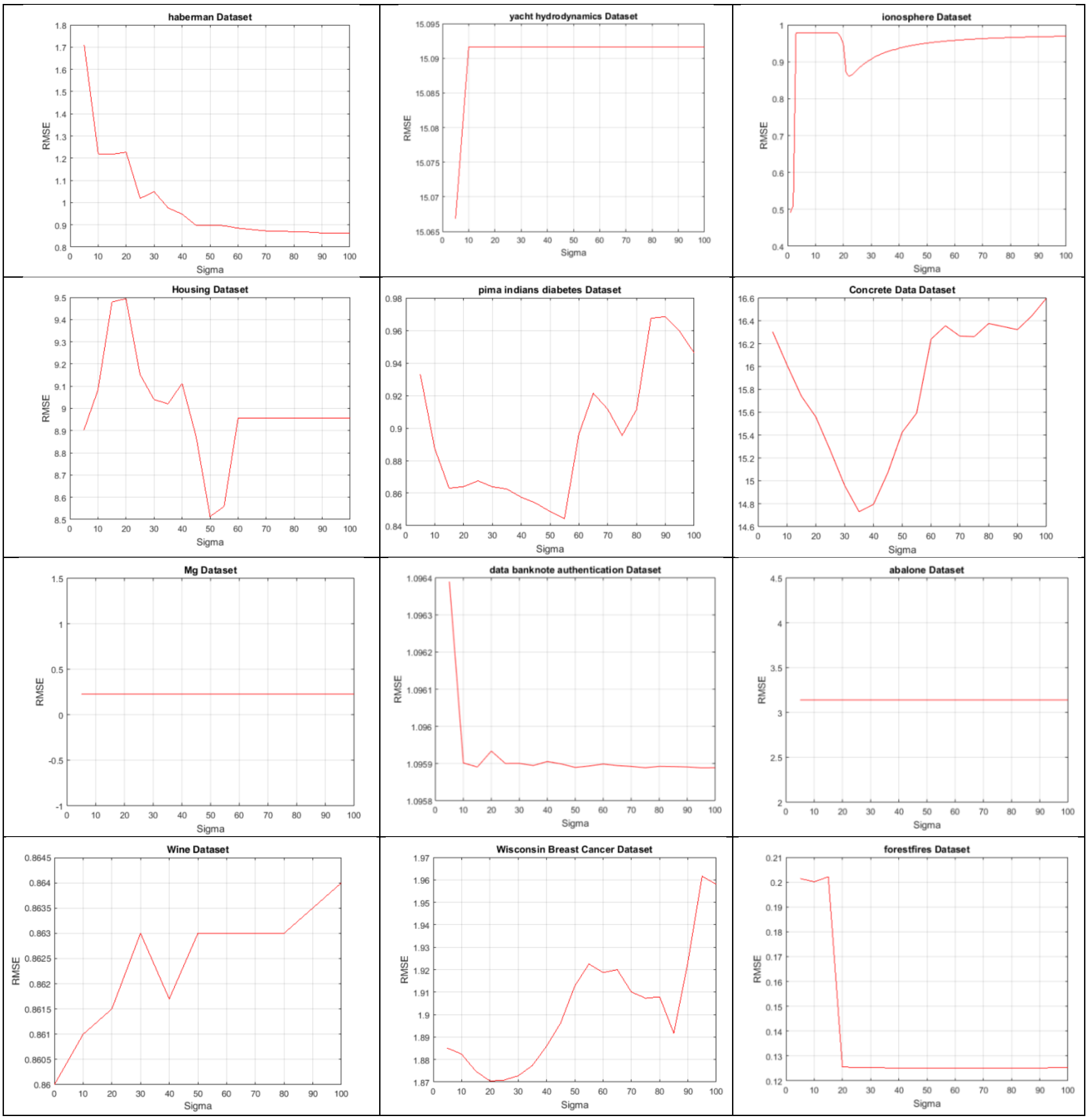


Figure 4. The sensitivity of our proposed method to the parameter σ for the best C_1 and C_2 reported in Table 2

5. CONCLUSIONS

The RLS method is one of the fastest methods for function estimation because it is an n + 1 variable problem with a close-form solution which is obtained by inverting an n × n matrix. The RLS has a high sensitivity to noisy data. Against this, ε-SVR has a good resistance against noise, but its runtime is not as good as the RLS runtime, because in order to solve the dual ε-SVR, we need to solve a quadratic programming problem with 2×n variables and linear constraints. ε-SVR supposes that the noise in the output variable is at most ε. Therefore, the center of a tube with radius ε, which is used as the estimated function, is determined in a way that the training data are located in that tube. Therefore, this method is robust to such noisy data. In this paper, to improve the runtime of ε-SVR, first, an initial estimated function was obtained using the RLS method. Then, unlike ε-SVR which uses all data to determine the lower and upper limits of the tube, our proposed method used the initial estimated function for determining the tube and the final estimated function. Strictly speaking, the data below and above the initial estimated function were used to estimate the upper and lower limits of the tube, respectively. Thus, the number of the model constraints and, consequently, the model runtime were reduced.

Our proposed method runtime is also lower than that of ε-TSVR and pair v-SVR because in each of ε-TSVR and pair v-SVR, two quadratic programming problems with n-variables and linear constraints are solved, while in our proposed method, an unconstrained quadratic programming problem with n+1 variables and a close form solution, and then, a quadratic programming problem with n-variables and linear constraints are solved. Solving an unconstrained quadratic programming problem is faster than a constrained quadratic programming problem of the same size. Our experiments on 15 benchmark dataset confirm that our proposed method is faster than ε-SVR, ε-TSVR and pair v-SVR, and its accuracy is comparable with the ε-SVR, ε-TSVR and pair v-SVR.

REFERENCES

[1] Otto Bretscher, S. (1997). Linear Algebra with Applications. USA: Prentice-Hall International Inc.

[2] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58(1): 267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

[3] Zou, H., Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2): 301-320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

[4] Vapnik, V. (2013). The nature of statistical learning theory. Springer Science & Business Media.

[5] Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L. (2000). New support vector algorithms. Neural Computation, 12(5): 1207-1245. https://doi.org/10.1162/089976600300015565

[6] Jeng, J.T., Chuang, C.C., Su, S.F. (2003). Support vector interval regression networks for interval regression analysis. Fuzzy Sets and Systems, 138(2): 283-300. https://doi.org/10.1016/S0165-0114(02)00570-5

[7] Hao, P.Y. (2009). Interval regression analysis using support vector networks. Fuzzy Sets and Systems, 160(17): 2466-2485. https://doi.org/10.1016/j.fss.2008.10.012

[8] Peng, X. (2010). TSVR: An efficient twin support vector machine for regression. Neural Networks, 23(3): 365-372. https://doi.org/10.1016/j.neunet.2009.07.002

[9] Shao, Y.H., Zhang, C.H., Yang, Z.M., Jing, L., Deng, N.Y. (2013). An ε-twin support vector machine for regression. Neural Computing and Applications, 23(1): 175-185. https://doi.org/10.1007/s00521-012-0924-3

[10] Peng, X., Xu, D., Kong, L.Y., Chen, D.J. (2016). L1-norm loss based twin support vector machine for data recognition. Information Sciences, 340-341: 86-103. https://doi.org/10.1016/j.ins.2016.01.023

[11] Hao, P.Y. (2017). Pair v-SVR: A novel and efficient pairing nu-support vector regression algorithm. IEEE Transactions on Neural Networks and Learning Systems, 28(11): 2503-2515. https://doi.org/10.1109/TNNLS.2016.2598182

[12] Huang, X., Shi, L., Pelckmans, K., Suykens, J.A.K. (2014). Asymmetric ν-tube support vector regression. Computational Statistics & Data Analysis, 77: 371-382. https://doi.org/10.1016/j.csda.2014.03.016

[13] Xu, Y., Li, X.Y., Pan, X.L., Yang, Z.J. (2017). Asymmetric ν-twin support vector regression. Neural Computing and Applications, 30: 1-16. https://doi.org/10.1007/s00521-017-2966-z

[14] Kuhn, H.W., Tucker, A.W. (2014). Nonlinear programming, in Traces and emergence of nonlinear programming. Springer, pp. 247-258.

NOMENCLATURE

x_i	i-th training data
y_i	Class label of i-th training data
$\xi_i, \hat{\xi}_i$	Slack variables for i-th training data
n	The number of training data
w	Weight vector of htperplane
b	Bias of hyperplane
$\alpha_i, \hat{\alpha}_i$	Lagrange coefficients for i-th training data
φ(.)	A function which map data from the input space into a high-dimensional feature space
k(.,.)	Kernel function
σ	Hyper-parameter of Gaussian kernel function
C_1,C_2,C_3,C_4	Hyper-parameters of models to control penalty terms
ε	Size of tube