Hindawi
Scientific Programming
Volume 2019, Article ID 7102946, 11 pages
https://doi.org/10.1155/2019/7102946

Research Article

Robust Proximal Support Vector Regression Based on Maximum Correntropy Criterion

Kuaini Wang,[1] Huimin Pei,[2] Xiaoshuai Ding,[3] and Ping Zhong[4]

[1] *College of Science, Xi'an Shiyou University, Xi'an, China*
[2] *School of Mathematics and Statistics, Jiangsu Normal University, Xuzhou, China*
[3] *School of Education, Xizang Minzu University, Xianyang, China*
[4] *College of Science, China Agricultural University, Beijing, China*

Correspondence should be addressed to Ping Zhong; zping@cau.edu.cn

Received 17 August 2018; Accepted 11 December 2018; Published 16 January 2019

Academic Editor: Antonio J. Peña

The robustness problem of the classical proximal support vector machine for regression estimation (PSVR) when confronting with samples in the presence of outliers is addressed in this paper. Correntropy is a local similarity measure between two arbitrary variables and has been proven the insensitivity to noises and outliers. Based on the maximum correntropy criterion (MCC), a correntropy-based robust PSVR framework is proposed, named as RPSVR-MCC. The half-quadratic optimization method is employed to solve the resultant optimization, and an iterative algorithm is developed to solve RPSVR-MCC. In each iteration, the complex optimization can be converted to a linear system of equations which can be easily solved by the widely popular optimization techniques. The experimental results on synthetic datasets and real-world benchmark datasets demonstrate that the effectiveness of the proposed method. Moreover, the superiority of the proposed algorithm is more evident in noisy environment, especially in the presence of outliers.

1. Introduction

Regression estimation from datasets is the basic subject in the field of machine learning. We are given the training samples of input vectors accompanied by the corresponding targets. The goal is to search a function estimation which best expresses the relationship between input vectors and their targets. In the real-world applications, many factors such as sampling errors, modeling errors, and instrument errors corrupt the training samples with noises and outliers, which are extremely far away from other samples [1]. Training the contaminated samples needs to build a robust model, which reflects the ability of dealing with a large number of outliers [2, 3]. Therefore, a robust method should be excellent enough to reject outliers and construct regression on only uncontaminated samples.

Support vector machine (SVM) was proposed by Vapnik et al. as a successful and powerful machine learning tool, which is based on the structural risk minimization principles [4–6]. It strikes a balance between the empirical risk and model complexity and can be derived by solving a convex quadratic programming problem. One efficient variant of SVM is the least squares SVM (LSSVM) introduced by Suykens et al. [7, 8], which replaces inequality constraints with equality ones, and then only needs to solve a linear system of equations. It tremendously accelerates the training speed. LSSVM has widely applied in a variety of real-world tasks. Different from LSSVM, proximal support vector machine (PSVM) is proposed by Fung and Mangasarian as another effective variant of SVM by adding an bias item $(1/2)b^2$ in the objective function, which leads to a strongly convex objective function [9]. It can be considered as a special case of regularized LS-SVM, which gives the simple optimal solution as well as the very fast computation time [9]. LSSVM and PSVM both attempt to minimize mean squares error (MSE) on training samples while simultaneously escaping from overfitting [7]. However, many researchers have shown that while training samples with noises and outliers, LSSVM and PSVM are sensitive to noises and outliers and lead to poor generalization performance and bad robustness since they tend to have the large loss owing to the SSE [8].

There are a lot of papers which concern about improving the robustness of LSSVM and PSVM. A direct method of enhancing LSSVM is outlier elimination by some advanced techniques [10, 11]. Wen et al. employed a criterion according to least trimmed squares to recursively eliminate outliers [10]. In [11], samples with the slack variables larger than the 3 times standard deviation were discard in the training phase. Another alternative approach develops the weight setting strategies to mitigate the influence of outliers. The key idea of this approach is how to set the weights of samples. Suykens et al. assigned different weights according to the error variables so that samples or outliers which were less important have small weights [8]. In [12, 13], researchers provided another weight rule that smaller weights were commonly given to those samples with large distances from other samples for the purpose of reducing their impact. Brabanter et al. [14] discussed four different types of the weighting function, which included Huber, Hampel, Logistic, and Myriad and concluded that in most cases, the performances of Logistic and Myriad weighting functions were ranked in the top two. But until now, it is unclear in theory whether these weighting strategies are appropriate or not owing to noises and outliers. Some researchers had made efforts to the nonconvex loss functions to obtain great improvement on generalization performance and robustness [15–20]. Nonconvex loss functions directing limited the maximal loss value directly and gave definite restrictions to the influence of outliers and led to much less sensitive to outliers. Recently, Correntropy derived from information theoretic learning was a generalized and local similarity measure between two arbitrary variables [21, 22]. It was proved as a robust measure which was appropriate for noise, nonGaussian noise and nonzero mean noise and had shown its superiority in robust learning of classification and regression [22, 23]. Maximum Correntropy Criterion (MCC) was introduced as a loss function to LSSVM for regression estimation to enhance the robustness of LSSVM, termed as RLSSVR-MCC [24].

Inspired by the advantages of maximum correntropy criterion, in this paper, we introduce maximum correntropy criterion into PSVM for regression estimation (PSVR) and derive a novel robust PSVR (RPSVR-MCC), attempting to suppress the negative influence of outliers and enhance the robustness of PSVR. The proposed model integrates maximum correntropy criterion, regularization technique, and kernel method. The proposed RPSVR-MCC cannot be solved directly by the classical optimization methods. The half-quadratic optimization technique is employed to derive an iterative algorithm for solving the corresponding optimization problem.

The contributions of the proposed RPSVR-MCC can be summarized as follows:

(1) A robust PSVM for the regression problem based on maximum correntropy criterion is proposed, which can reduce the negative influence of outliers in minimizing the objective function, which is interpreted as the regularized version of RLSSVR-MCC. Adding the bias item $(1/2)b^2$ derives simpler expression of optimal solution and wider choices of the kernel function.

(2) The resultant optimization problem of RPSVR-MCC can be transformed into a half-quadratic optimization. Furthermore, an iterative algorithm is developed which only needs to solve a linear system of equations in each iteration.

(3) The proposed RPSVR-MCC is illustrated on substantial datasets including six synthetic datasets and eleven real-world benchmark datasets under the cases of without outliers and with outliers. The experimental results show that RPSVR-MCC not only derives excellent estimation accuracy but also keeps stable performance in noisy environment, especially in the presence of outliers.

The reminder of this paper is organized as follows. We present briefly a background of PSVR and maximum correntropy criterion in Section 2. Section 3 proposes a robust PSVR in the kernel space and solves the proposed RPSVR-MCC by an iterative algorithm. The proposed method is evaluated by numerical experiments on synthetic datasets and benchmark datasets in Section 4. Section 5 concludes the paper.

2. Background

2.1. PSVR. In this section, we concisely present the basic principles of the classical PSVR. For more detail, the reader can refer to [9]. Consider a regression problem with a training dataset $\{(\mathbf{x}_i y_i)\}_{i=1}^{n}$, where the input vector $\mathbf{x}_i \in R^m$ and the corresponding target $y_i \in R$. The essence of the regression problem is to search a function $f(\mathbf{x})$ which best expresses the relationship between input vectors and their targets. PSVR is formulated as [9]

$$\min_{w,b,\xi_i} \quad \frac{1}{2}\|w\|^2 + \frac{1}{2}b^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2, \tag{1}$$

$$\text{s.t.} \quad w^\top \phi(x_i) + b - y_i = \xi_i, \quad i = 1, \ldots, n, \tag{2}$$

where ξ_i is the training error and $C > 0$ is the pre-given parameter that provides a tradeoff between the model complexity and empirical risk. To solve the optimization problem (1) and (2), we construct the following Lagrangian function:

$$\begin{aligned} L(w, b, \xi_i, \alpha_i) = {} & \frac{1}{2}\|w\|^2 + \frac{1}{2}b^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \\ & - \sum_{i=1}^{n}\alpha_i\left(w^\top \phi(x_i) + b - y_i - \xi_i\right). \end{aligned} \tag{3}$$

Utilizing the Karush-Kuhn-Tucker (KKT) conditions, training PSVR is equivalent to solving the dual problem in the form of a linear system of equations,

$$\left(K + \frac{1}{C} I_n + ee^\top\right)\alpha = y, \tag{4}$$

$$b = e^\top \alpha, \tag{5}$$

where I_n denotes $n \times n$ identity matrix, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top$, $e = (1, 1, \ldots, 1)^\top$, and $y = (y_1, y_2, \ldots, y_n)^\top$. $K = (K_{ij})_{n\times n}$ denotes the matrix with element $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$, and $k(\cdot, \cdot)$ is the kernel function which can be expressed as the inner product calculation in high dimensional feature space. Among all the kernel functions, the most popular choice is Gaussian kernel defined by $k(x_i, x_j) = \exp(-\sigma \|x_i - x_j\|^2)$, where $\sigma > 0$ is the kernel parameter.

The decision function of PSVR is

$$f(\mathbf{x}) = w^\top \phi(\mathbf{x}) + b = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b, \tag{6}$$

where α_i and b are the solutions of (4) and (5).

Different from LSSVM, there is a bias term $(1/2)b^2$ in the objective function of PSVM, and this formulation brings about the strong convexity of the objective function. The strong convexity plays a prominent part in the simpler type of optimal solution and derives that the kernel function of PSVM is not required to satisfy Mercer's theorem so that they can be selected from a wide range. PSVM can also be interpreted as a regularized version of LSSVM. The classical PSVR is based on MSE measurement that equally treats all the samples and has been considered to be sensitive to outliers, which would seriously affect the reliability. They are significant challenges mainly due to the unpredictable nature of the error, which may be arbitrarily large in magnitude and cannot be ignored or treated with methods devised for small noises [23].

2.2. Maximum Correntropy Criterion. Correntropy is a recently developed information-theoretic measure to deal with error distributions with nonGaussian characteristics [21, 22]. The MCC expresses the similarity between the predicted output and the real sample in the correntropy sense. Given two arbitrary random variables A and B, their correntropy can be defined by

$$V_\gamma(A, B) = E[k_\sigma(A - B)], \tag{7}$$

where $k_\gamma(\cdot)$ is the kernel function which satisfies Mercer's theory and $E(\cdot)$ denotes the mathematic expectation. It has a clear theoretical foundation and is symmetric, positive, and bounded [22, 23]. In practice, the joint distribution of A and B is commonly unknown and only a finite number of samples $(a_i, b_i), i = 1, 2, \ldots, n$ are given. Thus, correntropy can be estimated by the following expression:

$$\hat{V}_\gamma(A, B) = \frac{1}{n} \sum_{i=1}^{n} k_\gamma(a_i - b_i), \tag{8}$$

where $k_\gamma(\cdot)$ is the Gaussian kernel $k_\gamma(\cdot) \triangleq G(a - b, \gamma) = \exp(-(\|a - b\|^2/\gamma^2))$ with bandwidth γ. Therefore, we rewrite (8) as

$$\hat{V}_\gamma(A, B) = \frac{1}{n} \sum_{i=1}^{n} G(a_i - b_i, \gamma). \tag{9}$$

The maximum of correntropy of an error in (7) is called the maximum correntropy criterion (MCC) [21–23]. Different from the global similarity measure MSE, MCC is local since the value of correntropy mainly depends on the kernel function along the line $A = B$. Liu et al. proved that correntropy is a robust function for linear and nonlinear regression, and the kernel size controls all the properties of correntropy as one of its main merits [22].

3. Robust PSVR Based on Maximum Correntropy Criterion

As mentioned in Section 2.1, the classical PSVR model employs MSE measurement that has been considered to be sensitive to outliers. In order to address this issue, we introduce the correntropy in robust regression estimation. Replacing the third term of the objective function in (7) of PSVR with maximizing the correntropy between the true target and the predicted value $f(\mathbf{x}) = w^\top \phi(\mathbf{x}) + b$, a new criterion can be derived as follows:

$$\max_{w,b} \sum_{i=1}^{n} G(w^\top \phi(\mathbf{x}_i) + b - y_i, \gamma). \tag{10}$$

Different from the MSE measure, MCC in (10) differently considers the samples and gives more emphasis on the small error, which implies that if a target is contaminated or corrupted to be outlier, it will give small contribution to the objective function. Then we get the following maximum correntropy problem for training PSVR:

$$\max_{w,b} J(w, b) = \frac{C}{2} \sum_{i=1}^{n} G(w^\top \phi(\mathbf{x}_i) + b - y_i, \gamma) - \left(\frac{1}{2}\|w\|^2 + \frac{1}{2} b^2\right). \tag{11}$$

It combines maximum correntropy criterion, regularization technique, and kernel trick. However, the maximum correntropy in (11) is nonlinear, it is difficult to directly optimize. Recently, many researchers have made much effort to the half-quadratic technique [25], the expectation-maximization method [26], and the conjugate gradient algorithm [27], which are devoted to solve this optimization problem. In this paper, we use the half-quadratic technique to solve the optimization problem (11). By introducing an auxiliary variable, this method transforms the original objective function into a half quadratic objective function. According to the property of convex conjugated functions [27], the following proposition [25] holds (Algorithm 1).

Proposition 1. *There exists a convex conjugated function* $\varphi\colon R \longrightarrow R$*, such that*

```
Input: Training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, $C$, $\sigma$, $\gamma$, $K$, a integer $M > 0$, and a small real $\rho > 0$. Initialize $p_i = -\eta_i = -1$,  $i = 1, 2, \ldots, n$, the diagonal matrix $S^1 = (1/C)I_n$, and let $t = 1$.
    Step 1. Solve (25) and (26) to obtain $\alpha^t$ and $b^t$.
    Step 2. Obtain error variable $\xi_i^t$ by (22) and determine $p_i^t$ by (27).
    Step 3. Build the diagonal matrix $S^{t+1}$ and then solve (25) and (26) to obtain $\alpha^{t+1}$ and $b^{t+1}$.
    Step 4. Let $\beta = (\alpha^T, b)^T$. If $t > M$ or $\|\beta^{t+1} - \beta^t\| \leq \rho$, go to Step 5, else go to Step 2. Let $t = t + 1$.
    Step 5. Derive the final regression estimation by (6).
```

Algorithm 1: Half-quadratic optimization for RPSVR-MCC.

$$G(x, \gamma) = \max_{p<0}\left(p\frac{\|x\|^2}{\gamma^2} - \varphi(p) \right), \tag{12}$$

and for a fixed x, the maximum is reached at $p = -g(x, \gamma)$ [25].

Introducing (12) into the objective function of (11), the optimization (11) can be transformed into the following half-quadratic optimization according to Proposition 1,

$$\max_{w,b,p} \bar{J}(w, b, p) = \frac{C}{2}\sum_{i=1}^{n}\left(p_i\frac{(w^\top\phi(\mathbf{x}_i) + b - y_i)^2}{\gamma^2} - \varphi(p_i) \right) - \left(\frac{1}{2}\|w\|^2 + \frac{1}{2}b^2\right), \tag{13}$$

where $p = [p_1, p_2, \ldots, p_n]^\top$ stands for the auxiliary variables appeared in the half-quadratic optimization. According to Proposition 1, for a fixed (w, b), the following equation holds:

$$J(w, b) = \max_{p} \bar{J}(w, b, p). \tag{14}$$

Furthermore, we get

$$\max_{w,b} J(w, b) = \max_{w,b,p} \bar{J}(w, b, p). \tag{15}$$

Obviously, w, b, p can be derived by alternatingly optimizing with respect to (w, b) and p while holding the other fix. First, if p is fixed, (13) can be reduced as the following problem with respect to (w, b):

$$\max_{w,b} \frac{C}{2}\sum_{i=1}^{n} p_i\frac{(w^\top\phi(\mathbf{x}_i) + b - y_i)^2}{\gamma^2} - \left(\frac{1}{2}\|w\|^2 + \frac{1}{2}b^2\right). \tag{16}$$

The unconstrained optimization problem (16) can be expressed as the following constrained problem with respect to (w, b, ξ):

$$\min_{w,b,\xi} \quad \frac{1}{2}||w||^2 + \frac{1}{2}b^2 + \frac{C}{2}\sum_{i=1}^{n}\eta_i\frac{\xi_i^2}{\gamma^2}, \tag{17}$$

$$\text{s.t.} \quad w^\top\phi(\mathbf{x}_i) + b - y_i = \xi_i, \quad i = 1, \ldots, n, \tag{18}$$

where $\eta_i = -p_i$. The optimization problem (17)-(18) can be solved by constructing the following Lagrangian function:

$$\mathscr{L}(w, b, \xi_i, \alpha_i) = \frac{1}{2}||w||^2 + \frac{1}{2}b^2 + \frac{C}{2}\sum_{i=1}^{n}\eta_i\frac{\xi_i^2}{\sigma^2} - \sum_{i=1}^{n}\alpha_i(w^\top\phi(\mathbf{x}_i) + b - y_i - \xi_i). \tag{19}$$

From KKT conditions, we obtain

$$\frac{\partial\mathscr{L}}{\partial w} = w - \sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i) = 0 \Longrightarrow w = \sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i), \tag{20}$$

$$\frac{\partial\mathscr{L}}{\partial b} = b - \sum_{i=1}^{n}\alpha_i = 0 \Longrightarrow b = \sum_{i=1}^{n}\alpha_i, \tag{21}$$

$$\frac{\partial\mathscr{L}}{\partial\xi_i} = \frac{C\eta_i}{\gamma^2}\xi_i + \alpha_i = 0 \Longrightarrow \xi_i = -\frac{\gamma^2\alpha_i}{C\eta_i}, \tag{22}$$

$$\frac{\partial\mathscr{L}}{\partial\alpha_i} = 0 \Longrightarrow w^\top\phi(\mathbf{x}_i) + b - y_i = \xi_i. \tag{23}$$

Substituting (20)–(22) into (23), (20)–(23) can be written as a linear system of equations,

$$(K + S + ee^\top)\alpha = \mathbf{y}, \tag{24}$$

where $S = \text{diag}\{s_1, s_2, \ldots, s_n\}$ is a diagonal matrix with $s_i = -(\gamma^2/Cp_i) > 0$ and $e = (1, 1, \ldots, 1)^\top$. From (24) and (21), we get

$$\alpha = (K + S + ee^\top)^{-1}\mathbf{y}, \tag{25}$$

$$b = e^\top\alpha. \tag{26}$$

Second, the optimal p is directly obtained from Proposition 1 by

$$p_i = -g(\xi_i, \gamma) = -\exp\left(-\frac{\xi_i^2}{\gamma^2}\right). \tag{27}$$

Implementing half-quadratic optimization for RPSVR-MCC is given as Algorithm 1.

4. Experiments

In this section, we present experiments on six synthetic datasets and eleven benchmark datasets to validate the performance of the proposed RPSVR-MCC. To that end, we compare it with the least squares support vector regression (LSSVR) [7] and PSVR [9]; weighted LSSVR with the Hampel weight function

(WLSSVR-H) [8]; and weighted LSSVR with the Logistic weight function (WLSSVR-L) [14] and RLSSVR-MCC [24]. The Gaussian kernel is adopted in the experiments for all models. The performance of these models usually depends on parameter choices. All these models have two common parameters C and σ. In each algorithm, we select the optimal C^* and σ^* from the set $\{2^i | i = -9, -8, -7, \dots, 7, 8, 9\}$. The correntropy parameter γ in RPSVR-MCC is searched from the set $\{2^i | i = -5, -4, -3, \dots, 1, 2, 3\}$. In this paper, we choose the optimal parameters of these algorithms by the classical grid search such that they can derive best performance on the testing samples.

In order to evaluate the performances of the algorithms, we employ the following four regression accuracy measures [28], which are, respectively, defined as

(1) The root mean squares error (RMSE) and the mean absolute error (MAE),

$$\mathrm{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(y_i - \hat{y}_i \right)^2}, \tag{28}$$

$$\mathrm{MAE} = \frac{1}{m} \sum_{i=1}^{m} \left| y_i - \hat{y}_i \right|, \tag{29}$$

where y_i is the real target, $\hat{y}_i$ is the corresponding prediction, and m is the number of testing samples. $\overline{y} = (1/m) \sum_{i=1}^{m} y_i$ represents the average value of $y_1, y_2, \dots, y_m$. The smaller RMSE is, the better fitting performance is. MAE is also a popular deviation measurement between the real and predicted values.

(2) The ratio between the sum squared error SSE and the sum squared deviation testing samples SST (SSE/SST),

$$\frac{\mathrm{SSE}}{\mathrm{SST}} = \frac{\sum_{i=1}^{m} \left(\hat{y}_i - y_i \right)^2}{\sum_{i=1}^{m} \left(y_i - \overline{y} \right)^2}. \tag{30}$$

(3) The ratio between interpretable sum deviation SSR and SST (SSR/SST),

$$\frac{\mathrm{SSR}}{\mathrm{SST}} = \frac{\sum_{i=1}^{m} \left(\hat{y}_i - \overline{y}_i \right)^2}{\sum_{i=1}^{m} \left(y_i - \overline{y} \right)^2}. \tag{31}$$

In most situations, small SSE/SST shows good agreement between the real and predictions values. To derive smaller SSE/SST commonly follows an increase in SSR/SST. Nevertheless, the extremely small value of SSE/SST is in fact not good, for it probably means overfitting of the regressor. Therefore, a good estimator should strike balance between SSE/SST and SSR/SST.

4.1. Synthetic Datasets. In synthetic experiments, we consider the approximation of the following sinc function $y = \sin(3x)/3x$ which is a popular choice in regression estimation [28, 29]. We generate the synthetic datasets $\{(x_i, y_i)\}_{i=1}^{n}$ as follow:

$$y_i = \frac{\sin(3x_i)}{3x_i} + \eta_i, \quad x_i \in [-4, 4], \tag{32}$$

where η_i represents the different types of noises to obtain the contaminated datasets. In the experiments, the first and second kinds of noises η_i follow Gaussian distribution $N(0, 0.15^2)$ and $N(0, 0.3^2)$, respectively, and denote this two type noises as Type A and Type B. The third and fourth kinds of noises η_i follow uniform distribution $[-0.15, 0.15]$ and $[-0.3, 0.3]$, respectively, denoted as Type C and Type D noises. The fifth and last kinds of noises η_i follow student distribution $\mathrm{T}(4)$ and $\mathrm{T}(8)$, respectively, denoted as Type E and Type F noises, where $\mathrm{T}(c)$ is a student random variable with c degrees of freedom.

The goal of this evaluation is to measure the robustness of different algorithms against outliers. In order to avoid biased comparisons, we randomly generate ten independent groups of 850 samples according to the sinc function, and the training samples and test samples were 350 and 500, respectively. For each kind of noises, we first add them into training samples to obtain noisy samples for the contamination purpose and then randomly select 20% training samples and add large noises on their targets to simulate outliers. The test samples are clean, which are uniform from the sinc function without any noise. Table 1 reports the average performances of these six algorithms with ten independent runs under the different noises distributions. (·) behind the criterions (RMSE, MAE, SSE/SST, and SSR/SST) stands for the rank of this algorithm among these six algorithms. The optimal parameters $(C^*, \sigma^*, \gamma^*)$ are also listed in Table 1. Table 2 shows the average ranks of these algorithms with different types of noises. The best results are highlighted in bold.

From Tables 1 and 2, we can derive the results as follows:

(1) The classical LSSVR and PSVR obtain almost the dissatisfied results, reflected by its much higher RMSEs, MAEs, and SSE/SSTs. This shows that LSSVR and PSVR are sensitive to outliers. On the contrary, the weighted model (WLSSVR-H, WLSSVR-L) enhances the LSSVR and PSVR to a certain extent, yet less accurate than the RLSSVR-MCC and RPSVR-MCC.

(2) It has been shown that the proposed RPSVR-MCC performs best reflected by the smallest RMSE, MAE, and SSE/SST for most of the types of noises. Especially, the RPSVR obtains the four criteria all ranked first for Type E noises. However, in term of SSR/SST, RPSVR-MCC is not satisfactory.

(3) For the RMSE, MAE, and SSE/SST indexes in Table 2, the average rank of the RPSVR-MCC is better than the other algorithms. For the SSR/SST index, no obvious difference was found between these algorithms.

Table 1: Experiment results on $y = \sin(3x)/(3x)$.

Noise	Algorithm	$(C^{*}, \sigma^{*}, \gamma^{*})$	RMSE	MAE	SSE/SST	SSR/SST
Type A	LSSVR	$(2^{2}, 2^{-1}, /)$	0.0863(5.5)	0.0676(5)	0.0720(5)	1.0761(2)
	PSVR	$(2^{2}, 2^{-1}, /)$	0.0863(5.5)	0.0677(6)	0.0721(6)	**1.0762(1)**
	WLSSVR-H	$(2^{4}, 2^{-1}, /)$	0.0422(3)	0.0349(3)	0.0181(3)	1.0344(6)
	WLSSVR-L	$(2^{4}, 2^{-1}, /)$	0.0514(4)	0.0408(4)	0.0262(4)	1.0479(3)
	RLSSVR-MCC	$(2^{8}, 2^{-2}, 2^{-1})$	0.0409(2)	**0.0331(1.5)**	**0.0173(1.5)**	1.0446(4.5)
	RPSVR-MCC	$(2^{8}, 2^{-2}, 2^{-1})$	**0.0408(1)**	**0.0331(1.5)**	**0.0173(1.5)**	1.0446(4.5)
Type B	LSSVR	$(2^{-1}, 2^{0}, /)$	0.0979(5)	0.0795(5)	0.0944(5)	0.9374(6)
	PSVR	$(2^{-1}, 2^{0}, /)$	0.0980(6)	0.0796(6)	0.0945(6)	0.9378(5)
	WLSSVR-H	$(2^{2}, 2^{-1}, /)$	0.0723(3)	0.0599(3)	0.0503(3)	0.9976(4)
	WLSSVR-L	$(2^{3}, 2^{-1}, /)$	0.0771(4)	0.0642(4)	0.0588(4)	**1.0497(1)**
	RLSSVR-MCC	$(2^{3}, 2^{-1}, 2^{0})$	**0.0694(1.5)**	**0.0580(1.5)**	**0.0464(1.5)**	1.0399(2)
	RPSVR-MCC	$(2^{3}, 2^{-1}, 2^{0})$	**0.0694(1.5)**	**0.0580(1.5)**	**0.0464(1.5)**	1.0398(3)
Type C	LSSVR	$(2^{7}, 2^{-2}, /)$	0.0828(5.5)	0.0665(6)	0.0664(5.5)	**1.1137(1)**
	PSVR	$(2^{7}, 2^{-2}, /)$	0.0828(5.5)	0.0664(5)	0.0664(5.5)	1.1136(2)
	WLSSVR-H	$(2^{4}, 2^{-1}, /)$	0.0289(3)	0.0221(3)	0.0083(3)	0.9816(4)
	WLSSVR-L	$(2^{4}, 2^{-1}, /)$	0.0378(4)	0.0308(4)	0.0142(4)	1.0154(3)
	RLSSVR-MCC	$(2^{1}, 2^{-1}, 2^{-2})$	0.0276(2)	**0.0213(1.5)**	**0.0075(1.5)**	0.9737(5)
	RPSVR-MCC	$(2^{1}, 2^{-1}, 2^{-2})$	**0.0275(1)**	**0.0213(1.5)**	**0.0075(1.5)**	0.9736(6)
Type D	LSSVR	$(2^{-1}, 2^{0}, /)$	0.0899(5)	0.0707(5)	0.0772(5)	0.9488(6)
	PSVR	$(2^{-1}, 2^{0}, /)$	0.0900(6)	0.0708(6)	0.0774(6)	0.9492(5)
	WLSSVR-H	$(2^{1}, 2^{0}, /)$	**0.0531(1)**	**0.0424(1)**	**0.0287(1)**	1.0116(2)
	WLSSVR-L	$(2^{3}, 2^{-1}, /)$	0.0610(4)	0.0500(4)	0.0357(4)	**1.0495(1)**
	RLSSVR-MCC	$(2^{-1}, 2^{0}, 2^{-1})$	0.0546(2.5)	0.0443(2.5)	0.0293(2.5)	0.9703(4)
	RPSVR-MCC	$(2^{-1}, 2^{0}, 2^{-1})$	0.0546(2.5)	0.0443(2.5)	0.0293(2.5)	0.9704(3)
Type E	LSSVR	$(2^{-4}, 2^{1}, /)$	0.2120(6)	0.1749(6)	0.4268(6)	0.7142(4)
	PSVR	$(2^{-4}, 2^{1}, /)$	0.2109(5)	0.1742(5)	0.4226(5)	0.7110(5)
	WLSSVR-H	$(2^{-4}, 2^{1}, /)$	0.2004(4)	0.1636(4)	0.3836(4)	0.5573(6)
	WLSSVR-L	$(2^{-3}, 2^{1}, /)$	0.1887(3)	0.1552(2)	0.3434(3)	0.7500(3)
	RLSSVR-MCC	$(2^{0}, 2^{0}, 2^{1})$	0.1848(2)	0.1553(3)	0.3412(2)	0.7873(2)
	RPSVR-MCC	$(2^{0}, 2^{0}, 2^{1})$	**0.1845(1)**	**0.1548(1)**	**0.3405(1)**	**0.7881(1)**
Type F	LSSVR	$(2^{-3}, 2^{0}, /)$	0.1796(5)	0.1498(5)	0.3112(5)	0.9709(2)
	PSVR	$(2^{-3}, 2^{0}, /)$	0.1791(4)	0.1493(4)	0.3091(4)	0.9699(3)
	WLSSVR-H	$(2^{-3}, 2^{0}, /)$	0.1846(6)	0.1507(6)	0.3270(6)	0.8896(6)
	WLSSVR-L	$(2^{-2}, 2^{0}, /)$	**0.1730(1)**	**0.1406(1)**	**0.2990(1)**	**1.0589(1)**
	RLSSVR-MCC	$(2^{3}, 2^{0}, 2^{3})$	0.1770(3)	0.1466(3)	0.3019(3)	0.9197(4)
	RPSVR-MCC	$(2^{3}, 2^{0}, 2^{3})$	0.1765(2)	0.1461(2)	0.2996(2)	0.9185(5)

Table 2: The average ranks of these algorithms on $y = \sin(3x)/(3x)$ with different types of noises.

Algorithm	RMSE	MAE	SSE/SST	SSR/SST
LSSVR	5.3333	5.3333	5.25	3.5
PSVR	5.3333	5.3333	5.4167	3.5
WLSSVR-H	3.3333	3.3333	3.3333	4.6667
WLSSVR-L	3.3333	3.1667	3.3333	2
RLSSVR-MCC	2.1667	2.1667	2	3.5833
RPSVR-MCC	1.5	1.6667	1.6667	3.75

Figures 1–3 show one run regression estimations of these algorithms under Type A, Type C, and Type E noises. The results report that the curve derived by LSSVR and PSVR are disastrously damaged by outliers, regardless of the types of noises. Although WLSSVR-H and WLSSVR-L perform better than LSSVR and PSVR, they still have large deviations at some samples, especially outliers. We notice that the curve of the RPSVR-MCC always follows the original curve more closely for most of the test samples. On average, our proposed RPSVR-MCC can derive higher robustness and more satisfactory results in the presence of outliers than other algorithms.

4.2. Benchmark Datasets. In the real-world examples, we conduct experiments on eleven benchmark datasets to further evaluate the proposed RPSVR-MCC. They include Pyrimidines (Pyrim), Triazines, AutoMPG, Boston Housing (BH), Servo, and Abalone downloaded from the well-known UCI machine learning repository: https://archive.ics.uci.edu/ml/index.php, Bodyfat, Pollution and Concrete Compressive Strength (Concrete) from StatLib: http://lib.stat.cmu.edu/datasets/, and Machine CPU (MCPU) and Diabetes from the web page https://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html, which are widely appeared in the field of evaluating various regression algorithms.

The first column in Tables 3–5 displays the detailed information of these datasets including the numbers of the samples and attributes, training and test samples. For each dataset, we randomly divided them into two parts (training

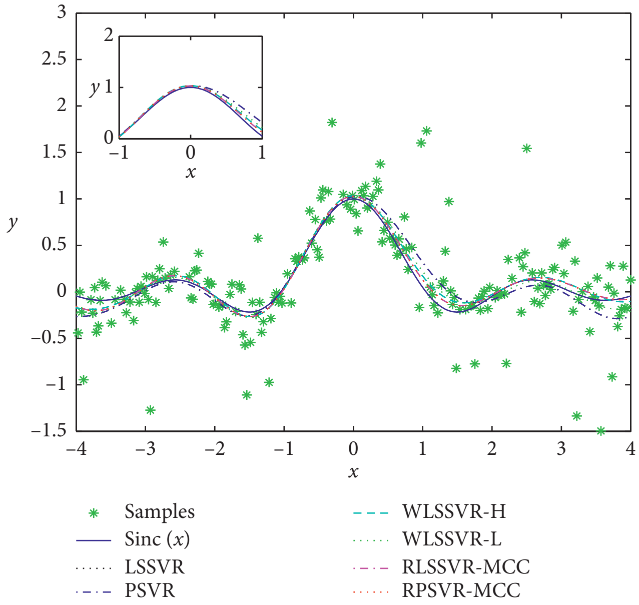


Figure 1: $y = \sin(3x)/(3x)$ under the Type A noise.

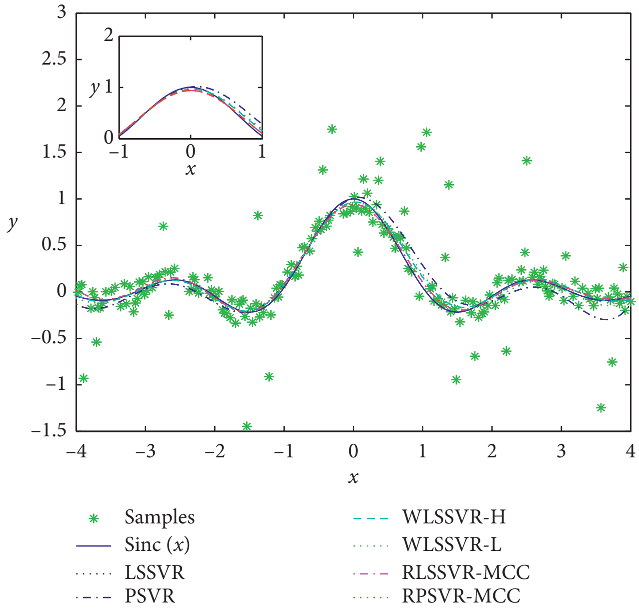


Figure 2: $y = \sin(3x)/(3x)$ under the Type C noise.

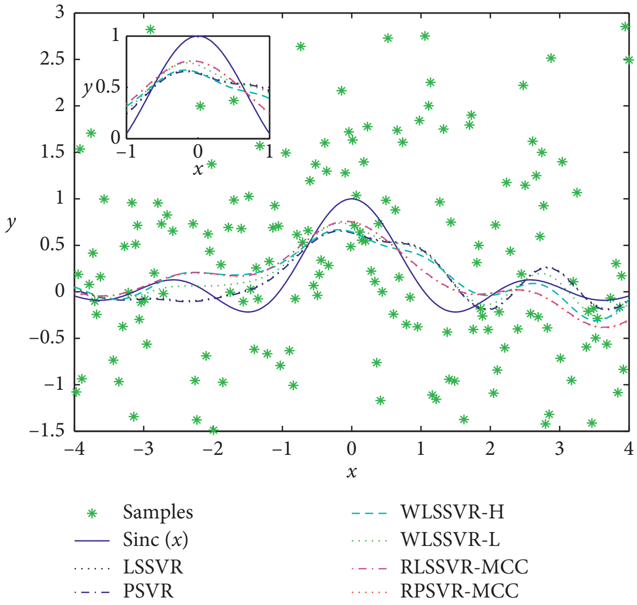


Figure 3: $y = \sin(3x)/(3x)$ under the Type E noise.

samples, test samples) according to the number in Tables 3–5. In this paper, we concern the robustness of outliers and samples were contaminated by large noises to simulate outliers in the training process. Concretely, the outliers are generated by randomly choosing 20% training samples and then adding noises to their targets according to the average value of the regression targets. Test samples are clean, which are not contaminated by any noise. All these regression algorithms are repeated ten times with different independent partition of training and test datasets. The ten accuracy measures were averaged to produce a single estimation. The performances and optimal parameters $(C^*, \sigma^*, \gamma^*)$ of these algorithms on ten datasets are reported in Tables 3–5.

In this subsection, we compare in detail the performance of the RPSVR-MCC with the other five algorithms. For the RMSE index, one can see in Tables 3–5 as follows:

(1) Under the case of without outliers, RPSVR-MCC is superior to LSSVR and PSVR on nine datasets except Concrete and performs better than WLSSVR-H and WLSSVR-L on eight datasets except AutoMPG and Servo. Meanwhile, it is more accurate than RLSSVR-MCC on seven datasets except Concrete and derives comparable results on Pyrim and Bodyfat.

(2) Under the case of with outliers, RPSVR-MCC is superior to LSSVR and PSVR on all the datasets and performs better than WLSSVR-H and WLSSVR-L on eight datasets except Pyrim and Servo. It obtains more satisfactory results than RLSSVR-MCC on six datasets except Pollution, Servo, and Concrete and derives comparable performance on Bodyfat. For the MAE and SSE/SST indexes, there are similar conclusions that on average, RPSVR-MCC outperforms these five algorithms under all cases.

In order to analyze the performance of these algorithms more accurately, we summarize the average ranks of these algorithms on benchmark datasets under the cases of without outliers in Table 6 and with outliers in Table 7. From Tables 6 and 7, in term of the RMSE, MAE, and SSE/SST indexes under the cases of without and with outliers, RPSVR-MCC always ranks first, and RLSSVR-MCC always ranks second among these six algorithms. The main reason is that RPSVR -MCC and RLSSVR-MCC both employ the robust loss function (maximum correntropy criterion) to

Table 3: Experimental results on Pyrim, Pollution, Servo, and Triazines without outliers (I) and with outliers (II).

Dataset		Algorithm	$(C^*, \sigma^*, \gamma^*)$	RMSE	MAE	SSE/SST	SSR/SST
Pyrim (74 × 27) (50, 24)	I	LSSVR	$(2^5, 2^{-3}, /)$	0.1028(4)	0.0606(3)	0.5618(4)	**0.6067(1)**
		PSVR	$(2^4, 2^{-3}, /)$	0.1036(6)	0.0608(4.5)	0.5715(6)	0.5628(3)
		WLSSVR-H	$(2^3, 2^{-3}, /)$	0.1013(3)	0.0610(6)	0.5502(3)	0.4216(6)
		WLSSVR-L	$(2^5, 2^{-3}, /)$	0.1032(5)	0.0608(4.5)	0.5658(5)	0.5747(2)
		RLSSVR-MCC	$(2^{-1}, 2^{-4}, 2^{-3})$	**0.1002(1.5)**	**0.0593(1)**	0.5440(2)	0.4788(4)
		RPSVR-MCC	$(2^{-1}, 2^{-4}, 2^{-3})$	**0.1002(1.5)**	0.0594(2)	**0.5432(1)**	0.4774(5)
	II	LSSVR	$(2^4, 2^{-3}, /)$	0.1056(4.5)	0.0649(3)	0.5925(3)	0.5729(3)
		PSVR	$(2^4, 2^{-3}, /)$	0.1063(6)	0.0653(5.5)	0.6022(6)	0.5819(2)
		WLSSVR-H	$(2^3, 2^{-3}, /)$	**0.1026(1)**	**0.0627(1)**	**0.5590(1)**	0.4788(6)
		WLSSVR-L	$(2^5, 2^{-3}, /)$	0.1056(4.5)	0.0651(4)	0.5947(4)	**0.5918(1)**
		RLSSVR-MCC	$(2^2, 2^{-3}, 2^{-3})$	0.1052(3)	0.0653(5.5)	0.5972(5)	0.4939(5)
		RPSVR-MCC	$(2^{-2}, 2^{-3}, 2^{-3})$	0.1044(2)	0.0646(2)	0.5837(2)	0.4941(4)
Pollution (60 × 16) (40, 20)	I	LSSVR	$(2^3, 2^{-2}, /)$	34.0594(5)	26.8079(5)	0.3921(5)	0.8644(2)
		PSVR	$(2^3, 2^{-2}, /)$	33.6752(2)	26.4472(2)	0.3821(2)	**0.8713(1)**
		WLSSVR-H	$(2^3, 2^{-2}, /)$	35.1268(6)	27.1519(6)	0.4097(6)	0.7284(6)
		WLSSVR-L	$(2^4, 2^{-2}, /)$	33.9942(3)	26.7418(3)	0.3895(4)	0.8631(3)
		RLSSVR-MCC	$(2^1, 2^{-2}, 2^1)$	33.9966(4)	26.7578(4)	0.3894(3)	0.8373(5)
		RPSVR-MCC	$(2^1, 2^{-2}, 2^1)$	**33.5695(1)**	**26.4044(1)**	**0.3784(1)**	0.8434(4)
	II	LSSVR	$(2^3, 2^{-3}, /)$	39.8592(6)	31.0642(5)	0.5358(6)	0.8910(2)
		PSVR	$(2^2, 2^{-2}, /)$	39.8591(5)	31.0858(6)	0.5326(5)	**0.9008(1)**
		WLSSVR-H	$(2^3, 2^{-2}, /)$	38.4257(4)	29.6791(4)	0.4890(4)	0.8240(3)
		WLSSVR-L	$(2^3, 2^{-2}, /)$	37.8460(3)	29.4832(3)	0.4749(3)	0.8182(4)
		RLSSVR-MCC	$(2^0, 2^{-3}, 2^{-2})$	**37.3457(1)**	**28.6822(1)**	0.4613(2)	0.7896(5)
		RPSVR-MCC	$(2^{-1}, 2^{-2}, 2^{-2})$	37.4479(2)	28.9103(2)	**0.4592(1)**	0.7791(6)
Servo (167 × 4) (100, 67)	I	LSSVR	$(2^6, 2^0, /)$	0.7055(6)	0.4017(6)	0.2121(6)	0.8589(2)
		PSVR	$(2^6, 2^0, /)$	0.7043(5)	0.4002(5)	0.2117(5)	**0.8596(1)**
		WLSSVR-H	$(2^7, 2^0, /)$	**0.6748(1)**	0.3530(3)	**0.1964(1)**	0.7610(4)
		WLSSVR-L	$(2^6, 2^0, /)$	0.6809(4)	0.3813(4)	0.1989(4)	0.7685(3)
		RLSSVR-MCC	$(2^1, 2^0, 2^{-3})$	0.6795(3)	**0.3416(1.5)**	0.1984(2)	0.7176(6)
		RPSVR-MCC	$(2^1, 2^0, 2^{-3})$	0.6794(2)	**0.3416(1.5)**	0.1985(3)	0.7181(5)
	II	LSSVR	$(2^5, 2^0, /)$	0.7054(6)	0.4194(6)	0.2126(6)	0.8353(2)
		PSVR	$(2^6, 2^0, /)$	0.7049(5)	0.4090(5)	0.2121(5)	**0.8695(1)**
		WLSSVR-H	$(2^8, 2^0, /)$	**0.6709(1)**	**0.3588(1)**	**0.1941(1)**	0.7946(3)
		WLSSVR-L	$(2^6, 2^0, /)$	0.6813(4)	0.3888(4)	0.1995(2)	0.7879(4)
		RLSSVR-MCC	$(2^2, 2^0, 2^{-2})$	0.6800(2)	0.3605(2)	0.2007(3)	0.7677(6)
		RPSVR-MCC	$(2^2, 2^0, 2^{-2})$	0.6802(3)	0.3606(3)	0.2011(4)	0.7683(5)
Triazines (186 × 60) (150, 36)	I	LSSVR	$(2^3, 2^{-4}, /)$	0.1443(6)	0.1063(6)	0.8759(6)	0.3521(2)
		PSVR	$(2^2, 2^{-4}, /)$	0.1428(5)	0.1060(5)	0.8536(5)	0.2769(6)
		WLSSVR-H	$(2^3, 2^{-3}, /)$	0.1422(3.5)	**0.1014(1)**	0.8449(3)	**0.3638(1)**
		WLSSVR-L	$(2^4, 2^{-4}, /)$	0.1422(3.5)	0.1033(4)	0.8469(4)	0.3337(3)
		RLSSVR-MCC	$(2^1, 2^{-4}, 2^{-1})$	0.1419(2)	0.1024(3)	0.8360(2)	0.2907(4)
		RPSVR-MCC	$(2^1, 2^{-4}, 2^{-1})$	**0.1404(1)**	0.1018(2)	**0.8205(1)**	0.2888(5)
	II	LSSVR	$(2^2, 2^{-4}, /)$	0.1481(6)	0.1126(6)	0.9295(6)	**0.3356(1)**
		PSVR	$(2^2, 2^{-4}, /)$	0.1466(4)	0.1122(5)	0.9129(5)	0.3334(2)
		WLSSVR-H	$(2^1, 2^{-3}, /)$	0.1474(5)	0.1097(4)	0.9080(4)	0.2471(6)
		WLSSVR-L	$(2^3, 2^{-4}, /)$	0.1464(3)	0.1096(3)	0.9033(3)	0.3082(3)
		RLSSVR-MCC	$(2^0, 2^{-4}, 2^{-1})$	0.1459(2)	0.1093(2)	0.8896(2)	0.2584(4)
		RPSVR-MCC	$(2^0, 2^{-4}, 2^{-1})$	**0.1442(1)**	**0.1088(1)**	**0.8722(1)**	0.2563(5)

limit the influence of outliers during the training phase. In addition, RPSVR-MCC outperforms RLSSVR-MCC in seven out of eleven datasets under the case of with outliers. For the SSR/SST index, the proposed RPSVR-MCC derives poor performance and ranks third among these algorithms. However, it is still superior to RLSSVR-MCC. On average, the RPSVR is more robust to outliers and has better generalization performance than the other algorithms.

5. Conclusion

In this paper, we introduce maximum correntropy criterion into the classical PSVR and propose a novel robust PSVR, namely, RPSVR-MCC, for dealing with regression estimation in noisy environment, especially in the presence of outliers. The proposed method integrates maximum correntropy criterion, regularization technique, and kernel

Table 4: Experimental results on MCPU, Bodyfat, AutoMPG, and BH without outliers (I) and with outliers (II).

Dataset		Algorithm	$(C^*, \sigma^*, \gamma^*)$	RMSE	MAE	SSE/SST	SSR/SST
MCPU (209 × 6) (150, 59)	I	LSSVR	$(2^{7}, 2^{-3}, /)$	53.3871(6)	27.3354(6)	0.1547 (6)	0.9474(4)
		PSVR	$(2^{7}, 2^{-3}, /)$	53.2070(5)	27.1529(4)	0.1530(5)	0.9419(6)
		WLSSVR-H	$(2^{9}, 2^{-3}, /)$	52.1390(3)	27.1578(5)	0.1410(3)	**0.9922(1)**
		WLSSVR-L	$(2^{8}, 2^{-3}, /)$	52.2495(4)	27.0081(3)	0.1427(4)	0.9249(5)
		RLSSVR-MCC	$(2^{3}, 2^{-3}, 2^{-3})$	51.8570(2)	**26.8429(1)**	0.1402(2)	0.9492(3)
		RPSVR-MCC	$(2^{2}, 2^{-3}, 2^{-3})$	**51.5925(1)**	27.0013(2)	**0.1388(1)**	0.9505(2)
	II	LSSVR	$(2^{7}, 2^{-3}, /)$	53.6345(6)	27.5341(5)	0.1567(6)	0.9485(2)
		PSVR	$(2^{7}, 2^{-3}, /)$	53.4728(5)	27.3651(4)	0.1551(5)	0.9431(5)
		WLSSVR-H	$(2^{9}, 2^{-3}, /)$	52.8168(4)	27.5543(6)	0.1489(4)	**1.0060(1)**
		WLSSVR-L	$(2^{8}, 2^{-3}, /)$	52.1319(3)	27.0399(3)	0.1436(3)	0.9335(6)
		RLSSVR-MCC	$(2^{3}, 2^{-3}, 2^{-3})$	50.8516(2)	26.5944(2)	0.1366(2)	0.9476(3)
		RPSVR-MCC	$(2^{3}, 2^{-3}, 2^{-3})$	**50.6647(1)**	**26.5101(1)**	**0.1355(1)**	0.9461(4)
Bodyfat (252 × 14) (200, 52)	I	LSSVR	$(2^{9}, 2^{-7}, /)$	0.0025(5.5)	0.0011(5.5)	0.0262(6)	0.9326(6)
		PSVR	$(2^{9}, 2^{-7}, /)$	0.0025(5.5)	0.0011(5.5)	0.0260(5)	0.9362(5)
		WLSSVR-H	$(2^{9}, 2^{-6}, /)$	0.0022(3.5)	7.1305E-4(3)	0.0233(3)	0.9477(3)
		WLSSVR-L	$(2^{9}, 2^{-6}, /)$	0.0022(3.5)	7.9633E-4(4)	0.0238(4)	0.9416(4)
		RLSSVR-MCC	$(2^{9}, 2^{-7}, 2^{-5})$	**0.0020(1.5)**	4.5370E-4(2)	**0.0224(1.5)**	**0.9781(1)**
		RPSVR-MCC	$(2^{8}, 2^{-7}, 2^{-5})$	**0.0020(1.5)**	**4.4860E-4(1)**	**0.0224(1.5)**	0.9765(2)
	II	LS-SVR	$(2^{3}, 2^{-5}, /)$	0.0075(5)	0.0056(6)	0.1868(5)	0.7799(6)
		PSVR	$(2^{8}, 2^{-9}, /)$	0.0076(6)	0.0055(5)	0.1970(6)	0.9459(3)
		WLSSVR-H	$(2^{9}, 2^{-7}, /)$	0.0031(3)	0.0017(3)	0.0372(3)	0.9148(4)
		WLSSVR-L	$(2^{9}, 2^{-8}, /)$	0.0037(4)	0.0023(4)	0.0476(4)	0.8947(5)
		RLSSVR-MCC	$(2^{4}, 2^{-9}, 2^{-5})$	**0.0021(1.5)**	6.7207E-4(2)	0.0225(2)	0.9652(2)
		RPSVR-MCC	$(2^{4}, 2^{-9}, 2^{-5})$	**0.0021(1.5)**	**6.6674E-4(1)**	**0.0224(1)**	**0.9670(1)**
AutoMPG (392 × 7) (300, 92)	I	LSSVR	$(2^{5}, 2^{0}, /)$	2.6050(6)	1.8858(5)	0.1057(6)	**0.8713(1)**
		PSVR	$(2^{4}, 2^{0}, /)$	2.6017(5)	1.8870(6)	0.1054(5)	0.8596(2)
		WLSSVR-H	$(2^{4}, 2^{1}, /)$	**2.5764(1)**	**1.8421(1)**	**0.1031(1)**	0.8519(5)
		WLSSVR-L	$(2^{6}, 2^{0}, /)$	2.5867(4)	1.8642(3)	0.1040(4)	0.8583(3)
		RLSSVR-MCC	$(2^{1}, 2^{0}, 2^{-2})$	2.5833(3)	1.8676(4)	0.1037(3)	0.8515(6)
		RPSVR-MCC	$(2^{1}, 2^{0}, 2^{-2})$	2.5812(2)	1.8637(2)	0.1035(2)	0.8520(4)
	II	LSSVR	$(2^{2}, 2^{0}, /)$	2.8980(6)	2.1601(6)	0.1308(6)	0.8421(5)
		PSVR	$(2^{2}, 2^{0}, /)$	2.8951(5)	2.1552(5)	0.1305(5)	0.8422(4)
		WLSSVR-H	$(2^{4}, 2^{0}, /)$	2.6334(3)	1.9324(3)	0.1080(3)	0.8459(3)
		WLSSVR-L	$(2^{4}, 2^{0}, /)$	2.7141(4)	1.9911(4)	0.1148(4)	0.8480(2)
		RLSSVR-MCC	$(2^{-2}, 2^{0}, 2^{-3})$	2.6194(2)	1.9087(2)	0.1065(2)	0.8376(6)
		RPSVR-MCC	$(2^{-1}, 2^{0}, 2^{-3})$	**2.6087(1)**	**1.8989(1)**	**0.1058(1)**	**0.8552(1)**
BH (506 × 13) (300, 206)	I	LSSVR	$(2^{7}, 2^{-1}, /)$	3.2018(4)	2.2747(6)	0.1276(4)	**0.8997(1)**
		PSVR	$(2^{7}, 2^{-1}, /)$	3.1967(2)	2.2734(5)	0.1271(2)	0.8994(2)
		WLSSVR-H	$(2^{7}, 2^{-1}, /)$	3.3161(6)	2.2609(2)	0.1375(6)	0.8816(6)
		WLSSVR-L	$(2^{8}, 2^{-1}, /)$	3.2319(5)	**2.2476(1)**	0.1302(5)	0.8925(5)
		RLSSVR-MCC	$(2^{7}, 2^{-1}, 2^{0})$	3.1993(3)	2.2693(4)	0.1274(3)	0.8985(3)
		RPSVR-MCC	$(2^{7}, 2^{-1}, 2^{0})$	**3.1941(1)**	2.2678(3)	**0.1270(1)**	0.8982(4)
	II	LSSVR	$(2^{4}, 2^{-3}, /)$	4.3860(5)	3.2163(5)	0.2402(5)	0.8563(4)
		PSVR	$(2^{5}, 2^{-3}, /)$	4.4092(6)	3.2482(6)	0.2433(6)	**0.9097(1)**
		WLSSVR-H	$(2^{5}, 2^{-2}, /)$	3.6464(3)	2.6132(3)	0.1661(3)	0.8808(2)
		WLSSVR-L	$(2^{8}, 2^{-4}, /)$	3.7909(4)	2.6898(4)	0.1800(4)	0.8676(3)
		RLSSVR-MCC	$(2^{1}, 2^{-3}, 2^{-4})$	3.4941(2)	**2.2972(1)**	0.1530(2)	0.8535(5)
		RPSVR-MCC	$(2^{-1}, 2^{-2}, 2^{-4})$	**3.4667(1)**	2.3061(2)	**0.1509(1)**	0.8487(6)

method. The half-quadratic optimization technique is adopted to derive an iterative algorithm for solving the corresponding optimization problem. Two groups of experiments on synthetic datasets and benchmark datasets with outliers are conducted, respectively, to test the robustness of the proposed algorithm. Compared with the classical LSSVR and PSVR, the proposed RPSVR-MCC obtains better generalization, especially under the case of with outliers. The possible reason is that the MCC is a local criterion of similarity and appropriate for samples with large outliers. Compared with other robust SVR algorithms (WLSSVR-H, WLSSVR-L), the proposed RPSVR-MCC outperforms in eight out of eleven datasets. The proposed algorithm is superior to the correntropy-based RLSSVR-MCC in seven out of eleven datasets. In conclusion, the developed algorithm not only derives higher robustness with better estimation accuracy and keeps stable performance in dealing with outliers.

This paper discusses outlier robustness which only concentrates on target noises. However, datasets in various practical applications are commonly contaminated by both

Table 5: Experimental results on Concrete and Abalone without outliers (I) and with outliers (II).

Dataset		Algorithm	$(C^*, \sigma^*, \gamma^*)$	RMSE	MAE	SSE/SST	SSR/SST
Concrete (1030 × 8) (500, 530)	I	LSSVR	$(2^8, 2^0, /)$	**6.2625(1)**	4.5672(2)	**0.1381(1.5)**	0.8974(4)
		PSVR	$(2^8, 2^0, /)$	6.2636(3)	4.5684(4)	0.1382(3.5)	0.8975(3)
		WLSSVR-H	$(2^9, 2^0, /)$	6.3971(6)	4.5935(6)	0.1443(6)	**0.9178(1)**
		WLSSVR-L	$(2^9, 2^0, /)$	6.3278(5)	**4.5636(1)**	0.1411(5)	0.9024(2)
		RLSSVR-MCC	$(2^8, 2^0, 2^0)$	6.2633(2)	4.5681(3)	**0.1381(1.5)**	0.8972(5.5)
		RPSVR-MCC	$(2^8, 2^0, 2^0)$	6.2644(4)	4.5693(5)	0.1382(3.5)	0.8972(5.5)
	II	LSSVR	$(2^9, 2^{-1}, /)$	7.0293(5)	5.1977(5)	0.1740(5.5)	0.9005(4)
		PSVR	$(2^9, 2^{-1}, /)$	7.0294(6)	5.1983(6)	0.1740(5.5)	0.9004(5)
		WLSSVR-H	$(2^9, 2^{-1}, /)$	7.0126(4)	5.1833(4)	0.1730(4)	0.8951(6)
		WLSSVR-L	$(2^8, 2^0, /)$	6.9237(3)	5.0999(3)	0.1687(3)	0.9034(3)
		RLSSVR-MCC	$(2^6, 2^{-1}, 2^{-2})$	**6.9051(1)**	**5.0862(1)**	**0.1679(1)**	**0.9159(1.5)**
		RPSVR-MCC	$(2^6, 2^{-1}, 2^{-2})$	6.9060(2)	5.0873(2)	0.1680(2)	**0.9159(1.5)**
Abalone (4177 × 7) (3000, 1177)	I	LSSVR	$(2^3, 2^2, /)$	2.1066(4)	1.5163(6)	0.4294(4)	**0.5688(1)**
		PSVR	$(2^3, 2^2, /)$	2.1057(3)	1.5158(5)	0.4290(3)	0.5684(2)
		WLSSVR-H	$(2^6, 2^1, /)$	2.1240(6)	**1.4863(1)**	0.4366(6)	0.5182(6)
		WLSSVR-L	$(2^4, 2^2, /)$	2.1086(5)	1.4915(2)	0.4302 (5)	0.5459(5)
		RLSSVR-MCC	$(2^1, 2^2, 2^{-1})$	2.1050(2)	1.5068(4)	0.4287(2)	0.5530(3)
		RPSVR-MCC	$(2^1, 2^2, 2^{-1})$	**2.1040(1)**	1.5063(3)	**0.4283(1)**	0.5525(4)
	II	LSSVR	$(2^1, 2^2, /)$	2.1319(6)	1.5454(5)	0.4399(6)	**0.5616(1)**
		PSVR	$(2^1, 2^2, /)$	2.1313(3.5)	1.5457(6)	0.4396(3.5)	0.5608(3)
		WLSSVR-H	$(2^3, 2^1, /)$	2.1313(3.5)	1.5370(2)	0.4396(3.5)	0.5534(5)
		WLSSVR-L	$(2^2, 2^2, /)$	2.1304(2)	**1.5354(1)**	0.4393(2)	0.5531(6)
		RLSSVR-MCC	$(2^1, 2^2, 2^0)$	2.1315(5)	1.5437(4)	0.4397(5)	0.5576(4)
		RPSVR-MCC	$(2^1, 2^1, 2^{-1})$	**2.1300(1)**	1.5389(3)	**0.4391(1)**	0.5614(2)

Table 6: The average ranks of these algorithms on benchmark datasets without outliers.

Algorithm	RMSE	MAE	SSE/SST	SSR/SST
LSSVR	4.75	5.05	4.85	2.4
PSVR	4.15	4.6	4.15	3.1
WLSSVR-H	3.9	3.4	3.5	3.9
WLSSVR-L	4.2	2.95	4.4	3.5
RLSSVR-MCC	2.4	2.75	2.2	4.05
RPSVR-MCC	1.6	2.25	1.9	4.05

Table 7: The average ranks of these algorithms on benchmark datasets with outliers.

Algorithm	RMSE	MAE	SSE/SST	SSR/SST
LSSVR	5.55	5.2	5.45	3
PSVR	5.15	5.35	5.2	2.7
WLSSVR-H	3.15	3.1	3.05	3.9
WLSSVR-L	3.45	3.3	3s.2	3.7
RLSSVR-MCC	2.15	2.25	2.6	4.15
RPSVR-MCC	1.55	1.8	1.5	3.55

features (or attributes) and targets noises. Our future work will focus on pinball loss [30] to yield a more useful and flexible method. In addition, the proposed approach can be extended to the one-class problem for imbalance classification [31], and we plan to address it in future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (Nos. 11626186, 11861060, and 11171346) and the Scientific Research Program Funded by Shaanxi Provincial Education Department (No. 18JK0623).

References

[1] V. J. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[2] P. Meer, C. V. Stewart, and D. E. Tyler, “Robust computer vision: an interdisciplinary challenge,” *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 1–7, 2000.

[3] S. Fidler, D. Skocaj, and A. Leonardis, “Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 337–350, 2006.

[4] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[5] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[6] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.

[7] J. A. K. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[8] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, “Weighted least squares support vector machines: robustness and sparse approximation,” *Neurocomputing*, vol. 48, no. 1–4, pp. 85–105, 2002.

[9] G. Fung and O. Mangasarian, “Proximal support vector machine classifiers,” in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’01)*, pp. 77–86, New York, NY, USA, 2001.

[10] W. Wen, Z. Hao, and X. Yang, “Robust least squares support vector machine based on recursive outlier elimination,” *Soft Computing*, vol. 14, no. 11, pp. 1241–1251, 2009.

[11] C.-C. Chuang and Z.-J. Lee, “Hybrid robust support vector machines for regression with outliers,” *Applied Soft Computing*, vol. 11, no. 1, pp. 64–72, 2011.

[12] W. Wen, Z. Hao, and X. Yang, “A heuristic weight-setting strategy and iteratively updating algorithm for weighted least-squares support vector regression,” *Neurocomputing*, vol. 71, no. 16, pp. 3096–3103, 2008.

[13] J. Liu, J. Li, W. Xu, and Y. Shi, “A weighted Lq adaptive least squares support vector machine classifiers—robust and sparse approximation,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 2253–2259, 2011.

[14] K. Brabanter, K. Pelckmans, J. Brabanter et al., “Robustness of kernel based regression: acomparison of iterative weighting schemes,” in *Proceedings of the 19th International Conference on Artificial Neural Networks, ICANN 2009*, Limassol, Cyprus, September 2009.

[15] Y. Wu and Y. Liu, “Robust truncated hinge loss support vector machines,” *Journal of the American Statistical Association*, vol. 102, no. 479, pp. 974–983, 2007.

[16] L. Wang, H. Jia, and J. Li, “Training robust support vector machine with smooth ramp loss in the primal space,” *Neurocomputing*, vol. 71, no. 13–15, pp. 3020–3025, 2008.

[17] Y. Zhao and J. Sun, “Robust support vector regression in the primal,” *Neural Networks*, vol. 21, no. 10, pp. 1548–1555, 2008.

[18] P. Zhong, “Training robust support vector regression with smooth non-convex loss function,” *Optimization Methods and Software*, vol. 27, no. 6, pp. 1039–1058, 2012.

[19] K. Wang and P. Zhong, “Robust non-convex least squares loss function for regression with outliers,” *Knowledge-Based Systems*, vol. 71, pp. 290–302, 2014.

[20] H. Pei, K. Wang, Q. Lin, and P. Zhong, “Robust semi-supervised extreme learning machine,” *Knowledge-Based Systems*, vol. 159, pp. 203–220, 2018.

[21] J. C. Principe, D. Xu, Q. Zhao, and J. W. Fisher III, “Learning from examples with information theoretic,” *Journal of VLSI Signal Processing*, vol. 26, no. 1-2, pp. 61–77, 2000.

[22] W. Liu, P. P. Pokharel, and J. C. Principe, “Correntropy: properties and applications in non-Gaussian signal processing,” *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[23] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, “A regularized correntropy framework for robust pattern recognition,” *Neural Computation*, vol. 23, no. 8, pp. 2074–2100, 2011.

[24] X. Chen, J. Yang, J. Liang, and Q. Ye, “Recursive robust least squares support vector regression based on maximum correntropy criterion,” *Neurocomputing*, vol. 97, pp. 63–73, 2012.

[25] X. Yuan and B. Hu, “Robust feature extraction via information theoretic learning,” in *Proceedings of the International Conference on Machine Learning*, ACM, Montreal, Canada, June 2009.

[26] S. Yang, H. Zha, S. Zhou, and B. Hu, “Variational graph embedding for globally and locally consistent feature extraction,” in *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 538–553, Springer, Bled, Slovenia, September 2009.

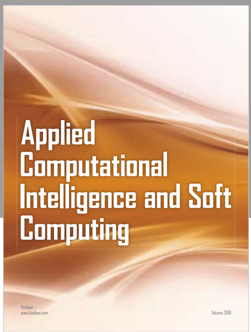
[27] Y. Grandvalet and Y. Bengio, “Entropy regularization,” in *Semi-Supervised Learning*, MIT Press, Boston, MA, USA, 2006.

[28] X. Peng, “TSVR: an efficient twin support vector machine for regression,” *Neural Networks*, vol. 23, no. 3, pp. 365–372, 2009.

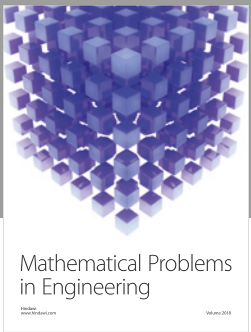
[29] P. Zhong, Y. Xu, and Y. Zhao, “Training twin support vector regression via linear programming,” *Neural Computing and Applications*, vol. 21, no. 2, pp. 399–407, 2011.

[30] X. Huang, L. Shi, and J. Suykens, “Support vector machine classifier with pinball loss,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 984–997, 2014.

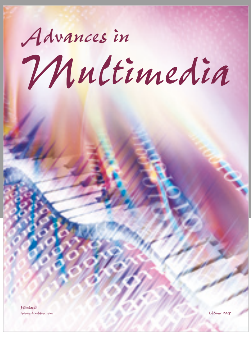
[31] Y.-S. Choi, “Least squares one-class support vector machine,” *Pattern Recognition Letters*, vol. 30, no. 13, pp. 1236–1240, 2009.

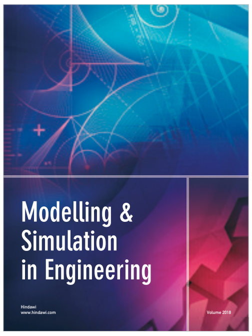

Applied Computational Intelligence and Soft Computing

The Scientific World Journal


Mathematical Problems in Engineering

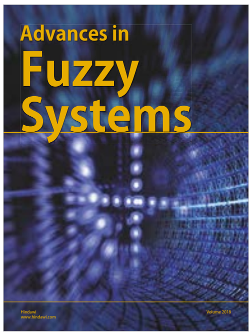
Journal of Engineering

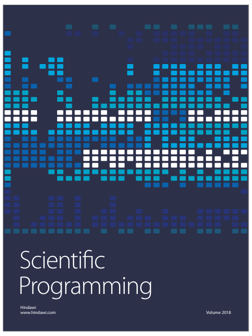

Advances in Multimedia

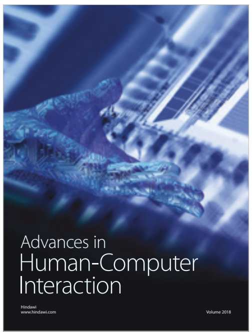

Modelling & Simulation in Engineering

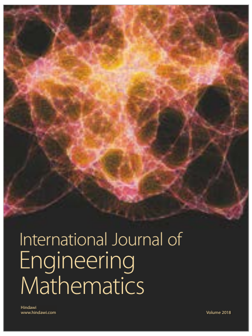
Advances in Artificial Intelligence

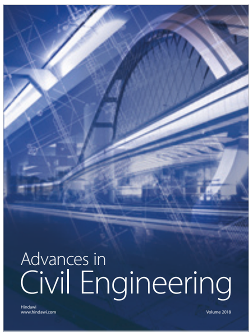
International Journal of Reconfigurable Computing

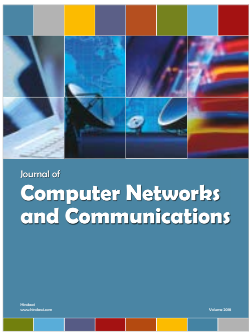

Advances in Fuzzy Systems


Scientific Programming


Advances in Human-Computer Interaction

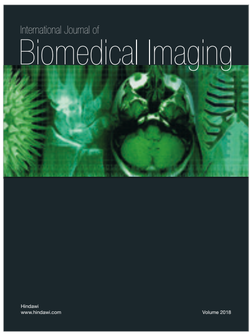

International Journal of Engineering Mathematics


Advances in Civil Engineering

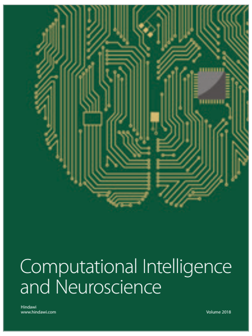

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology


International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience