Research Article

An Efficient Feature Weighting Method for Support Vector Regression

Minghua Xie ^{1,2}, Lili Xie ² and Peidong Zhu¹

¹*School of Electronic Information and Electrical Engineering, Changsha University, Changsha 410022, China*

²*School of Automatic, Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Lili Xie; xielili@nwpu.edu.cn

Received 17 December 2020; Revised 9 February 2021; Accepted 22 February 2021; Published 2 March 2021

Academic Editor: K. M. Pandey

Copyright © 2021 Minghua Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Support vector regression (SVR) is a powerful kernel-based method which has been successfully applied in regression problems. Regarding the feature-weighted SVR algorithms, its contribution to model output has been taken into account. However, the performance of the model is subject to the feature weights and the time consumption on training. In the paper, an efficient feature-weighted SVR is proposed. Firstly, the value constraint of each weight is obtained according to the maximal information coefficient which reveals the relationship between each input feature and output. Then, the constrained particle swarm optimization (PSO) algorithm is employed to optimize the feature weights and the hyperparameters simultaneously. Finally, the optimal weights are used to modify the kernel function. Simulation experiments were conducted on four synthetic datasets and seven real datasets by using the proposed model, classical SVR, and some state-of-the-art feature-weighted SVR models. The results show that the proposed method has the superior generalization ability within acceptable time.

1. Introduction

Support vector regression (SVR) is widely applied for regression problems due to its ability of converting the original low-dimensional problem to a high-dimensional kernel space linear problem by introducing kernel functions [1, 2]. The empirical risk and the confidence interval are balanced by adopting the principle of Structural Risk Minimization [3, 4]. SVR showed state-of-the-art performance in system modeling, such as remaining life prediction for lithium-ion batteries, electricity load forecasting, industrial analysis, coal flotation responses prediction, and so on [5–9].

Feature weighting is an effective method to solve the problem that all features are treated equally. It makes the feature importance better matched with its influence on the kernel space. Our previous work [10] has analyzed the necessity of feature weighting and verified it by using the grid search (GS) method to select the optimal combination of weights. Grid search is an exhaustive search method because its computational cost is very high. Therefore, an effective identification of weights is needed. The grey

correlation degree (GCD) theory was employed to calculate the weights for input features [11, 12]. The main idea of GCD is to evaluate the relationship between the input feature and the output of training samples according to the similarity of the geometric shape of sequence curves. The higher the similarity between the curves, the more important the corresponding input feature. However, the importance of the feature whose geometric shape is not similar to the output may be underestimated. Hou et al. developed a weighted kernel function for SVR based on maximal information coefficient (MIC) which is a distinct correlation statistic [13]. The relationship of both linear and nonlinear between input feature and output was evaluated. MIC can accurately identify the most important features of the input dataset. However, it depends too much on the quality and size of training data. Furthermore, it does not guarantee that the obtained weight combination is optimal, especially when MIC is regarded as an optimal weight. Wen et al. proposed a novel attribute feature weighting method called variable-weighted support vector machine [14]. In this method, the feature weights and optimal model parameters were both

tuned by particle swarm optimization (PSO). The similar methods have also appeared in literature [15, 16]. A more flexible rational model can be constructed based on the PSO algorithm. Its further application in feature-weighted SVR is hindered due to the drawbacks of slow convergence and the local optima nevertheless.

The paper proposed an efficient feature weighting method for support vector regression. Firstly, the MIC of each feature is calculated to reveal its relationship with output. The larger the value of MIC, the greater its contribution. The value constraint of each weight is obtained according to the MIC. Secondly, an improved constrained PSO is employed to tune the optimal feature weights and model parameters under the abovementioned constraints. Finally, the optimal weights are used to modify the kernel function.

The contributions of the work are two folds. Firstly, regarding the feature weighting with limited training samples, the contribution is hard to be calculated accurately. In addition, there is no evidence showing that the contribution can be regarded as an optimal weight. In the paper, the value constraint of weight is obtained according to the MIC. Therefore, the above problems have been avoided. Secondly, both weights and parameters are tuned by a constrained PSO which can converge to the global optima more efficiently to obtain superior generalization ability in the case of acceptable time consumption.

The paper is organized as follows: in Section 2, the SVR modeling method is briefly described. In Section 3, firstly, the value constraint of each weight is obtained according to the MIC. Then, the constrained PSO algorithm is explicitly illustrated. Simulation examples are given in Section 4. In Section 5, general conclusions are given.

2. Basic Review of SVR

The training dataset is given as $T = \{(x_1, y_1), \dots, (x_l, y_l)\} \in (R^m \times Y)^l$, where $x_i \in R^m$ is an input sample with m features and $y_i \in Y = R$ is the output, $i = 1, \dots, l$. The SVR model function can be regarded as a hyperplane in the kernel space. It is expressed as follows:

$$f(x) = \langle w, \phi(x) \rangle + b, \quad (1)$$

where $\phi(x)$ maps the low-dimensional input features to a high-dimensional kernel space, $w \in R^n$ is a weight vector of the hyperplane, and b is a bias term.

An insensitive loss function $\varepsilon > 0$ is introduced to avoid overfitting, and additional nonnegative slack variables ξ_i, ξ_i^* are adopted to weaken the constraints of some certain sample points. SVR modeling is formulated as a convex quadratic programming problem expressed as follows:

$$\min_{w, \xi, \xi^*} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*), \quad (2)$$

subject to

$$\begin{cases} y_i - f(x_i) \leq \varepsilon + \xi_i, \\ f(x_i) - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \end{cases} \quad (3)$$

where $C > 0$ is a penalty parameter. The above convex quadratic programming problem can be solved by constructing a Lagrange function.

The above convex quadratic programming problem can be reformulated by constructing the Lagrange function:

$$L = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) - \sum_{i=1}^l (\xi_i \eta_i + \xi_i^* \eta_i^*) - \sum_{i=1}^l \alpha_i [\xi_i + \varepsilon + y_i - \langle w \cdot \phi(x_i) \rangle - b] - \sum_{i=1}^l \alpha_i^* [\xi_i + \varepsilon - y_i + \langle w \cdot \phi(x_i) \rangle + b], \quad (4)$$

where $\alpha_i, \alpha_i^* \geq 0$ and $\eta_i, \eta_i^* \geq 0$ are Lagrange multipliers.

The kernel function $K(\cdot, \cdot)$, which satisfies the Mercer condition, is introduced to replace the inner product of the high-dimensional kernel space in equation (4). The commonly used kernel function is Gaussian kernel [17]. It is expressed as follows:

$$K(x_i, x) = \exp\left(-\gamma \|x_i - x\|^2\right), \quad (5)$$

where γ is the kernel parameter that controls the width of the Gaussian kernel.

The optimized problem can be expressed as follows:

$$\min_{\alpha, \alpha^*} \frac{1}{2} \sum_{i=1, j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) + \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) - \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i, \quad (6)$$

subject to

$$\left\{ \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, 0 \leq \alpha_i, \alpha_i^* \leq C. \right. \quad (7)$$

The solution can be obtained as $\bar{\alpha} = (\bar{\alpha}_1, \bar{\alpha}_1^*, \dots, \bar{\alpha}_l, \bar{\alpha}_l^*)^T$. The model function $f(x)$ can be represented as follows:

$$f(x) = \sum_{i=1}^l (\bar{\alpha}_i^* - \bar{\alpha}_i) K(x_i, x) + \bar{b}. \quad (8)$$

When $0 < \bar{\alpha}_j < C$ is selected, $\bar{b}$ can be expressed as follows:

$$\bar{b} = y_j - \sum_{i=1}^l (\bar{\alpha}_i^* - \alpha_i) K(x_i, x_j) + \varepsilon. \quad (9)$$

When $0 < \bar{\alpha}_k^* < C$ is selected, $\bar{b}$ can be expressed as follows:

$$\bar{b} = y_k - \sum_{i=1}^l (\bar{\alpha}_i^* - \alpha_i) K(x_i, x_k) - \varepsilon. \quad (10)$$

3. Feature-Weighted Support Vector Regression

3.1. Method of Weight Constraint. The j^{th} input feature of the training set with l samples is $F_j = \{x_{1j}, x_{2j}, \dots, x_{lj}\}$, the output sequence is $Y = \{y_1, y_2, \dots, y_l\}$, $D_j = \{F_j^T, Y^T\}$. A $p - by - q$ grid G is drawn on the scatterplot of D_j that partitions the data to encapsulate that relationship. Both p and q are positive integers.

For each pair (p, q) , the possible largest mutual information achieved by any $p - by - q$ grid and applied to D_j is found as follows:

$$I^*(D_j, p, q) = \max I(D_j | G). \quad (11)$$

Then, the mutual information value is normalized to ensure a fair comparison between grids of different dimensions:

$$M(D_j)_{p,q} = \frac{I^*(D_j, p, q)}{\log \min\{p, q\}}. \quad (12)$$

$M = (M(D_j)_{p,q})$ is defined as a characteristic matrix where $M(D_j)_{p,q}$ is the highest normalized mutual information achieved by any $p - by - q$ grid.

Finally, the maximum value in M is defined as MIC as follows:

$$\text{MIC}(D_j) = \max_{pq \leq B(l)} \{M(D_j)_{p,q}\}, \quad (13)$$

where B is a function of samples size. It is usually set to $B(l) = l^{0.6}$.

When the j^{th} input feature and output are statistically independent, $\text{MIC}(D_j)$ tends to 0. When they are entirely correlated, $\text{MIC}(D_j)$ tends to 1.

3.2. Implementation of the Constrained PSO. PSO is a widely used optimization tool based on a population, where each member is dubbed as particle. It simulates bird's searching food. Each particle is a potential solution to the optimization problem. Particles fly through the problem space according to simple mathematical formulae which are deduced from their position and velocity.

Each particle's movement is influenced by its personal best-known position (pBest), but it is also guided toward the global best position (gBest), which is found by other particles. Assume the solution space is N -dimensional; the position, the velocity, and the pBest of i^{th} particle are represented as $p_i = (p_{i1}, p_{i2}, \dots, p_{iN})$, $v_i = (v_{i1}, v_{i2}, \dots, v_{iN})$, and $b_i = (b_{i1}, b_{i2}, \dots, b_{iN})$, respectively. The gBest is represented as $g = (g_1, g_2, \dots, g_N)$. At each iterative process, i^{th} particle updates its velocity and position according to the following formulae:

$$\begin{aligned} v_{in}(\text{new}) &= w' \times v_{in}(\text{old}) + c_1 \times r_1 \times (b_{in} - p_{in}) + c_2 \times r_2 \times (g_n - p_{in}), \\ p_{in}(\text{new}) &= p_{in}(\text{old}) + \mu \times v_{in}(\text{new}), \end{aligned} \quad (14)$$

where w' is a weighting factor to balance the global search and local search, c_1 and c_2 are both learning factors, r_1 and r_2 are random values between 0 and 1, and μ is a parameter about flying times.

In the paper, PSO is employed to search the optimal hyperparameters (C, γ, ε) and the optimal combination of feature weights $(w_1, w_2, \dots, w_m)$ by minimizing the fitness function. The values of the MIC are used to constrain the weights. With the movement of the particles, both the optimal hyperparameters and the optimal weights will be obtained. The root mean square error (RMSE) is usually employed to evaluate the feasibility of the SVR method. It is selected as fitness function:

$$\text{RMSE} = \sqrt{\frac{1}{l} \sum_{i=1}^l [y_i - f(x_i)]^2}, \quad (15)$$

where y_i is the actual output sample and $f(x_i)$ is its corresponding predicted value. The smaller the value of RMSE, the better its generalization ability.

3.3. Steps of the Feature-Weighted Support Vector Regression. Steps of the proposed feature weighting support vector regression (MP-FWSVR) modeling method based on the MIC and the PSO are as follows:

Step 1. MIC between each feature and output is calculated.

Step 2. The values of the MIC are used to constrain the weights as follows:

$$\kappa_1 \frac{\text{MIC}(D_i)}{\max(F_i) - \min(F_i)} \leq w_i \leq \kappa_2 \frac{\text{MIC}(D_i)}{\max(F_i) - \min(F_i)}, \quad (16)$$

where κ_1 and κ_2 are set to 0.01 and 1, respectively.

Step 3. The i^{th} particle is looked as a point in the $(2 + m)$ -dimensional space and represented as $p_i = (C_i, \gamma_i, w_{i1}, w_{i2}, \dots, w_{im})$, and its velocity is represented as $v_i = (v_{C_i}, v_{\gamma_i}, v_{w_{i1}}, v_{w_{i2}}, \dots, v_{w_{im}})$.

Step 4. Obtain the optimal hyperparameters (C, γ, ϵ) and the optimal combination of feature weights $(w_1, w_2, \dots, w_m)$ by using PSO.

Step 5. Regarding the k^{th} feature, if a weight value w_k is given, the kernel element will be changed as

$$K_w(x_i, x_j) = \exp\left(-\gamma \left(\sum_{k=1}^m (w_k(x_{ik} - x_{jk}))^2\right)\right). \quad (17)$$

The model function $f(x)$ can be further developed as follows:

$$f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i) K_w(x_i, x) + b. \quad (18)$$

4. Simulation Examples

In this section, four synthetic datasets and seven real datasets are employed to evaluate the feasibility of the MP-FWSVR. All the codes are compiled on a Windows 10 PC with Intel Core i5-3470 CPU (3.2 GHz) and 12.0 GB RAM by MATLAB R2013a. The SVR algorithm is implemented via LIBSVM 3.22 [18].

4.1. Experiments on Artificial Synthetic Datasets. The definitions of four functions are listed in Table 1. σ is the added Gaussian noise with a mean zero and a standard deviation of 0.01.

The synthetic dataset “F1” is chosen as an example. The training data of Feature 1 (x_{i1}) and Feature 2 (x_{i2}) are taken from the sinusoidal functions of 0.01 Hz and 0.05 Hz, respectively. Their test datasets are extracted from linear functions and the sinusoidal signal 0.125 Hz, respectively. The training size and test size are both 101×2 . The two datasets are shown in Figure 1.

Firstly, we compare the optimal weights obtained by proposed MP algorithm with GS, MIC, and PSO algorithms. For MP and PSO, the parameters $w' = 0.6$, $c_1 = 1.5$, $c_2 = 1.7$, and $\mu = 1$. The number of particles is set to 20, and the iteration count is 200. The searching range of hyperparameters (C, γ, ϵ) is $[2^{-8}, 2^9]$, $[2^{-8}, 2^{10}]$, and $[10^{-3}, 1]$,

respectively. For GS and MIC, the optimal weights (w_1, w_2) of Feature 1 and Feature 2 are (0.01, 0.0001) and (0.9496, 0.2461), respectively. For MP and PSO, each method is repeatedly implemented 20 times. The optimal weights are shown as the box charts in Figure 2.

As can be seen from Figure 2, the variation in w_1 obtained by the PSO method is larger than that of the MP method. Moreover, the values of w_2 optimized by PSO are all 0 which will cause the feature to be ignored during model training. According to [10], the weights combination (0.01, 0.0001) searched by GS is effective for model training. The mean weight values (0.0143, 0.0001) and (0.0219, 0.0000) obtained by MP and PSO, respectively, are closer to the GS method than the MIC method.

Then, the MP algorithm is compared with MIC and PSO algorithms to observe the differences in the generalization ability. The near-average weights in Figure 2 of MP and PSO are selected. The results are shown in Table 2.

As shown in Table 2, the MP algorithm achieves the best generalization performance compared with the other two feature weighting methods. MIC can reveal the feature's relationship with output. However, it does not consider the amplitude of the features. The amplitude ratio of MIC is only 1.2862, far less than 116.2602 of MP algorithm, which means that the influence of Feature 2 with a low contribution to the output is not obviously weakened. PSO can obtain both weights and parameters. However, the weight is not constrained to a reasonable interval. In Table 2, w_2 is set to 0 which is obviously unreasonable.

In order to facilitate the comparison, the predicted outputs for the test set are shown in Figure 3.

Figure 3 indicates that the output curve of MP-FWSVR is almost coincident with the true function. The prediction curve of MIC-FWSVR is quite different from the real output. The PSO-FWSVR achieves better results than MIC-FWSVR, but underfitting occurs because Feature 2 is not involved in model training due to $w_2 = 0$.

Finally, we model four synthetic datasets in Table 3 to compare the proposed MP-FWSVR with GS-FWSVR [10], classical SVR, MIC-FWSVR [13], and PSO-FWSVR [14]. For classical SVR, the min-max normalization is employed to convert raw data to $[0, 1]$ by linearization. The k -th feature of the i -th sample x_{ik} is normalized to x'_{ik} :

$$x'_{ik} = \frac{x_{ik} - x_{k \min}}{x_{k \max} - x_{k \min}}, \quad (19)$$

where $x_{k \max}$, $x_{k \min}$ is the maximum and minimum value of the k -th feature, respectively. Each program is repeatedly implemented 10 times. The mean values and standard deviation of the RMSE are shown in Table 3.

According to Table 3, MP-FWSVR achieves a competitive generalization performance. Compared with the classical SVR, MIC-FWSVR, and PSO-FWSVR, MP-FWSVR achieves the best generalization performance on almost all four datasets. Compared with the GS-FWSVR, only one optimal result is obtained by MP-FWSVR. However, what needs to be noticed is that other three results are close to the GS-FWSVR ones in the case of much less training time consumption.

TABLE 1: Functions used for generating synthetic datasets.

Name	Function definition	Domain of definition
F1	$y_i = (\sin(x_{i1})/x_{i1}) + (x_{i2}/1000) + \sigma$	$x_{i1} \in [-10, 10], x_{i2} \in [-30, 30]$
F2	$y_i = 10 \cos(3x_{i1}) + (1/10)\sin(x_{i2}) + \sigma$	$x_{i1}, x_{i2} \in [-2\pi, 2\pi]$
F3	$y_i = x_{i1}^2 \sin(x_{i2}) + (3/1 + e^{-x_{i3}}) + \sigma$	$x_{i1} \in [-3, 3], x_{i2} \in [-2\pi, 2\pi], x_{i3} \in [-200, 200]$
F4	$y_i = (x_{i1}/100) + (x_{i2}/10000) + \sigma$	$x_{i1} \in [-10, 10], x_{i2} \in [-30, 30]$

FIGURE 1: The input feature data of (a) training dataset and (b) test dataset.

FIGURE 2: The optimal weights obtained by MP and PSO.

TABLE 2: Comparison of performance among MP, MIC, and PSO.

Method	w_1	w_2	Amplitude ratio of weighted feature 1 to 2	(C, γ, ε)	RMSE
MP	0.0143	0.000041	116.2602	(5.2130, 363.0250, 0.001)	0.0131
MIC	0.9496	0.2461	1.2862	(362.0387, 0.0313, 0.001)	0.1025
PSO	0.0257	0	—	(0.3415, 243.6, 0.001)	0.0305

4.2. *Experiments on Benchmark Datasets.* We randomly choose seven UCI benchmark datasets [19], and the results are compared as in Table 4.

According to Table 4, two optimal results and three suboptimal results are obtained by MP-FWSVR. Compared

with the MIC-FWSVR, MP-FWSVR achieves better results in 6 of the 7 datasets. Compared with the PSO-FWSVR, MP-FWSVR achieves better results in 5 of the 7 datasets.

To summarize, the Wilcoxon signed rank tests [20] at the 0.05 significance level are implemented to test the differences

FIGURE 3: The predicted outputs for test set.

TABLE 3: Performance comparison of SVR modeling for the synthetic datasets.

Function name	MP-FWSVR RMSE (mean $\pm$ std.) (training time (s))	GS-FWSVR RMSE (mean $\pm$ std.) (training time (s))	Classical SVR RMSE (mean $\pm$ std.) (training time (s))	MIC-FWSVR RMSE (mean $\pm$ std.) (training time (s))	PSO-FWSVR RMSE (mean $\pm$ std.) (training time (s))
F1	0.0109 $\pm$ 0.0032 128.5738	0.0095 $\pm$ 0.0030 222.5062	0.1006 $\pm$ 0.0195 9.2700	0.1029 $\pm$ 0.0175 15.5053	0.0285 $\pm$ 0.0059 53.0665
F2	0.0896 $\pm$ 0.0047 382.3847	0.0864 $\pm$ 0.0024 110.8701	5.9021 $\pm$ 0.0004 4.5508	8.2245 $\pm$ 0.0012 9.4366	0.0892 $\pm$ 0.0014 293.4736
F3	1.7280 $\pm$ 0.1384 116.2665	1.3450 $\pm$ 0.0095 627.0583	2.9849 $\pm$ 0.0066 5.1132	3.1479 $\pm$ 0.0071 5.9891	2.3882 $\pm$ 0.5042 15.2894
F4	0.0027 $\pm$ 0.0002 61.6933	0.0037 $\pm$ 0.0009 242.8880	0.0219 $\pm$ 0.0031 9.9771	0.0115 $\pm$ 0.0021 16.6370	0.0029 $\pm$ 0.0011 44.8777

TABLE 4: Performance comparison of SVR modeling for the UCI datasets.

Datasets (training size, test size)	MP-FWSVR RMSE (mean $\pm$ std.) (training time (s))	GS-FWSVR RMSE (mean $\pm$ std.) (training time (s))	Classical SVR RMSE (mean $\pm$ std.) (training time (s))	MIC-FWSVR RMSE (mean $\pm$ std.) (training time (s))	PSO-FWSVR RMSE (mean $\pm$ std.) (training time (s))
CCPP (300 $\times$ 4, 9268 $\times$ 4)	4.3690 $\pm$ 0.0164 165.5336	4.3739 $\pm$ 0.0000 2.2334e + 004	4.4532 $\pm$ 0.0000 37.3370	4.4220 $\pm$ 0.0000 38.9261	4.3844 $\pm$ 0.0326 100.2486
Concrete (300 $\times$ 8, 730 $\times$ 8)	10.3616 $\pm$ 1.2075 330.8808	8.2697 $\pm$ 0.0000 1.4203e + 007	11.3164 $\pm$ 0.0000 36.3742	17.0857 $\pm$ 0.0000 38.2225	13.9326 $\pm$ 3.2440 357.7611
Airfoil (376 $\times$ 5, 1127 $\times$ 5)	2.4946 $\pm$ 0.0443 691.7991	2.6239 $\pm$ 0.0000 1.7233e + 005	3.0588 $\pm$ 0.0000 56.0422	6.3633 $\pm$ 0.0000 58.9148	4.3604 $\pm$ 0.1586 184.5284
Auto-MPG (196 $\times$ 7, 196 $\times$ 7)	3.2494 $\pm$ 0.1922 61.9812	2.5555 $\pm$ 0.0000 1.1709e + 006	2.9386 $\pm$ 0.0000 14.8691	5.6032 $\pm$ 0.0000 14.8908	3.1244 $\pm$ 0.3231 147.5384
Machine CPU (109 $\times$ 7, 100 $\times$ 7)	39.0181 $\pm$ 5.8341 19.6631	20.9777 $\pm$ 0.0000 3.9267e + 005	48.7881 $\pm$ 0.0000 5.1938	156.0875 $\pm$ 0.0000 5.2505	59.4450 $\pm$ 15.2211 16.8121
Servo (84 $\times$ 4, 83 $\times$ 4)	1.1359 $\pm$ 0.0229 8.8402	1.0071 $\pm$ 0.0000 3.0122e + 003	1.2255 $\pm$ 0.0000 4.5598	1.2115 $\pm$ 0.0000 4.5116	1.1439 $\pm$ 0.0623 9.1346
Yacht (154 $\times$ 6, 154 $\times$ 6)	4.6284 $\pm$ 1.1031 1.5130e + 03	0.8789 $\pm$ 0.0000 6.4835e + 005	15.0084 $\pm$ 0.0000 41.4972	3.8010 $\pm$ 0.0000 35.5184	3.2324 $\pm$ 1.3273 1.3726e + 03

TABLE 5: Wilcoxon signed rank test for the prediction results.

Method	GS-FWSVR	Classical SVR	MIC-FWSVR	PSO-FWSVR
MP-FWSVR	0.0674	0.0098	0.0098	0.1016
GS-FWSVR	—	$9.7656e-04$	$9.7656e-04$	0.0020

between the MP-FWSVR, GS-FWSVR, and other SVR methods for all synthetic and UCI datasets to substantiate the indications in Tables 3 and 4. The test results are presented in Table 5.

The overall results obtained by the Wilcoxon tests presented in Table 5 show that there is significant difference among MP-FWSVR, classical SVR, and MIC-FWSVR. MP-FWSVR achieves the generalization ability which is more approximate to the GS-FWSVR than the PSO-FWSVR method when the time consumption is acceptable.

5. Conclusions

In the paper, we propose a feature-weighted SVR method named MP-FWSVR which integrates the advantages of MIC and PSO algorithms to solve the time-consumption problem of the GS-FWSVR algorithm proposed in our previous work. The contribution of each input feature is expressed as MIC which is used to constrain the value of feature weight in the PSO optimization of the weight combination. Numerical experiments show the effectiveness of the proposed algorithm. Our future work will continue to focus on the feature weighting algorithm to get better generalization ability.

Data Availability

In the paper, four synthetic datasets and seven real datasets are employed to evaluate the feasibility of the MP-FWSVR. We randomly chose seven UCI benchmark datasets from <http://archive.ics.uci.edu/ml/index.php>.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant nos. 61572514 and 61903049), Scientific Research Fund of Hunan Provincial Education Department (Grant nos. 20B057 and 19C0160), Changsha Natural Science Foundation (Grant no. kq2007088), and Open Research Fund of Hunan Provincial Key Laboratory of Network Investigational Technology (Grant no. 2020WLZC003).

References

- [1] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [2] M. Xie and L. Xie, "Decoupling control of permanent magnet synchronous motor with support vector regression inverse system method," *IEEE Access*, vol. 8, pp. 212687–212698, 2020.
- [3] V. Vapnik, "Principles of risk minimization for learning theory," *Advances in Neural Information Processing Systems*, vol. 4, pp. 831–838, 1992.
- [4] K. Cheng and Z. Lu, "Active learning bayesian support vector regression model for global approximation," *Information Sciences*, vol. 544, pp. 549–563, 2021.
- [5] J. Wei, G. Dong, and Z. Chen, "Remaining useful life prediction and state of health diagnosis for lithium-ion batteries using particle filter and support vector regression," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5634–5643, 2018.
- [6] Y. Yaslan and B. Bican, "Empirical mode decomposition based denoising method with support vector regression for time series prediction: a case study for electricity load forecasting," *Measurement*, vol. 103, pp. 52–61, 2017.
- [7] C. Hua, J. Wu, J. Li, and X. Guan, "Silicon content prediction and industrial analysis on blast furnace using support vector regression combined with clustering algorithms," *Neural Computing and Applications*, vol. 28, no. 12, pp. 4111–4121, 2017.
- [8] S. Chelgani, B. Shahbazi, and E. Hadavandi, "Support vector regression modeling of coal flotation based on variable importance measurements by mutual information method," *Measurement*, vol. 114, pp. 102–108, 2018.
- [9] J. Ruan, Q. Zhan, L. Tang, and K. Tang, "Real-time temperature estimation of three-core medium-voltage cable joint based on support vector regression," *Energies*, vol. 11, 2018.
- [10] M. Xie, D. Wang, and L. Xie, "A feature-weighted SVR method based on kernel space feature," *Algorithms*, vol. 11, 2018.
- [11] F. Xiao, Y. Wang, Y. Gao, Y. Zhu, and J. Zhao, "Continuous estimation of elbow joint angle by multiple features of surface electromyographic using grey features weighted support vector machine," *Journal of Medical Imaging and Health Informatics*, vol. 7, no. 3, pp. 574–583, 2017.
- [12] J. N. K. Liu and Y. Hu, "Application of feature-weighted support vector regression using grey correlation degree to stock price forecasting," *Neural Computing and Applications*, vol. 22, no. S1, pp. 143–152, 2013.
- [13] H. Hou, Y. Gao, and D. Liu, "A support vector machine with maximal information coefficient weighted kernel functions for regression," in *Proceedings of the 2014 2nd International Conference on Systems and Informatics*, pp. 938–942, Shanghai, China, 2014.
- [14] J.-H. Wen, K.-J. Zhong, L.-J. Tang et al., "Adaptive variable-weighted support vector machine as optimized by particle swarm optimization algorithm with application of QSAR studies," *Talanta*, vol. 84, no. 1, pp. 13–18, 2011.
- [15] Q. Yang, H.-Y. Zou, Y. Zhang et al., "Multiplex protein pattern unmixing using a non-linear variable-weighted support vector machine as optimized by a particle swarm optimization algorithm," *Talanta*, vol. 147, pp. 609–614, 2016.
- [16] H.-Y. Zou, H.-L. Wu, H.-Y. Fu et al., "Variable-weighted least-squares support vector machine for multivariate spectral analysis," *Talanta*, vol. 80, no. 5, pp. 1698–1701, 2010.
- [17] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 1, pp. 26–33, 1986.
- [18] C.-C. Chang and C.-J. Lin, "LIBSVM," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.
- [19] K. Bache and M. Lichman: UCI Machine Learning Repository, 2013.
- [20] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.