An Efficient Twin Projection Support Vector Machine for Regression

Xinyu Ouyang, Nannan Zhao, Chuang Gao, and Lidong Wang

Abstract—Taking motivation from \( \epsilon \)-insensitive twin support vector regression (\( \epsilon \)-TSVR) and the projection idea, this paper proposes a novel \( \epsilon \)-twin projection support vector regression models, called \( \epsilon \)-TPSVR. The proposed \( \epsilon \)-TPSVR, which is based on \( \epsilon \)-TSVR, determines the regression function through a pair of nonparallel hyperplanes solved by two smaller sized quadratic programming problems. Different from \( \epsilon \)-TSVR, a projection axis is sought for each optimization problem of \( \epsilon \)-TPSVR such that the variance of the projected points is minimized. Therefore, the empirical correlation coefficient between each hyperplane and the projected inputs can be optimized. The experimental results indicate that the proposed \( \epsilon \)-TSVR obtains the better prediction performance than TSVR and \( \epsilon \)-TSVR methods that were widely adopted.

Index Terms—Support vector machine (SVM), Regression analysis, Projection algorithms, Benchmark testing, \( \epsilon \)-twin support vector regression (\( \epsilon \)-TSVR)

I. INTRODUCTION

Support vector machine (SVM) [1–3] has become a powerful tool in pattern classification and regression field for its excellent generalization performance, and has been successfully applied to various real-world problems [4–6]. As for support vector regression model (SVR), which is a standard tool in regression tasks, there are some classical methods, such as \( \epsilon \)-support vector regression (\( \epsilon \)-SVR) [2], the sequential minimal optimization (SMO) [7], smooth SVR [8], the parametric insensitive SVR [9], PSO-SVR [10], and \( \nu \)-support vector classification (\( \nu \)-SVC) [11].

One of the main challenges for the SVR is the high cost of training, i.e. \( O(m^3) \), where \( m \) is the number of training samples. Motivated by the researches on twin support vector machine (TSVM) [12–15], Peng [16] proposed a twin support vector regression (TSVR), in which the complexity of algorithm is reduced to \( O(2 \times (m/2)^3) \), that means the TSVR is four times faster than the usual SVR in theory. Different from SVR, the TSVR method generates two nonparallel functions by solving a pair of smaller sized quadratic programming problems (QPPs) instead of a large size one, so the algorithm TSVR is much faster than a usual SVR. Consequently, many kinds of twin-type SVR have been studied extensively. For example, Shao et al. [17] proposed an \( \epsilon \)-insensitive twin support vector regression (\( \epsilon \)-TSVR), in which the structural risk minimization principle is implemented by introducing the regularization term in the primal problems. R. Rastogi et al. [18] presented another twin model for regression termed as \( \nu \)-TSVR, which extended the \( \epsilon \)-TSVR techniques by automatically adjusting the value \( \epsilon_1 \) and \( \epsilon_2 \) via user defined parameters \( v_1 \) and \( v_2 \), so that the fraction of errors and support vectors can be controlled.

However, the previous algorithms and models are very suitable for the assumption that the noise level is uniform on training data, or at least. For the heteroscedastic noise structure, in which the amount of noise depends on location or input, the assumption of a uniform noise level is not satisfied. Recently, motivated by the par-\( \nu \)-SVR [19], Peng [20] presented a novel parametric-insensitive SVR model for data regression, termed as twin parametric insensitive SVR (TPISVR). Although both par-\( \nu \)-SVR and TPISVR are more suitable for the case that the noise is more heteroscedastic than the classical SVR, the difference between them is that the strategy TPISVR is to solve two smaller sized QPPs rather than one large QPP, which makes it learn faster than par-\( \nu \)-SVR and SVR. Later, another efficient twin projection support vector regression (TPSVR) algorithm proposed by Peng [21] improved TPISVR through seeking a suitable projection axis such that the empirical variance of projected points is minimized.

In addition, it is interesting to note that, in the \( \epsilon \)-TSVR model, the regularization term \( \frac{w_k^T w_k + b_k^T}{2}, k = 1, 2 \) is added to the primal optimization problems because of the structural risk minimization principle. The regularization terms in the TPSVR model, however, is \( \frac{w_k^T w_k}{2}, k = 1, 2 \). Because minimizing the projection zone of input points is unrelated to the offset \( b_k \), \( b_k^* \) has to be removed from the regularization term in order to getting the solutions of optimization problems correctly. This implies that the method of solving \( b_k^* \) has to be deduced separately, which will increase the complexity of the model. In fact, what we expect is that the augmented vector \( u_k = [w_k^T b_k] \text{T} \) is directly solved, not directly obtained \( w_k \). To this end, inspired by \( \epsilon \)-TSVR and the idea of projection, an \( \epsilon \)-twin projection support vector machine for regression is proposed in this paper, which is termed as \( \epsilon \)-TPSVR. Some of the major features of the proposed \( \epsilon \)-TPSVR are as follows:
(1) The proposed ε-TPSVR model aims to seek a suitable projection axis such that the empirical variance of projected points is minimized. It is more suitable for the assumption that the noise level is heteroscedastic. In addition, because the propose method is based on the ε-TSVR model, the structural information of data is then embedded into the regression model.

(2) The Introduction of matrices J and K makes it possible to solve the augmented vector \( u_k = [w_k^T \ b_k^T]^T \). It also reduces computational complexity. For the definition of matrices J and K, see section III for details.

(3) In terms of generalization performance, the results of the experiment have indicated that the performance obtained by the proposed ε-TPSVR is superior to that of other classical twin-type SVR algorithms for the artificial dataset with the heteroscedastic error structure and UCI datasets.

The rest texts of paper are organized as follows. Section II introduces notations used in this paper and briefly describes ε-TSVR and the projection axis. Section III proposes the linear ε-TPSVR model and its extension for the non-linear case. Numerical experiments have been done on both synthetic and real-world datasets and their results have been compared with TSVR, ε-TSVR and ν-TSVR in Section IV, and section V contains concluding remarks.

II. BACKGROUND

In this section, we first give a brief description of ε-twin support vector regression (ε-TSVR) [17] and then introduce the concept of projection axis [21]. Without loss of generality, given a training set \( T = \{ (x_i, y_i) \} \), \( i = 1, 2, ..., l \), where \( x_i \in \mathbb{R}^n \) and \( y_i \in \mathbb{R} \). Also let \( A = (A_1; A_2; ...; A_l) \) be the input training sample, and \( Y = (A_1; y_2; ...; y_l) \) be the response of the training samples, where \( A \) is a \( l \times n \) matrix, the \( i \)-th row \( A_i \) represents the \( i \)-th training sample, \( Y \) is a \( 1 \times l \) vector, and \( y_i \) represents the \( i \)-th response.

A. ε-Twin Support Vector Regression

Following the idea of TWSVM and TSVR, Shao et al. [15] proposed an approach termed as ε-twin support vector regression (ε-TSVR). Just like any other twin-type SVR, it also finds two proximal insensitive function: \( f_k(x) = w_k^T x + b_k, \) \( k = 1,2 \). By introducing the regularization terms \( (w_k^T w_k + b_k^2)/2 \) and the slack variables \( \xi, \xi^*, \eta \) and \( \eta^* \), the primal problems can be expressed as

\[
\min_{\xi, \xi^*, \eta, \eta^*} \frac{1}{2} \xi^T \xi + \frac{1}{\varepsilon} \xi^T \xi^* + c_1 \varepsilon \xi^* \\
\text{s.t.} \quad Y \geq (A_1 w_1 + b_1), \quad \eta \geq 0,
\]

and

\[
\min_{\xi, \xi^*, \eta, \eta^*} \frac{1}{2} \xi^T \xi + \frac{1}{\varepsilon} \xi^T \xi^* + c_1 \varepsilon \xi^* \\
\text{s.t.} \quad (A_2 w_2 + b_2) - Y = \eta^*,
\]

\( w_1, w_2 \neq 0 \).

where \( c_1, c_2, c_3, c_4, \) \( \varepsilon_1 \) and \( \varepsilon_2 \) are positive parameters, \( e \) is column vector of ones of appropriate dimension. The main difference between TSVR and ε-TSVR is an extra regularization term \((w_k^T w_k + b_k^2)/2, k = 1,2, \) in (1) and (2), thus the structural risk minimization principle is implemented [17].

In order to get the solutions of problems (1) and (2), their dual problems need to be derived. By introducing the Lagrangian multiplies \( \alpha \) and \( \gamma \), the dual problems of the ε-TSVR are given by

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \alpha^T G \alpha + c_1 \varepsilon \alpha - \gamma^T (G \alpha + c_1 \varepsilon) \\
\text{s.t.} & \quad 0 \leq \alpha \leq c_1 \varepsilon, \\
\end{align*}
\]

and

\[
\begin{align*}
\min_{\gamma} & \quad \frac{1}{2} \gamma^T G \gamma + c_1 \varepsilon \gamma - \alpha^T (G \gamma + c_1 \varepsilon) \\
\text{s.t.} & \quad 0 \leq \gamma \leq c_2 \varepsilon, \\
\end{align*}
\]

where \( G = [A \quad e] \). Once the QPPs (3) and (4) are solved, \( u_1 \) and \( u_2 \) and the final regressor function \( f(x) \) can be obtained as follows

\[
\begin{align*}
u_1 &= \left[ w_1^T \right], \\
u_2 &= \left[ w_2^T \right], \\
f(x) &= \frac{1}{2} (f_1(x) + f_2(x)) = \frac{1}{2} (w_1 x + w_2 x) + \frac{1}{2} (b_1 + b_2).
\end{align*}
\]

B. Projection Axis

In the twin-type SVR, there is usually a pair of insensitive bound function \( f_k(x) = w_k^T x + b_k, k = 1,2 \) , i.e. the insensitive up- and down-bound functions by solving a pair of smaller sized QPPs. In order to obtain a suitable \( w_k \), the variance of \( w_k^T x + b_k - y_i \) are supposed to be as small as possible. In the other word, in each QPP, it finds a projection axis, denoted as \( w_k = [w_k^T -1]^T, \) \( k = 1,2, \) such that each points will be projected on the projection axis \( w_k \), and the projected zone is expected to be minimized. It’s important to note here that \( b_k \) does not contribute to the variance of \( w_k^T x_i + b_k - y_i \) because \( b_k \) is a fixed variable. The projected zone for each bound function is minimized as

\[
\begin{align*}
\min_{w_k} & \quad \frac{1}{2} \sum_{i=1}^{l} \left[ w_k^T x_i - (y_i - \bar{y}) \right]^2 \\
\text{s.t.} & \quad \bar{x}^T w_k - \bar{x}^T \bar{y} = 0,
\end{align*}
\]

(8)

where \( \Sigma_x, \Sigma_y \) are the empirical covariance matrices of inputs, and responses, and \( \Sigma_{xy} \) is the empirical covariance matrix of the inputs and responses. They are defined respectively as

\[
\begin{align*}
\Sigma_x &= \sum_{i=1}^{l} \left( x_i - \bar{x} \right) \left( x_i - \bar{x} \right)^T, \\
\Sigma_y &= \sum_{i=1}^{l} \left( y_i - \bar{y} \right) \left( y_i - \bar{y} \right)^T, \\
\Sigma_{xy} &= \sum_{i=1}^{l} \left( x_i - \bar{x} \right) \left( y_i - \bar{y} \right),
\end{align*}
\]

where \( \bar{x} \) and \( \bar{y} \) are the centroid points of inputs and responses, respectively.

III. ε-TWIN PROJECTION SUPPORT VECTOR REGRESSION

In this section, a novel ε-twin projection support vector machine for regression, termed as ε-TPSVR is presented, which is motivated by the ε-TSVR and the idea of projection. Using (8), the optimization problems of (1) and (2) can be rewritten as the following equations

\[
\begin{align*}
\min_{w_k, b_k, \xi, \eta, \eta^*} & \quad \frac{1}{2} \xi^T \xi + \frac{1}{\varepsilon} \xi^T \xi^* + c_1 \varepsilon \xi^* \\
\text{s.t.} & \quad Y \geq (A_1 w_1 + b_1), \quad \xi \geq 0,
\end{align*}
\]

(9)

\[
\begin{align*}
\min_{w_k, b_k, \xi, \eta, \eta^*} & \quad \frac{1}{2} \xi^T \xi + \frac{1}{\varepsilon} \xi^T \xi^* + c_1 \varepsilon \xi^* \\
\text{s.t.} & \quad (A_2 w_2 + b_2) - Y = \eta^*, \quad \eta \geq 0,
\end{align*}
\]

and

\[
\begin{align*}
\min_{w_k, b_k, \xi, \eta, \eta^*} & \quad \frac{1}{2} \xi^T \xi + \frac{1}{\varepsilon} \xi^T \xi^* + c_1 \varepsilon \xi^* \\
\text{s.t.} & \quad (A_2 w_2 + b_2) - Y = \eta^*, \quad \\
\end{align*}
\]

(10)

\[
\begin{align*}
\min_{w_k, b_k, \xi, \eta, \eta^*} & \quad \frac{1}{2} \xi^T \xi + \frac{1}{\varepsilon} \xi^T \xi^* + c_1 \varepsilon \xi^* \\
\text{s.t.} & \quad (A_2 w_2 + b_2) - Y \geq -c_2 \xi - \eta, \quad \eta \geq 0.
\end{align*}
\]
To solve (9), the Lagrangian function is given by
\[
\mathcal{L} = \frac{1}{2}c(w_1^T w_1 + b^T b) + \frac{1}{2}(Y - (Aw_1 + eb_1))^T(Y - (Aw_1 + eb_1)) + \epsilon^T \xi + \lambda_1^T (\frac{1}{2}w_1^T w_1 - \Sigma_{xy} w_1) - a^T(Y - (Aw_1 + eb_1) + \epsilon e + \xi) - \beta^T \xi,
\]
where \(a = (a_1, a_2, ..., a_\ell)\) and \(\beta = (\beta_1, \beta_2, ..., \beta_\ell)\) are the vectors of Lagrange multipliers. The K.T.T necessary and sufficient optimality conditions (9) are given by
\[
\begin{align*}
\left. \frac{\partial \mathcal{L}}{\partial a} \right|_{a*} &= 0, \\
\left. \frac{\partial \mathcal{L}}{\partial \beta} \right|_{\beta*} &= -e^T(Y - (Aw_1 + eb_1)) + c_\ell b_\ell + e^T a = 0, \\
\left. \frac{\partial \mathcal{L}}{\partial \xi} \right|_{\xi*} &= -a - \beta = 0, \\
\left. \frac{\partial \mathcal{L}}{\partial \epsilon} \right|_{\epsilon*} &= -e^T(Y - (Aw_1 + eb_1)) + c_\ell b_\ell + e^T a = 0, \\
\left. \frac{\partial \mathcal{L}}{\partial \lambda_1} \right|_{\lambda_1*} &= a^T(Y - (Aw_1 + eb_1) + \epsilon e + \xi) - \beta^T \xi = 0, \quad \beta \geq 0.
\end{align*}
\]
(11)

As \(\beta \geq 0\), from (14), we have \(0 \leq \alpha \leq c_\ell e\). Next, combing (12) and (13) leads to
\[
\begin{align*}
-G = I_{\epsilon}, \quad u_1 = \begin{bmatrix} w_1^T \\ b_1 \end{bmatrix}, \\
F = I_{\epsilon}, \quad u_2 = \begin{bmatrix} 0 \\ w_2^T \end{bmatrix}, \quad K = \begin{bmatrix} \Sigma_{xy} \\ 0 \end{bmatrix}.
\end{align*}
\]
(19)
equation (18) can be rewritten as
\[
-G^T(Y - G u_1) + c_\ell u_1 + G^T a + \lambda_1 f u_1 - \lambda_1 K = 0.
\]
(20)

Then the augmented vector \(u_1\) is given by \(u_1 = (G^T G + c_\ell I + \lambda_1 J)^{-1}(G^T y - G^T a + \lambda_1 K)\).
(21)

Then putting (21) into (11) and using the above K.T.T. dual problem, the dual problem of the problem (9) is obtained as
\[
\begin{align*}
\max_{a, \beta, e, \xi} &= -a^T D_1 a + a^T D_2 Y + \lambda_1 a^T (G^T G + c_\ell I + \lambda_1 J)^{-1} K - a^T f, \\
\text{s.t.} &= 0 \leq \alpha \leq c_\ell e, \\
D_1 &= (G^T G + c_\ell I + \lambda_1 J)^{-1} G^T, \quad D_2 = (G^T G + c_\ell I + \lambda_1 J)^{-1} K.
\end{align*}
\]
(22)

The augmented vector \(u_2\) is given by
\[
u_2 = \begin{bmatrix} w_2^T \\ b_2 \end{bmatrix} = (G^T G + c_\ell I + \lambda_1 J)^{-1}(G^T y + Y f^2).
\]
(24)

Once the solutions \(w_1, w_2, b_1\) and \(b_2\) are obtained, the estimated regressor can be constructed as (7). For the case of nonlinear regressors, an appropriately chosen kernel function \(K\) should be introduced, and the more detailed derivation is similar to the references [17, 21].

IV. EXPERIMENTS

To check the performance of the proposed \(\varepsilon\)-TPSVR, we compare it with the popular TSVR, \(\epsilon\)-TSVR and \(\nu\)-TSVR in several artificial and benchmark [22] datasets. All the computations are carried out on Windows 7 OS Intel Core i5-4210U CPU(2.4GHz) with 4GB RAM and MATLAB R2014a environment. In order to decrease the computational complexity of parameter selection, we set \(c_1 = c_2 = c_3 = c_\epsilon\), \(\epsilon_1 = \epsilon_2\) and \(\lambda_1 = \lambda_2\). And Gaussian kernel function defined by \(K(a, b) = \exp(-||a - b||^2/\rho)\) is used to process nonlinear data, where vectors \(a, b \in \mathbb{R}^n\), and the parameter \(\rho \geq 0\). Some commonly-used evaluation criterions [16] shown in Table I are introduced before evaluating the performance of these methods.

To test the regression performance of our proposed \(\varepsilon\)-TPSVR, \(f_1(x) = \sin(x)/x, x \in [-4\pi, 4\pi]\) and \(f_2(x) = |x|^{1/3}, x \in [-4\pi, 4\pi]\) are introduced to generate all artificial datasets. The training data's observed values are polluted by the form \(y = f(x) + err\), where the noise \(err = (0.5 - |x|/8\pi)e\) depends on input. Variable \(e\) is the form of \(u[a, b] \text{ or } N(\mu, \sigma^2)\), where \(u[a, b]\) represents the uniformly random variable in \([a, b]\) and \(N(\mu, \sigma^2)\) represents the Gaussian random variable with means \(\mu\) and variance \(\sigma^2\). It implies that these examples have the heteroscedastic error structure.

<p>| TABLE I |
| <strong>PERFORMANCE METRICS AND THEIR CALCULATION</strong> |</p>
<table>
<thead>
<tr>
<th>Metric</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE</td>
<td>(\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2)</td>
</tr>
<tr>
<td>SSR</td>
<td>(\Sigma_{i=1}^{n}(\hat{y}_i - \bar{y})^2)</td>
</tr>
<tr>
<td>SST</td>
<td>(\Sigma_{i=1}^{n}(y_i - \bar{y})^2)</td>
</tr>
<tr>
<td>SSE/SST</td>
<td>(\Sigma_{i=1}^{n}(y_i - \hat{y}<em>i)^2/\Sigma</em>{i=1}^{n}(y_i - \bar{y})^2)</td>
</tr>
<tr>
<td>SSR/SST</td>
<td>(\Sigma_{i=1}^{n}(\hat{y}<em>i - \bar{y})^2/\Sigma</em>{i=1}^{n}(y_i - \bar{y})^2)</td>
</tr>
</tbody>
</table>

Fig. 1 shows the one run results of \(\varepsilon\)-TPSVR and our \(\varepsilon\)-TPSVR on \(f_1(x)\) with Gaussian noise with mean zero and standard deviation 0.2, and Fig. 2 makes the same comparison with \(f_2(x)\). In Fig. 1 and Fig. 2, the solid line is the noisefree test data, and heteroscedastic noise is added to the test data as training data, which is the cross in the figures. It can be seen from the figures that the heteroscedastic noise different from usual additive noise has the characteristic of depending on location or input, so it’s harder to deal with. 260 noisy data are selected as training data, 500 data are selected as test data without noise, and the final regression function is trained by \(\varepsilon\)-TSVR and our \(\varepsilon\)-TPSVR. Horizontal coordinate represents the number of training points. In particular, in order to see the regression effect more clearly, the horizontal coordinates only show the range from 50 to 450 in Fig. 1, and the horizontal coordinates only show the range from 200 to 300 in Fig. 2. One of the dotted lines is the regression result of the proposed \(\varepsilon\)-TPSVR method, and the other dotted line is the regression result of the \(\varepsilon\)-TSVR method. The SSE values of \(\varepsilon\)-TSVR and \(\varepsilon\)-TPSVR algorithms in Fig. 1 are 0.4276 and 0.3674 in order, and the SSE values of them in Fig. 2 are 1.5449 and 1.0983 in order. It can be clearly observed from the figures that the result of \(\varepsilon\)-TPSVR is closer to the black solid line than the result of \(\varepsilon\)-TSVR. It is because that some points in the heteroscedastic zone are discarded, whereas \(\varepsilon\)-TSVR cannot filter out the possible noise points. It is also for this reason that data structure information is embedded in our \(\varepsilon\)-TPSVR learning process.

Next, the effectiveness of the proposed \(\varepsilon\)-TPSVR is further verified by comparing it with TSVR, \(\epsilon\)-TSVR and \(\nu\)-TSVR. To fairly compare with the performance of TSVR, \(\epsilon\)-TSVR, \(\nu\)-TSVR and \(\varepsilon\)-TPSVR, 20 independent groups’ data on the two functions with different types of noise are generated randomly using Matlab toolbox, including 260 samples during training and 500 samples during testing for each function. Besides, testing data points are uniformly sampled from the objective function without any noise. The measure results are listed in Table II and Table III. It is easy to see that our method obtains the smaller SSE values and SSE/SST values than the other methods on these two problems, which indicates that the proposed \(\varepsilon\)-TPSVR gets better performance.
than the other algorithms and it is more suitable for the case that the noise is heteroscedastic.

![Fig. 1 Predictions of ε-TSVR and ε-TPSVR on f₁(x) with err=N(0, 0.2²).](image1)

![Fig. 2 Predictions of ε-TSVR and ε-TPSVR on f₂(x) with err=N(0, 0.2²).](image2)

In addition, we test three UCI datasets: Servo, Wisconsin breast cancer datasets, Auto-Mpg. Concrete compressive strength, which are usually used in testing machine learning algorithms. To avoid biased comparisons, the standard ten-fold cross-validation is used to compute the optimal values. The results of the performance criteria are listed in Table IV. Also, it can be seen from Table IV that our method outperforms the other methods.

<table>
<thead>
<tr>
<th>DataSets</th>
<th>Regressor</th>
<th>SSE</th>
<th>SRE/SST</th>
<th>SSR/SST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε-TSVR</td>
<td>0.1296±</td>
<td>0.0020±</td>
<td>1.0330±</td>
<td></td>
</tr>
<tr>
<td>ε-TPSVR</td>
<td>0.0198±</td>
<td>0.0003±</td>
<td>0.1024±</td>
<td></td>
</tr>
<tr>
<td>ε-TSVR</td>
<td>0.1240±</td>
<td>0.0019±</td>
<td>1.0329±</td>
<td></td>
</tr>
<tr>
<td>ε-TPSVR</td>
<td>0.1198±</td>
<td>0.0019±</td>
<td>0.9860±</td>
<td></td>
</tr>
<tr>
<td>ε-TSVR</td>
<td>0.0236±</td>
<td>0.0004±</td>
<td>0.1117±</td>
<td></td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, a novel ε-twin projection support vector regression models (ε-TPSVR) is proposed. The idea of proposed ε-TPSVR is based on ε-TSVR formulation. However, because the term of projection axis is introduced to maximize the empirical correlation coefficient between the up-or down-bound targets and the projected inputs, our proposed ε-TPSVR derives better approximate than other ε-TSVR algorithm. The experimental results on several artificial and UCI datasets show that our proposed method gives similar or better generalization performance with TSVR, ε-TSVR and ε-TSVR. Moreover, how to select the optimal hyper parameters and how to determine them rapidly are the difficult problems and should be studied in the following work.

### REFERENCES


Advance online publication: 1 February 2019


XINYU OUYANG was born in Hunan Province, China, in 1974. He received the B.S. and M.S. degrees from Northeastern University and University of Science and Technology Liaoning, China in 1998 and 2003 respectively, and he received Ph.D. degree in intelligent control and expert system from Northeastern University in 2006. Since 1998 he has been with University of Science and Technology Liaoning, and he is currently a Professor of Automatic Control. He published over 30 scientific papers in diverse fields, some of which were indexed by SCI and EI. His research interests include artificial intelligence, machine learning and digital image processing.

NANNAN ZHAO was born in Liaoning Province, China, in 1975, received the B.S. and M.S. degrees from Northeastern University and University of Science and Technology Liaoning, China in 1998 and 2003 respectively, and she received Ph.D. degree in intelligent control and expert system from Northeastern University in 2006. She is currently an Associate Professor in the Communication Engineering with University of Science and Technology Liaoning. She has authored over 20 research papers. Her research interests include pattern recognition, machine learning and image processing.

CHUANG GAO was born in Liaoning Province, China, in 1982, received the B.S. and M.S. degrees from Warwick University and King’s College London, UK in 2005 and 2007 respectively. He is currently a Ph.D. candidate in University of Science and Technology Liaoning, China. He has authored over 10 research papers indexed by SCI and EI. His research interests include nonlinear system control, machine learning and intelligent control.

LIDONG WANG was born in Liaoning province, China in 1971, received the B.S. and M.S. degrees from Harbin Institute of Technology and University of Science and Technology Liaoning, China in 1993 and 2004 respectively, and he received Ph.D. degree from Gyeongsang National University, Korea in 2008. Since 1999 he has been with University of Science and Technology Liaoning, and he is currently an associate professor in Communication Engineering with University of Science and Technology Liaoning. He has authored over 30 research papers. His research interests include pattern recognition, machine learning and digital signal processing.

(Advance online publication: 1 February 2019)