Development Of An Adaptive Soft Sensor Based On FCMILSSVR

Ebrahim Gomnam, Hooshang Jazayeri-rad

Abstract- Facing with dynamic environment of industrial plants involves us design soft sensors capable of online learning. To response this requirement, an adaptive soft sensor based on a combination of Least Square Support Vector Regression (LSSVR) with Fuzzy C-Means (FCM) clustering is proposed in this paper. In this approach, first the samples are divided into several partitions. Consequently, for each partition we develop a local model using a new formulation of LSSVR which enables incremental learning. The proposed method is implemented on a chemical plant and compared with the online Support Vector Regression (SVR) algorithm. Simulation results indicate that the proposed method improves the generalization ability of soft sensor and the computation time decreases to a large extent in comparison to the online SVR.

Keywords- soft sensor, incremental least square support vector regression, fuzzy c-means clustering, data mining

1 Introduction

A soft sensor is a model that uses software techniques to estimate the value of a quality variable with the aim of process monitoring and control. At a general level, one can distinguish two different classes of soft sensors, namely model-driven and data-driven soft sensors. If a first principle model (FPM) describes the process sufficiently accurately, a model-driven soft sensor can be developed. However, a soft sensor based on detailed FPM is computationally intensive and in most of times a full process description cannot be achieved for real-time applications. As a solution to these issues, data-driven soft sensors were proposed and gained increasing popularity in the process industry. Modern measurement methods permit a large amount of operating data to be acquired, stored and analyzed, thereby enabling data-driven soft sensor development a feasible alternative [1]. There are a large variety of data-driven methods including but not limited to Principal Component Analysis (PCA), Partial Least Square (PLS), Artificial Neural Network (ANN), Genetic Algorithm (GA), Neuro-Fuzzy Systems (NFS) and Support Vector Machine (SVM). Hybrid methods are also popular in the field of data-driven soft sensor design. Among these methods SVM has gained a high popularity within the past few years due to its benefits over other methods. As an example, compared to ANN, not being trapped in local minima [2], less dependence on quantity and quality of training data and better generalization ability [3] are some of its benefits. SVM, introduced by Vapnik, in its early form was designed for classification problems and later applied to regression.

- Ebrahim Gomnam is currently pursuing masters degree program in Instrumentation and Automation engineering in Petroleum University of Technology, Ahwaz, Iran. E-mail: e.gomnam@ahwaz.put.ac.ir
- Hooshang Jazayeri-rad is Assistant Prof. in Instrumentation and Automation engineering department in Petroleum University of Technology, Ahwaz, Iran.

Soft sensor employs the regression capability of SVM. For SVM we must solve a convex optimization problem with inequality constraints which imposes a high computational burden. This disadvantage has been overcome by the Least Squares Support Vector Machines (LSSVM) which solves linear equations instead of a quadratic programming problem [4]. In this paper, we propose an incremental version of LSSVR[1] (ILSSVR) which is capable of dealing with the dynamic characteristics of the real world industrial plants. To improve its generalization ability and computation time we first cluster the data. Fuzzy C-Means (FCM) is chosen as the clustering method because it is a powerful clustering algorithm [5] and it can be claimed that it is the most widely used fuzzy clustering algorithm in practice [6]. The rest of paper is organized as follows: In section 2 a full description of the method is proposed. The results of implementing the method on a simulated chemical plant together with a short description of the plant are presented in section 3. In this section our method is compared with the online SVR described in [7] and ILSSVR. The paper is terminated with the conclusion in section 4.

2 Methodology

2.1 Least Square Support Vector Regression

The training data set is given as $\{(x_i, y_i)\}_{i=1}^{N}$ $where\ x_i \in R^m$ is the input observation vector and $y_i \in R$ is its corresponding output observation. As demonstrated in [8], for LSSVR we deal with the following optimization problem:

$$\min\ J(w,e) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{N} e_i^2 \qquad \textbf{(1)}$$

$$s.t. \qquad y_i = w^T\varphi(x_i) + b + e_i \qquad i = 1, \dots, N$$

where w relates to the flatness of the model [9], b is the bias, e_i represents the training errors, and C is the regularization parameter adjusting the tradeoff between the model flatness ($\|w\|^2$) and the model errors (e_i). For nonlinear models the data is mapped from the input space into a high-dimensional feature space employing the mapping $\varphi(.)$. The optimization problem (1) is solved in its dual form using the Lagrange multipliers. The Lagrange function is:

$$L(w,b,e,\alpha)=\frac{1}{2}\|w\|^2+\frac{C}{2}\sum_{i=1}^{N}e_i^2+\sum_{i=1}^{N}\alpha_i(w^T\varphi(x_i)+b+e_i-y_i) \quad \textbf{(2)}$$

where α_i is the Lagrange multiplier and is also called the dual variable. According to the duality theory, $L(w,b,e,\alpha)$ has a saddle point with respect to the primal and dual variables at the solution [10]. This means that:

$$\frac{\partial L}{\partial w}=0 \;\Rightarrow\; w=\sum_{i=1}^{N}\alpha_i\varphi(x_i)$$
$$\frac{\partial L}{\partial b}=0 \;\Rightarrow\; \sum_{i=1}^{N}\alpha_i=0 \quad \textbf{(3)}$$
$$\frac{\partial L}{\partial e_i}=0 \;\Rightarrow\; \alpha_i=Ce_i$$
$$\frac{\partial L}{\partial \alpha_i}=0 \;\Rightarrow\; w^T\varphi(x_i)+b+e_i-y_i=0$$

The above relations can be summarized in the matrix form as:

$$\begin{bmatrix}0 & \vec{\mathbf{1}}^T\\ \vec{\mathbf{1}} & Q+C^{-1}I_N\end{bmatrix}\begin{bmatrix}b\\ \vec{\alpha}\end{bmatrix}=\begin{bmatrix}0\\ \vec{y}\end{bmatrix} \quad \textbf{(4)}$$

where $\vec{\mathbf{1}}=[1_1,1_2,\dots,1_N]^T$, $\vec{\alpha}=[\alpha_1,\alpha_2,\dots,\alpha_N]^T$, $\vec{\mathbf{y}}=[y_1,y_2,\dots,y_N]^T$, $Q_{ij}=K(x_i,x_j)=\varphi(x_i)\varphi(x_j)$ is the kernel function, and I_N is the identity matrix of the size K.

After solving (4) we have:

$$\begin{bmatrix}b\\ \vec{\alpha}\end{bmatrix}=A^{-1}\begin{bmatrix}0\\ \vec{y}\end{bmatrix} \quad with \quad A=\begin{bmatrix}0 & \vec{\mathbf{1}}^T\\ \vec{\mathbf{1}} & Q+C^{-1}I_N\end{bmatrix} \quad \textbf{(5)}$$

The regression model is then constructed as:

$$f(x)=w^T\varphi(x)+b=\sum_{i=1}^{N}\alpha_iK(x_i,x)+b \quad \textbf{(6)}$$

Now we face with two problems. Firstly, this algorithm is offline which is not efficient for real cases due to the changes occurring in the operation condition of industrial plants. The second one is that calculating A^{-1} may be time consuming and imprecise especially when the size of A (number of samples) is large. The next section explains the incremental approach for LSSVR that resolves these two challenges.

2.2 Incremental Least Square Support Vector Regression (ILSSVR)

A new sample (x_{new},y_{new}) can be added to the model described by equation (4) as:

$$\begin{bmatrix}0 & \vec{\mathbf{1}}^T & 1\\ \vec{\mathbf{1}} & Q+C^{-1}I_N & \overrightarrow{p1}\\ 1 & \overrightarrow{p1}^T & p2\end{bmatrix}\begin{bmatrix}b\\ \vec{\alpha}\\ \alpha_{new}\end{bmatrix}=\begin{bmatrix}0\\ \vec{y}\\ y_{new}\end{bmatrix} \quad \textbf{(7)}$$

where $\overrightarrow{p1}=[K(x_1,x_{new}),K(x_2,x_{new}),\dots,K(x_N,x_{new})]^T$ and $p2=K(x_{new},x_{new})+C^{-1}$.

Now for pedagogical reasons we reformulate the new A matrix (A_2) as:

$$A_2=\begin{bmatrix}A_1 & \vec{u}\\ \vec{u}^T & a\end{bmatrix} \quad \textbf{(8)}$$

where $A_1=\begin{bmatrix}0 & \vec{\mathbf{1}}^T\\ \vec{\mathbf{1}} & Q+C^{-1}I_N\end{bmatrix}$, $\vec{u}=\begin{bmatrix}1\\ \overrightarrow{p1}\end{bmatrix}$ and $a=p2$ is a scalar quantity. As demonstrated in [11] we can calculate ${A_2}^{-1}$ using the Schur complements as follow:

$${A_2}^{-1}=\begin{bmatrix}{A_1}^{-1}+{A_1}^{-1}\vec{u}S^{-1}\vec{u}^T{A_1}^{-1} & -{A_1}^{-1}\vec{u}S^{-1}\\ -S^{-1}\vec{u}^T{A_1}^{-1} & S^{-1}\end{bmatrix} \quad \textbf{(9)}$$

or

$${A_2}^{-1}=\begin{bmatrix}T^{-1} & -T^{-1}\vec{u}a^{-1}\\ -a^{-1}\vec{u}^TT^{-1} & a^{-1}+a^{-1}\vec{u}^TT^{-1}\vec{u}a^{-1}\end{bmatrix} \quad \textbf{(10)}$$

where $S=a-\vec{u}^T{A_1}^{-1}\vec{u}$ and $T=A_1-\vec{u}a^{-1}\vec{u}^T$ are called the Schur complements of A_1 and a, respectively. As it is obvious S is scalar so calculating S^{-1} is easy but T is a matrix with the same size as the size of A_1 which causes some difficulties when calculating T^{-1}. In order to overcome this challenge we use the Sherman-Morrison formula [11]:

$$T^{-1}=(A_1-\vec{u}a^{-1}\vec{u}^T)^{-1}={A_1}^{-1}+\frac{{A_1}^{-1}\vec{u}\vec{u}^T{A_1}^{-1}}{a-\vec{u}^T{A_1}^{-1}\vec{u}} \quad \textbf{(11)}$$

So our ILSSVR algorithm will be as follow:
1. Get the first sample (x_1,y_1) and construct *A*.
2. Calculate A^{-1} directly and then α_1 and *b* using (5).
3. Set $A_1=A$ and then get the new sample.
4. Construct A_2 using (8) and then calculate ${A_2}^{-1}$ employing (9) or (10).
5. Compute the new values of α_i and *b* using (5).
6. Set $A_1=A_2$ and ${A_1}^{-1}={A_2}^{-1}$, get the new sample and then go to step 4.

As it is obvious, this procedure is incremental. In addition, the procedure enables us to avoid the direct computation of A^{-1}. However, to improve performance of the algorithm we cluster the data (using FCM) before applying to the ILSSVR algorithm.

2.3 Fuzzy C-Means Clustering

FCM is one of the renowned clustering methods developed by Dunn [12] and further improved by Bezdek [13]. It is based on minimization of the following objective function [14]:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{R} (u_{ij})^m \|x_i - c_j\|^2 \quad \textbf{(12)}$$

where N and R are the number of samples and clusters respectively, x_i is the ith sample, c_j is the center of the jth cluster, m is the weighting exponent determining the fuzziness degree and should satisfy $m \in (1, \infty)$, u_{ij} is the degree of membership of x_i to the cluster j which must satisfy $\sum_{j=1}^{R} u_{ij} = 1$ and $\|*\|$ is any norm expressing the similarity between any sample and the cluster center (Euclidian norm is used in this paper). The fuzzy clustering is carried out through an iterative optimization of the objective function (12), with the update of membership u_{ij} and the cluster centers c_j are given by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{R} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad \textbf{(13)}$$

$$c_j = \frac{\sum_{i=1}^{N} {u_{ij}}^m . x_i}{\sum_{i=1}^{N} {u_{ij}}^m} \quad \textbf{(14)}$$

This algorithm is composed of the following steps:

1. Initialize U=[u_{ij}] matrix.
2. At the step k calculate the centers using (14).
3. Update $U^{(k)}$ according to (13).
4. If $\|U^{(k)} - U^{(k-1)}\| < stopping\ criteion$ then stop; otherwise go to step 2.

In this algorithm the number of clusters R should be predefined. The number of clusters plays an important role in the performance of FCM. Wang and Zhang [15] have a good review on different indices for determining the number of clusters. In our work an index called Exponential Compactness And Separation (ECAS) proposed by Fazel Zarandi et al. [16] is used due to its superior performance against the other well-known indices. ECAS considers two measures: compactness and separation. Compactness shows closeness of elements in a cluster. It is quantified by the formula:

$$EC_{comp}(R) = \sum_{i=1}^{N} \sum_{j=1}^{R} u_{ij}^m \exp\left(-\left(\frac{\|x_i - c_j\|^2}{\beta_{comp}} + \frac{1}{R+1} \right) \right) \quad \textbf{(15)}$$

where $\beta_{comp} = \frac{\sum_{k=1}^{N} \|x_k - \bar{v}\|^2}{n(j)}$ with $\bar{v} = \frac{\sum_{i=1}^{N} x_i}{N}$ and $n(j)$ is number of samples in the cluster j. A large value for $EC_{comp}(R)$ shows a compact partitioning. Separation indicates how distinct the fuzzy clusters are from each other. This measure is defined as follow:

$$ES_{sep}(R) = \sum_{j=1}^{R} \exp\left(-\min_{k \neq j} \left\{ \frac{(R-1)\|c_j - c_k\|^2}{\beta_{sep}} \right\} \right) \quad \textbf{(16)}$$

where $\beta_{sep} = \frac{\sum_{k=1}^{R} \|c_k - \bar{v}\|^2}{R}$. A small ES_{sep} is an indication of a well-separated clustering. Finally these two measures are normalized and combined in the following way to construct the ECAS index defined as:

$$ECAS(R) = \frac{EC_{comp}(R)}{\max_R \left(EC_{comp}(R) \right)} - \frac{ES_{sep}(R)}{\max_R \left(ES_{sep}(R) \right)} \quad \textbf{(17)}$$

The larger the value of $ECAS(R)$, the better the clustering is performed. So:

$$optimum\ number\ of\ clusters = \arg\left(\max_R \{ECAS(R)\}_2^{R_{max}} \right) \quad \textbf{(18)}$$

3 Case study

In this paper, a model of a highly nonlinear chemical plant consisting of two cascade Continuous Stirred Tank Reactors (CSTRs) followed by a nonadiabatic flash separator with a recycle has been selected [17]. A simple diagram of the plant is shown in Figure 1:

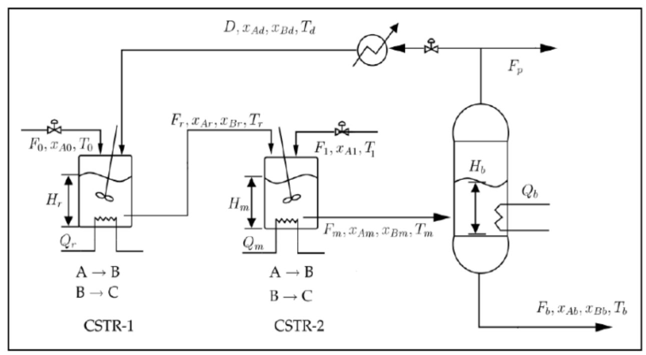

Figure 1. Two CSTRs plus flash separator [17]

This combination is prevalent in chemical industries such as the styrene polymerization process. The desired product B is generated by an irreversible first order reaction $A \xrightarrow{k_1} B$. An undesirable side reaction $B \xrightarrow{k_2} C$ occurs and leads to the consumption of B and the production of the superfluous side product C. The product stream from CSTR-2 is directed into a flash to separate the excess A from the product B and the side product C. Reactant A has the highest relative volatility and is the major component in the vapor phase. A fraction of the vapor phase is purged and the residual stream is condensed and recycled back to CSTR-1. The liquid phase exiting from the flash consists predominantly of B and C. The physical model and the parameters of the plant are described in [17]. The inputs of soft sensor are: D, F_0, F_1, H_b, H_m, H_r, Q_b, Q_m, Q_r, T_0, T_b, T_d, T_m, T_r which can easily be measured and the output is the product mass fraction x_{Bb}. In order to examine the capability of soft sensor to track the process changes, we define some realistic scenarios of change on the plant as follow:

- A random change in Q_m, Q_r, T_0, T_1, D, F_0 ,F_1 with the amplitude of about 20% of their operating point values.
- A step change in the concentration of the input feed. At first it is assumed that the input feed is pure A. After the step change the input feed contains both A and B with fractions of 90% and 10% , respectively.

We generate 400 samples from this model. 100 data points are used for training process and 300 data points are used as the test samples. In the testing process when an incremental sample arrives, first it should be determined to which cluster it belongs to. In order to do this, we compute the membership function of this data to each cluster and choose the cluster with the highest membership value. Then the new sample is added to the model of this cluster via ILSSVR. The RBF kernel, which is the most prevalent kernel in practice [18], is used for the soft sensor proposed in this work because in comparison to other kernels it performs better and has less numerical difficulties [19]. C and the kernel parameter is determined using the ten-fold cross validation on the training data. The best value for C and kernel parameter are shown in Table 1. The number of clusters determined by the ECAS index is 3. For FCM the value of m (the weighting exponent) is 2, the stopping criterion is 0.001 and the maximum number of iterations is 100. We compare the result of our code with the online SVR codes of Parrella [7], with an ε amount of 0.005, and also with ILSSVR (our method but without clustering).

Table 1. Values of parameters of the algorithms

	Online SVR	**ILSSVR**	**FCMILSSVR**		
			Cluster 1	Cluster 2	Cluster 3
C	*2048*	*1024*	*8192*	*4096*	*8192*
Kernel parameter	*0.25*	*0.25*	*0.0625*	*0.5*	*4*

Figure 2 illustrate the prediction performance of these algorithms:

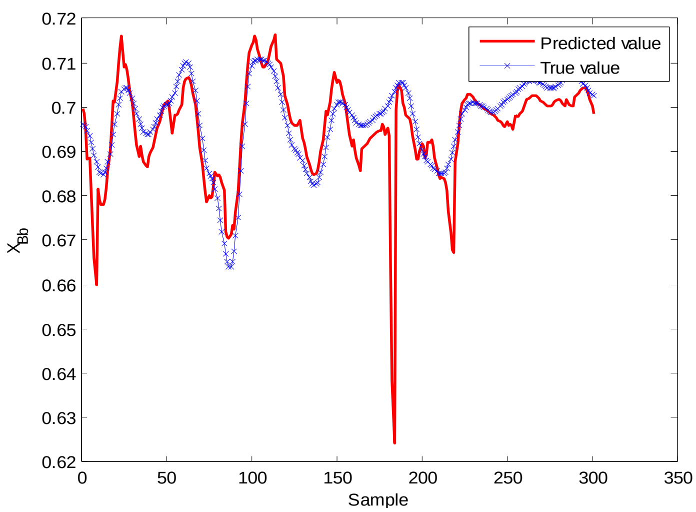

Figure 2(a). Prediction performance of the online SVR algorithm

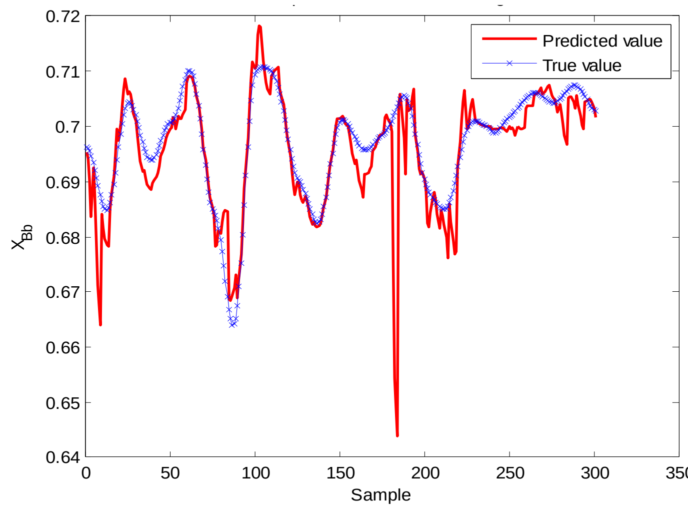

Figure 2(b). Prediction performance of the ILSSVR algorithm

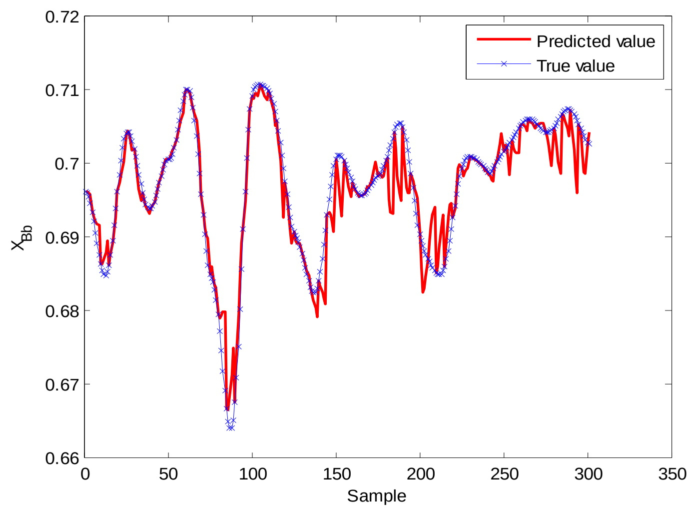

Figure 2(c).Prediction performance of the FCMILSSVR algorithm

The RMS (root mean squared) of training and test errors, the correlation coefficient and the training time for ISVR, ILSSVR and FCMILSSVR are shown in Table 2.

Table 2. Comparison of different online learning SVR approaches

	Online SVR	**ILSSVR**	**FCMILSSVR**
Training RMSE	*0.0040*	*7.18e-004*	*9.43e-005*
Test RMSE	*0.0086*	*0.0062*	*0.0030*
correlation coefficient	*0.6882*	*0.8246*	*0.9498*
Computation time(s)	*85*	*2.17*	*1.34*

It is observed that the online SVR method takes too much time to respond. The main reason behind this is the stabilization process – a process that corrects the errors generated by the floating-point operations [7]. The other point that should be noted is that we can improve the prediction ability of online SVR by reducing the amount of ε which increases the computation time. This is not favorable for real industrial plants. Both Figure 2 and Table 2 confirm the superior performance of FCMILSSVR against the other two algorithms. Comparing FCMILSSVR with ILSSVR we can conclude that clustering has two benefits: one is that it can improve the prediction ability because several local models perform better than one general model. The other benefit is the reduction of training time. The reason is that by partitioning the data into groups, the matrix A is also divided into several smaller sub-matrices. So, the overall calculations on these sub-matrices are performed faster than the computation of the original matrix.

4 Conclusion

In this paper we proposed a new formulation for LSSVR that enables it for incremental learning (which is an important property of a soft sensor in industrial environments). Before applying the algorithm to our data, the samples are partitioned into smaller groups. Because LSSVR involves matrix calculations and a reduction of the matrix size decreases the computation time and enhances the model accuracy. Partitioning of data is performed by utilization of the FCM technique. The optimum number of clusters is selected according to the ECAS index which considers both separation and compactness measures. The ten-fold cross validation is used as the parameter selection method. The soft sensor is applied to a chemical plant with some disturbance scenarios that may happen in real world cases. The results show that the proposed algorithm has an acceptable performance and can quickly track the process changes.

5 References

[1]. B. Lin, B. Recke, J. K. Knudsen and S. B. Jørgensen, "A systematic approach for soft sensor development," Computers and Chemical Engineering, vol. 31, no. 5-6, pp. 419-425, 2007.

[2]. W. Yan, H. Shao and X. Wang, "Soft Sensing Modeling Based on Support Vector Machine and Bayesian Model Selection," Computers and Chemical Engineering, vol. 28, no. 8, pp. 1489-1498, July 2004.

[3]. R. Feng, W. Shen and H. Shao, "A Soft Sensor Modeling Approach Using Support Vector Machines," in Proceedings of the American Control Conference, Denver, Colorado, 2003.

[4]. H. Wang and D. Hu, "Comparison of SVM and LS-SVM for Regression," in International Conference on Neural Networks and Brain, 2005 (ICNN&B '05), Beijing, 2005.

[5]. M. S. Yang, "A Survey of Fuzzy Clustering," Mathematical and Computer Modelling, vol. 18, no. 11, pp. 1-16, 1993.

[6]. H. J. Xing and B. G. Hu, "An adaptive fuzzy c-means clustering-based mixtures of experts model for unlabeled data classification," Neurocomputing, vol. 71, no. 4-6, pp. 1008-1021, 2008.

[7]. F. Parrella, "Online Support Vector Regression," Genoa, 2007.

[8]. Y.-P. Zhao, J.-G. Sun, Z.-H. Due, Z.-A. Zhang and Y.-B. Li, "Online Independent Reduced Least Squares Support Vector Regression," Information Sciences, vol. 201, pp. 37-52, October 2012.

[9]. D. Basak, S. Pal and D. C. Patranabis, "Support Vector Regression," Neural Information Processing, vol. 11, no. 10, pp. 203-224, October 2007.

[10]. A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Statistics and Computing, vol. 14, no. 3, pp. 199-222, 2004.

[11]. C. D. Meyer, Matrix analysis and applied linear algebra, Philadelphia: Society for Industrial and Applied Mathematics, 2000.

[12]. J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well separated cluster," Journal of Cybernet, vol. 3, no. 3, pp. 32-57, 1973.

[13]. J. C. Bezdek, Pattern recognition with fuzzy objective function algoritms, vol. 2, New York: Plenum Press, 1981, pp. 1-8.

[14]. F. Yongfeng, S. Hongye, Z. Ying and C. Jian, "Adaptive Soft-sensor Modeling Algorithm Based on FCMISVM and Its Application in PX Adsorption Separation Process," Chinese Journal of Chemical Engineering, vol. 16, no. 5, pp. 746-751, 2008.

[15]. W. Wang and Y. Zhang, "On fuzzy cluster validity indices," Fuzzy Sets and Systems, vol. 158, no. 19, pp. 2095-2117, October 2007.

[16]. M. Fazel Zarandi, M. Faraji and M. Karbasian, "An Exponential Cluster Validity Index for Fuzzy Clustering with Crisp and Fuzzy Data," Scientia Iranica Transaction on Industrial Engineering, vol. 17, no. 2, pp. 95-110, December 2010.

[17]. S. Kamelian, "Adaptive distributed control of industrial plants using stability-based nonlinear technique," Ahwaz, 2012.

[18]. A. Farag and R. M. Mohamed, "Regression Using Support Vector Machines: Basic Foundations," Louisville, 2004.

[19]. D. Sun, Y. Li, Q. Wang, C. Le, H. Lv, C. Huang and S. Gong, "A novel support vector regression model to estimate the phycocyanin concentration in turbid inland waters from hyperspectral reflectance," Hydrobiologia, vol. 680, no. 1, pp. 199-217, January 2012.