

Adaptive Soft-sensor Modeling Algorithm Based on FCMISVM and Its Application in PX Adsorption Separation Process^{*}

FU Yongfeng (傅永峰)¹, SU Hongye (苏宏业)^{2,**}, ZHANG Ying (张英)³ and CHU Jian (褚健)²

¹ Modern Education Technology Center, College of Education, Zhejiang University, Hangzhou 310027, China

² National Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China

³ BI Center of Competency, IBM China SWG, Shanghai 200021, China

Abstract To overcome the problem that soft sensor models cannot be updated with the process changes, a soft sensor modeling algorithm based on hybrid fuzzy c-means (FCM) algorithm and incremental support vector machines (ISVM) is proposed. This hybrid algorithm FCMISVM includes three parts: samples clustering based on FCM algorithm, learning algorithm based on ISVM, and heuristic sample displacement method. In the training process, the training samples are first clustered by the FCM algorithm, and then by training each clustering with the SVM algorithm, a sub-model is built to each clustering. In the predicting process, when an incremental sample that represents new operation information is introduced in the model, the fuzzy membership function of the sample to each clustering is first computed by the FCM algorithm. Then, a corresponding SVM sub-model of the clustering with the largest fuzzy membership function is used to predict and perform incremental learning so the model can be updated on-line. An old sample chosen by heuristic sample displacement method is then discarded from the sub-model to control the size of the working set. The proposed method is applied to predict the *p*-xylene (PX) purity in the adsorption separation process. Simulation results indicate that the proposed method actually increases the model's adaptive abilities to various operation conditions and improves its generalization capability.

Keywords soft sensor, fuzzy c-means, incremental support vector machines, heuristic sample displacement method, *p*-xylene purity

1 INTRODUCTION

In chemical industrial process, it is very popular that some variables cannot be measured directly on-line and the soft-sensor technique is a very effective method to implement the real-time estimation of these variables. Overcoming several deficiencies of artificial analysis and on-line analytical instrumentation, the soft-sensor technique has become an important method for implementing on-line quality control, advanced process control, and optimization control [1, 2].

The most popular methods of soft-sensor modeling are the first principle modeling method, the data-driven modeling method, and the hybrid modeling method. In practical operation, as time proceeds, whichever model will be ineffective because of the time-varying characteristic of industrial process. The process parameters will change with time, and therefore, the soft-sensor model will no longer suit the new operation condition. In such cases, the soft-sensor model should be rectified. The general way is to re-build the soft-sensor model using samples collected recently, but it will abandon the historical training results. If the collected samples do not evenly distribute throughout the operation interval or have large noise, when the process parameters change very complex, the soft-sensor model built by these samples may be a partial model and cannot estimate exactly. The best method is to build an adaptive model that can learn and update on-line [3].

Incremental learning is an important research area of machine learning. It is a method that adds samples to learn on the basis of the original learning

result. Compared with ordinary learning methods, on one hand, it can take full advantages of the historical training results and significantly reduce the follow-up training time; on the other hand, it does not need to save historical data and reduces the storage space requirements. It is very suitable for on-line learning [4].

In recent years, Vapnik proposed support vector machines (SVM) that are firmly grounded in the framework of statistical learning theory for limited samples [5]. SVM seeks the optimal classification hyperplane of samples by solving a quadratic programming. Since SVM effectively solves a series of difficulties such as nonlinearity and overlearning, it is widely used in soft-sensor modeling. SVM can easily achieve incremental learning owing to its characteristics [6].

In this article, an adaptive soft-sensor modeling algorithm based on hybrid fuzzy c-means (FCM) and incremental SVM (ISVM) is proposed. Since most industrial processes have more than one operation point, FCM is first used to classify the training samples. Then, each cluster of the training samples is trained by SVM respectively and obtains its corresponding regressive function. In the estimate process, for each incremental sample, its fuzzy membership function to each cluster is first calculated, and then, it is estimated by the regressive function of the cluster with the largest fuzzy membership function. At the same time, this incremental sample that has new process information is introduced in the model for incremental learning so that the model can be updated on-line. To control the size of the working sets, when a new support vector is added, an old one in the corresponding working set is discarded using the heuristic

Received 2007-06-22, accepted 2008-05-28.

^{*} Supported by the National Natural Science Foundation of China (60421002) and priority supported financially by "the New Century 151 Talent Project" of Zhejiang Province.

^{**} To whom correspondence should be addressed. E-mail: hysu@iipc.zju.edu.cn

sample displacement method. This method can increase the model's adaptive abilities to various operation conditions and improve its estimate performance. Based on practical process data, the proposed method is applied to build *p*-xylene (PX) purity soft-sensor model. Simulation results indicate that the proposed method is effective.

2 ALGORITHM DESCRIPTION

The architecture of the hybrid algorithm is shown in Fig. 1. X is the training set and it is clustered by FCM to X_i ($i=1,2,\dots,M$); M is the number of clusters. Each cluster is trained by the respective SVM_i to obtain its corresponding regressive function f_i ($i=1,2,\dots,M$). In the predicting process, for each incremental sample (x_c, y_c) , FCM is used to compute the fuzzy membership function of x_c to each cluster and estimates it by the regressive function with the largest fuzzy membership function value. Then, incremental and decremental learning are used for the corresponding SVM to obtain the new regressive function.

The complexity of the SVM learning algorithm depends on the number of support vectors. If the number of samples is very large, its calculating workload is very large. Furthermore, there usually exists more than one work point of industrial processes; if the input/output relationship is described by only one model, the generalization capacity of the model will be very poor. In our algorithm, FCM is used to cluster the samples and different SVM models are used to describe the input/output relationship of different clusters. Therefore, it can greatly reduce the calculating workload and improve the generalization capacity. Besides, when a sample that has new process information is introduced in the model, incremental learning is used such that the model can be updated on-line, which accordingly improves the model's adaptive abilities. At the same time, when a new support vector

joins the working set, an old one is discarded according to the heuristic sample displacement method thereby avoiding the support vector exploding when the model is used on-line. Each part of the algorithm is described as follows.

2.1 FCM algorithm

The FCM algorithm was first known as the fuzzy ISODATA algorithm. It is a widely used algorithm, which was extrapolated by Dunn [7] from c-means and was generalized by Bezdek [8].

Suppose there is a sample set $X = \{x_1, x_2, \dots, x_k, \dots, x_n\}$, where, $x_k = [x_{k1} x_{k2} \dots x_{km}]$, $k=1,2,\dots,n$. When FCM is used for samples clustering, the algorithm is described as follows,

$$J(\tilde{U}, V) = \sum_{k=1}^n \sum_{i=1}^R (\mu_{ik})^{m'} (d_{ik})^2 \quad (1)$$

where, $\tilde{U} = [U_1 U_2 \dots U_i \dots U_R]^T$ is the fuzzy membership matrix of the k th sample to the i th cluster, $U_i = [\mu_{i1} \mu_{i2} \dots \mu_{ik} \dots \mu_{in}]$, $i=1,2,\dots,R$, $k=1,2,\dots,n$,

and μ_{ik} should satisfy $\mu_{ik} \in (0,1)$ and $\sum_{i=1}^R \mu_{ik} = 1$,

$k=1,2,\dots,n$; $V = [v_1 v_2 \dots v_i \dots v_R]^T$ is the cluster center matrix, $v_i = [v_{i1} v_{i2} \dots v_{ij} \dots v_{im}]$; m' is the exponential weight and should satisfy $m' \in (1, \infty)$, which controls the fuzzy degree of the clusters; and d_{ik} is the distance of the k th sample to the i th cluster center, which is calculated by the equation $d_{ik} = \|x_k - v_i\|^2$.

Minimizing Eq. (1), its solution can be described as follows:

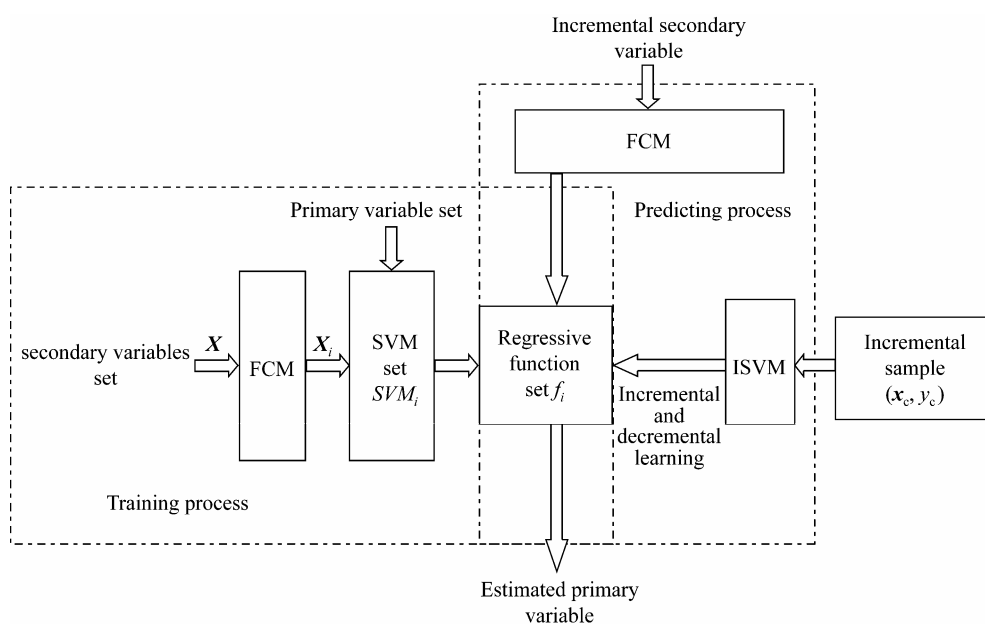


Figure 1 The architecture of the hybrid algorithm

$$\mu_{ik} = \frac{1}{\sum_{j=1}^R \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m'-1)}} \quad (2)$$

$$\mathbf{v}_i = \frac{1}{\sum_{k=1}^n (\mu_{ik})^{m'}} \sum_{k=1}^n (\mu_{ik})^{m'} \mathbf{x}_k \quad (3)$$

Equations (2) and (3) have no analytic solution; the FCM algorithm provides an iterative algorithm to calculate the optimal solution approximately. See related Refs. [7, 8] on the details of the FCM algorithm.

2.2 Incremental support vector machines (ISVM)

2.2.1 Support vector regression (SVR)

SVM was initially used in classification. Subsequently, Vapnik proposed the support vector regression algorithm based on ε -insensitive loss function [5]. Suppose there is a given training data set with l independent and identical distribution samples,

$$\mathbf{X} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), \dots, (\mathbf{x}_l, y_l)\},$$

$$\mathbf{x}_i \in \mathbf{R}^n, \quad i = 1, \dots, l \quad (4)$$

SVR seeks an optimal function $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b$ (where, $\mathbf{w}$ is the weight vector and $b \in \mathbf{R}$ is the threshold value), and thus minimizes the expected risk of prediction. This problem can be described as follows,

$$\begin{aligned} \min W = & \frac{1}{2} \sum_{i,j=1}^l (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(\mathbf{x}_i \cdot \mathbf{x}_j) + \\ & \varepsilon \sum_{i=1}^l (\alpha_i^* + \alpha_i) - \sum_{i=1}^l y_i (\alpha_i^* - \alpha_i) \\ \text{s.t.} & \begin{cases} \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, 2, \dots, l. \end{cases} \end{aligned} \quad (5)$$

In the case of nonlinearity, samples are mapped into a feature space $\mathbf{H}$ from input space $\mathbf{R}^n$ by the map $\varphi: \mathbf{R}^n \rightarrow \mathbf{H}$. The optimal function is then solved in the feature space $\mathbf{H}$ to make the predetermined risk function minimization. According to the Mercer theorem [9], there is a kernel function $K(\cdot, \cdot)$ and it should be met $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$. After the introduction of kernel function, the regressive function becomes

$$f(\mathbf{x}) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) + b \quad (6)$$

where, α_i, α_i^* are Lagrangian multipliers.

2.2.2 Incremental support vector machines (ISVM)

Recently, Cauwenberghs and Poggio proposed ISVM by analyzing the changes of the KKT conditions when an incremental sample was added.

Define sets $\mathbf{X}_{\text{nbsv}} = \{(\mathbf{x}_i, y_i) | 0 < \alpha_i, \alpha_i^* < C\}$, $\mathbf{X}_{\text{esv}} = \{(\mathbf{x}_i, y_i) | \alpha_i, \alpha_i^* = 0\}$, and $\mathbf{X}_{\text{bsv}} = \{(\mathbf{x}_i, y_i) | \alpha_i, \alpha_i^* = C\}$ as non-bound support vectors set, error

support vectors set, and bound support vectors set, respectively. Since $K(\mathbf{x}_i, \mathbf{x}_j)$ is semi-positive definite and all the equation-constraints in Eq. (5) are linear, the following equation can be deduced according to the KKT conditions [10],

$$g_i^* = \begin{cases} |y_i - f(\mathbf{x}_i)| - \varepsilon < 0, \alpha_i, \alpha_i^* = 0, (\mathbf{x}_i, y_i) \in \mathbf{X}_{\text{nsv}} \\ |y_i - f(\mathbf{x}_i)| - \varepsilon = 0, 0 < \alpha_i, \alpha_i^* < C, (\mathbf{x}_i, y_i) \in \mathbf{X}_{\text{nbsv}} \\ |y_i - f(\mathbf{x}_i)| - \varepsilon > 0, \alpha_i, \alpha_i^* = C, (\mathbf{x}_i, y_i) \in \mathbf{X}_{\text{bsv}} \end{cases} \quad (7)$$

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \quad \alpha_i, \alpha_i^* \in [0, C]$$

It can be seen from Eq. (7) that if the samples satisfy the KKT conditions, the samples are within the margin and $g_i^* < 0$. These samples do not change the regression function when they join in the working set. Only if an incremental sample does not satisfy the KKT conditions (in such case, $g_i^* \geq 0$), the incremental learning is carried out.

When the incremental sample $(\mathbf{x}_c, y_c)$ joins in the working set, the KKT conditions are expressed differentially as

$$\begin{aligned} \Delta g_i &= K(\mathbf{x}_i, \mathbf{x}_c) \Delta(\alpha_c - \alpha_c^*) + \\ & \sum_{(\mathbf{x}_j, y_j) \in \mathbf{X}_{\text{nbsv}}} K(\mathbf{x}_i, \mathbf{x}_j) \Delta(\alpha_j - \alpha_j^*) + \Delta b, \\ \Delta g_i^* &= -K(\mathbf{x}_i, \mathbf{x}_c) \Delta(\alpha_c - \alpha_c^*) - \\ & \sum_{(\mathbf{x}_j, y_j) \in \mathbf{X}_{\text{nbsv}}} K(\mathbf{x}_i, \mathbf{x}_j) \Delta(\alpha_j - \alpha_j^*) - \Delta b, \\ \Delta(\alpha_c - \alpha_c^*) + \sum_{(\mathbf{x}_j, y_j) \in \mathbf{X}_{\text{nbsv}}} \Delta(\alpha_j - \alpha_j^*) &= 0. \end{aligned} \quad (8)$$

Define $\gamma_j = \alpha_j - \alpha_j^*$. Since $g_i^* = 0$ for the non-bound support vectors set $\mathbf{X}_{\text{nbsv}} = \{(\mathbf{x}_{s_1}, y_{s_1}), \dots, (\mathbf{x}_{s_n}, y_{s_n})\}$,

the change of g_i^* satisfies: $\Delta g_i^* = 0 | (\mathbf{x}_i, y_i) \in \mathbf{X}_{\text{nbsv}}$. Suppose the kernel function used is Gauss kernel; the following matrix can be obtained from Eq. (8) [11, 12],

$$\mathbf{Q} \cdot \begin{bmatrix} \Delta b \\ \Delta \gamma_{s_1} \\ \vdots \\ \Delta \gamma_{s_n} \end{bmatrix} = - \begin{bmatrix} 1 \\ K(\mathbf{x}_{s_1}, \mathbf{x}_c) \\ \vdots \\ K(\mathbf{x}_{s_n}, \mathbf{x}_c) \end{bmatrix} \Delta \gamma_c \quad (9)$$

where,

$$\mathbf{Q} = \begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & K(\mathbf{x}_{s_1}, \mathbf{x}_{s_1}) & \dots & K(\mathbf{x}_{s_1}, \mathbf{x}_{s_n}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(\mathbf{x}_{s_n}, \mathbf{x}_{s_1}) & \dots & K(\mathbf{x}_{s_n}, \mathbf{x}_{s_n}) \end{bmatrix} \quad (10)$$

In Eq. (10), $\mathbf{x}_{s_j}$ denotes the j th non-bound support vector ($j = 1, \dots, n$, n is the number of non-bound support vectors). With coefficient sensitivities given

by $\beta = \Delta b / \Delta \gamma_c, \dots, \beta_j = \Delta \gamma_j / \Delta \gamma_c, \dots, \beta_{s_n} = \Delta \gamma_{s_n} / \Delta \gamma_c, j = 1, \dots, s_n$, the following equation is obtained,

$$\begin{bmatrix} \beta \\ \beta_{s_1} \\ \vdots \\ \beta_{s_n} \end{bmatrix} = -\mathbf{R} \begin{bmatrix} 1 \\ K(\mathbf{x}_{s_1}, \mathbf{x}_{s_c}) \\ \vdots \\ K(\mathbf{x}_{s_n}, \mathbf{x}_{s_c}) \end{bmatrix} \quad (11)$$

where, $\mathbf{R} = \mathbf{Q}^{-1}$.

For each incremental sample, it satisfies

$$\Delta g_i^* = \begin{cases} \pm \Delta \gamma_c \pi_i, (\mathbf{x}_i, y_i) \notin \mathbf{X}_{\text{nbsv}} \\ 0, (\mathbf{x}_i, y_i) \in \mathbf{X}_{\text{nbsv}} \\ i = 1, \dots, l+1 \end{cases} \quad (12)$$

where, $\pi_i = K(\mathbf{x}_i, \mathbf{x}_c) + \sum_{(\mathbf{x}_j, y_j) \in \mathbf{X}_{\text{nbsv}}} K(\mathbf{x}_i, \mathbf{x}_j) \beta_j + \beta$.

When an incremental sample joins $\mathbf{X}_{\text{nbsv}}$, the matrix $\mathbf{R}$ is updated as follows,

$$\mathbf{R} \leftarrow \begin{bmatrix} & & 0 \\ & \mathbf{R} & \vdots \\ & & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix} + \frac{1}{\pi_c} \begin{bmatrix} \beta \\ \beta_{s_1} \\ \vdots \\ \beta_{s_n} \\ 1 \end{bmatrix} \begin{bmatrix} \beta & \beta_{s_1} & \dots & \beta_{s_n} & 1 \end{bmatrix} \quad (13)$$

As described above, when a new sample joins, the coefficient sensitivities matrix is first calculated according to current matrix $\mathbf{R}$ by Eq. (11); then, sets $\mathbf{X}_{\text{bsv}}$, $\mathbf{X}_{\text{esv}}$, and $\mathbf{X}_{\text{nbsv}}$ are updated by Eqs. (8) and (12). If the incremental sample is a support vector, matrix $\mathbf{R}$ is updated by Eq. (13).

2.2.3 Heuristic sample displacement method

When a new sample joins, ISVM does incremental learning based on the original training result. It does not need retraining the whole samples set. If ISVM is used in soft-sensor modeling directly, the following problem will appear: as time proceeds, the number of support vectors will become very large and it is very difficult to memorize and update matrix $\mathbf{R}$. Thus, the size of the support vector set must be controlled in a certain scale. The approach adopted here is: when a new support vector joins, a not very important support vector will be discarded from the working set, thus maintaining a fixed number of support vectors. In this article, a heuristic sample displacement method based on support vectors data domain description is used to search the support vector that should be discarded.

Support vectors data domain description (SVDD) is described as follows: the data objects in input space are mapped into a higher dimensional feature space; then, a sphere with minimum radius, which contains all (or most of) the data objects is tried to be found in this feature space,

$$\begin{aligned} \max \mathbf{Q}(\beta) &= \sum_{i=1}^l K(\mathbf{x}_i, \mathbf{x}_i) \beta_i - \sum_{i,j=1}^l \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t. } 0 &\leq \beta_i \leq C, \quad i = 1, 2, \dots, l \end{aligned} \quad (14)$$

Solving the above optimization problem, the radius of the sphere in this feature space and the distance $[D(\mathbf{x}_i)]$ of a sample $\mathbf{x}_i$ to the center of the sphere can be obtained. $D(\mathbf{x}_i)$ can be calculated by the following equation,

$$D^2(\mathbf{x}_i) = \sum_{j=1}^l \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) + K(\mathbf{x}_i, \mathbf{x}_i) - 2 \sum_{j=1}^l K(\mathbf{x}_j, \mathbf{x}_i) \beta_j \quad (15)$$

The value of $D(\mathbf{x}_i)$ reflects the extent of a sample deviating from the cluster that it belongs to. If $D(\mathbf{x}_i)$ is smaller, the sample $\mathbf{x}_i$ is close to the regression line; along with the accretion of $D(\mathbf{x}_i)$, the sample $\mathbf{x}_i$ is gradually away from the regression line. If the value of $D(\mathbf{x}_i)$ is large enough, the sample $\mathbf{x}_i$ is a novelty. Therefore, the sample that has the largest value of $D(\mathbf{x}_i)$ can be discarded from the working set after each incremental learning [13].

For the sample displacement method mentioned above, decremental unlearning should be carried out after a support vector is discarded. Decremental unlearning is the reversal process of incremental learning. Decremental unlearning may eliminate the impact brought by discarding a sample directly. Similar to incremental learning, after a decremental sample $(\mathbf{x}_k, y_k)$ is discarded from the working set, matrix $\mathbf{R}$ is updated according to the following steps,

$$R_{ij} \leftarrow R_{ij} - R_{kk}^{-1} R_{ik} R_{kj} \quad (16)$$

where, R_{ij} indicates the element (i, j) in matrix $\mathbf{R}$.

2.2.4 The modeling steps of FCMISVM

Suppose the current working set is $\mathbf{X}$, and the incremental set is $\mathbf{I}$. Firstly, $\mathbf{X}$ is clustered by FCM; thus, $\mathbf{X}$ is clustered to $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_b, \dots, \mathbf{X}_M\}$ ($b = 1, \dots, M$; M is the number of clusters). Then, each $\mathbf{X}_b$ is trained by SVR, respectively, and its corresponding regression functions $f(\mathbf{x})$ and g_i^* can be obtained. For each sample $(\mathbf{x}_c, y_c)$ in $\mathbf{I}$, its fuzzy membership function to each cluster is first calculated, and then estimated using the regression function with the largest value of the fuzzy membership function. Then, incremental learning and decremental unlearning are carried out according to the following steps [14].

Step 1: Initialize $\Delta \alpha_c^*$ to zero. Check the incremental sample $(\mathbf{x}_c, y_c)$; if it satisfies the KKT conditions, that is, $g_c^* < 0$, terminate.

Step 2: If the incremental sample does not satisfy the KKT conditions, that is, $g_c^* \geq 0$, then, apply the largest possible increment $\Delta \alpha_c^*$ so that one of the following conditions occurs,

(1) $g_c^* = 0$: Add $(\mathbf{x}_c, y_c)$ to $\mathbf{X}_{\text{nbsv}}$, and update $\mathbf{R}$ according to Eq. (13).

(2) $\Delta \alpha_c^* = C$: Add $(\mathbf{x}_c, y_c)$ to $\mathbf{X}_{\text{bsv}}$, and terminate.

(3) Update the KKT conditions according to Eqs. (8) and (12) and then update the membership of elements of sets $\mathbf{X}_{\text{bsv}}$, $\mathbf{X}_{\text{esv}}$, $\mathbf{X}_{\text{nbsv}}$ according to the KKT conditions; repeat Step 2 till convergence.

Step 3: Find out the decremental sample $(\mathbf{x}_k, y_k)$

according to the heuristic sample displacement method and initialize $\Delta\alpha_k^*$ to zero.

Step 4: If $g_k^* > 0$, apply the largest possible increment $\Delta\alpha_k^*$ such that one of the following conditions occurs,

(1) $\alpha_k^* = 0$: Decremental sample (x_k, y_k) is an error vector. Discard (x_k, y_k) from the corresponding X_b , that is, $X_b = X_b \setminus \{(x_k, y_k)\}$ and terminate. Go to Step 6.

(2) Update the KKT conditions according to Eqs. (8) and (12), and then update the membership of sets X_{bsv} , X_{esv} , X_{nbsv} according to the KKT conditions; repeat Step 4 till convergence.

Step 5: Update matrix R according to Eq. (16)

and update the regression function $f(x) = \sum_{i=1}^t (\hat{\alpha}_i - \hat{\alpha}_i^*) K(\hat{x}_i, x) + \hat{b}$ (t is the number of samples of the corresponding cluster), where, $\hat{\alpha}_i$, $\hat{\alpha}_i^*$, $\hat{x}_i$, $\hat{b}$ are taken from the previous steps.

Step 6: Repeat Step1 if a new sample arrives.

3 SOFT-SENSOR MODELING BASED ON FCMISVM AND ITS APPLICATION IN PX ADSORPTION SEPARATION PROCESS

3.1 Process description

With the rapid development of polyester industry, the consumption of PX increases sharply. At present, the main method to produce PX is separating PX from a mixture of C₈ aromatics by simulated moving bed adsorption separation process. The flow sheet of this process is shown in Fig. 2. Two pumps provide the liquid circulation from the bottom of one adsorbent chamber to the top of another one. The position of the injection and withdrawal of the different streams along the adsorbent chambers varies during the cycle in each stage of the total 24-stages. A five-automatic-valve manifold is provided at each stage to introduce feed, desorbent, backwash, and to withdraw extract and raffinate. The dilute extract is sent to the extract column for separation of the extract from the desorbent and the overhead from the extract column is sent to a rectifying column where the highly pure PX product is separated from any toluene and benzene

cuts, which may have been present in the feed. To increase the concentration of PX in zone II, a stream is withdrawn from the top section of the extract column and sent back to the desorbent column as a backwash. The dilute raffinate is sent to the raffinate column for separation of the raffinate from the desorbent and the raffinate product is then sent to an isomerization unit where additional PX is formed and then recycled back to the process. The desorbent from the bottom of both extract and raffinate column is recycled back to the adsorbent chambers through pipe line [15, 16].

In practice, the PX purity in the adsorption separation process cannot be measured on-line. It is sampled three times each day and measured by manual analysis. Manual analysis is a laboratory technique with considerable delay, and therefore, it cannot meet the requirements of real-time control. To implement on-line PX purity control, the real-time estimate values must be obtained. Building a soft-sensor model is a good alternative method.

3.2 PX purity soft-sensor modeling based on FCMISVM

The main factors that can affect the PX purity are the flow of the five process streams (feed, extract, raffinate, desorbent, and backwash), the temperature of the column, and the pressure of the column. Through in-depth study of the mechanism, the ratio of desorbent and the feed flow, the ratio of the recycling and the feed flow, the ratio of extract and raffinate, the ratio of backwash and the feed flow, the operating temperature, and the operating pressure are chosen to be the secondary variables, and the primary variable is PX purity. After preprocessing and standardizing, a data set with 400 samples can be obtained. The data set is divided into two sets averagely. One is used as training set, and the other is used as testing set.

The FCMISVM algorithm is reprogrammed by MATLAB based on LIBSVM. In the training process of FCMISVM and SVR, the kernel function is chosen as Gauss kernel $K(x_i, x_j) = e^{-p\|x_i - x_j\|^2}$, and the optimal values of parameters p and C (obtained by 10 times cross test) are 0.5 and 10, respectively. The predicted results of the soft-sensor model based on SVR and FCMISVM (the sample displacement method is based on SVDD) are shown in Fig. 3.

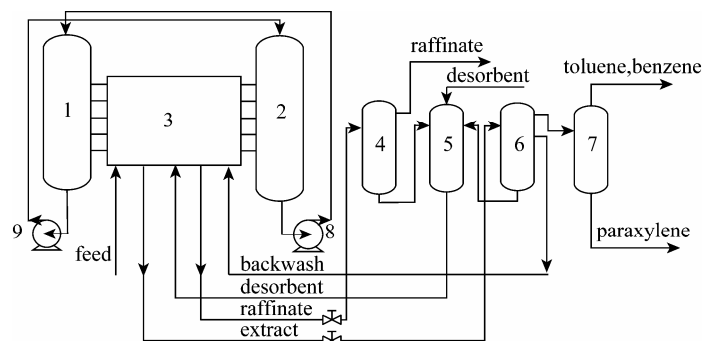
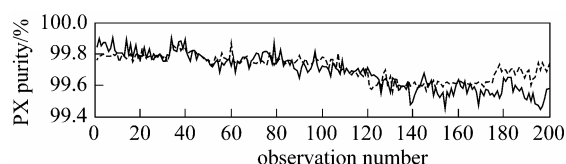
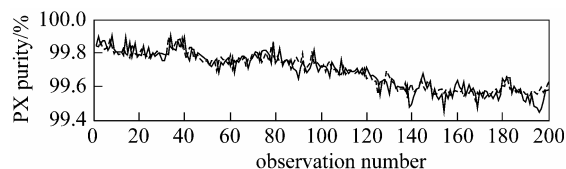


Figure 2 Schematic layout of simulated moving bed adsorption separation process

1, 2—adsorbent chamber; 3—concentrated pipes; 4—raffinate column; 5—desorbent buffer tank; 6—extract column; 7—PX rectifying column; 8, 9—recycle pump



(a) Analysis values and predicted values of SVR



(b) Analysis values and predicted values of FCMISVM

Figure 3 Comparison of the predicted values of SVR and FCMISVM

— predict value; ---- analysis value

From Fig. 3, it can be seen that the predicted errors of the soft-sensor model based on SVR will become larger and larger as time proceeds. The predicting line begins to leave the analysis line at the 150th sample, while the soft-sensor models based on FCMISVM has better prediction capability.

To test the estimate capability of FCMISVM, the prediction root-mean-square-error of FCMISVM and SVR based on different sizes of testing sets is shown in Table 1. From Table 1, it can be seen that the root-mean-square-error of SVR will become larger and larger with the increase of the number of testing samples, but the root-mean-square-error of FCMISVM remains almost unchanged. The reason is that the soft-sensor model based on FCMISVM first finds the cluster that the testing sample belongs to, and then adopts on-line incremental learning. Thus, it can rectify the model on-line when a new sample is collected. Therefore, it possesses better generalization capacity.

Table 1 The prediction root-mean-square-errors of SVR and FCMISVM based on different sizes of testing set

Observation samples	SVR	FCMISVM
50	0.1185	0.0954
100	0.1206	0.0956
150	0.1424	0.0961
200	0.1785	0.0965

4 CONCLUSIONS

A soft-sensor modeling method based on FCMISVM is proposed in this article. First, FCM is used to classify the training set. Then, each cluster of the training samples is trained by the respective SVM and the corresponding regressive function is solved. In the predicting process, for each incremental sample, its fuzzy membership function to each cluster is first calculated, and then it is predicted by the regressive function of the cluster with the largest fuzzy membership function. At the same time, this incremental sample that has new process information is introduced in the model to perform incremental learning. To control the size of the working sets, when a new support vector is added, an old one in the respective working sets

is discarded using the heuristic samples displacement method. Based on practical process data, the proposed method is applied to build PX purity soft-sensor model. Simulation results indicate that the proposed method can enhance the model's adaptive capability to various operation conditions, improve its generalization capacity, and increase its prediction capacity.

NOMENCLATURE

b	threshold value
d_{ik}	the distance of the k th sample to the i th cluster center
f	regressive function
H	feature space
M	the number of clusters
m'	exponential weight
$\tilde{U}$	the fuzzy membership matrix
V	the cluster center matrix
w	weight vector
X	independent matrix
X_{bsv}	bound support vectors set
X_{esv}	error support vectors set
X_{nbsv}	non-bound support vectors set
Y	dependent matrix
α	Lagrangian multiplier
α^*	Lagrangian multiplier
β	coefficient sensitivities
ε	insensitive loss function

REFERENCES

- McAvoy, T.J., "Contemplative stance for chemical process control", *Automatica*, **28** (2), 441–442 (1992).
- Zhong, W., Liu, A.L., Yu, J.S., "Studies on building soft-sensing models of multivariable systems", *Control and Decision*, **15** (2), 209–212 (2000).
- Liu, R.L., "Some studies on soft sensor technology and their applications to industry process", Ph.D. Thesis, Zhejiang Univ., China (2004). (in Chinese)
- Xiao, R., Wang, J.C., Sun, Z.X., Zhang, F.Y., "An SVM incremental learning algorithm α -ISVM", *Journal of Software*, **12** (12), 1818–1824 (2001).
- Vapnik, V.N., *Statistical Learning Theory* (translated by Xu, J.H., Zhang, X.G.), Publishing House of Electronics Industry, Beijing (2004).
- Yan, W.W., Shao, H.H., "Application of support vector machines and least squares support vector machines to heart disease diagnoses", *Control and Decision*, **18** (3), 358–360 (2003).
- Dunn, J.C., "A fuzzy relative of the ISODATA process and its use in detecting compact well separated cluster", *Journal of Cybernet*, **3**, 32–57 (1974).
- Bezdek, J.C., "A convergence theorem for the fuzzy ISODATA clustering algorithm", *IEEE Trans Pattern Anal Mach Intel*, **2** (1), 1–8 (1980).
- Burges, C.J.C., "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, **2** (2), 1–47 (1998).
- Chang, C.C., Lin, C.J., "Training nu-support vector regression: theory and algorithms", *Neural Computation*, **14** (2), 1959–1977 (2002).
- Naqa, I.E., Yang, Y., "Relevance feedback based on incremental learning for mammogram retrieval", *Image Processing*, **4** (2), 7–11 (2003).
- Cauwenberghs, G., Poggio, T., "Incremental and decremental support vector machine learning", *Advances in Neural Information Processing Systems*, (NIPS' 2000), **13**, 409–415, MIT Press, MA (2001).
- Tax, D.M.J., Duin, R.P.W., "Data domain description using support vectors", In: *Proc. of 8th European Symposium on Artificial Neural Networks*, D-Facto Public., Bruges, Belgium, 251–256 (1999).
- Zhang, Y., "A study on process industrial data mining based on support vector machines", Ph. D. Thesis, Zhejiang Univ., China (2005). (in Chinese)
- Pavone, D., Hotier, G., "System approach modeling applied to the eluxyl process", *Oil & Gas Science and Technology*, **55** (4), 437–446 (2000).
- Minceva, M., Rodrigues, A.E., "Modeling and simulation of a simulated moving bed for the separation of *p*-xylene", *Ind. Eng. Chem. Res.*, **41**, 3454–3461 (2002).