

Dynamic Modeling of Biotechnical Process Based on Online Support Vector Machine

Xianfang Wang^{1,2}

1. Jiangnan University / School of Communication and Control Engineering, Wuxi, China

2. Henan Institute of Science and Technology / Department of Information Engineering, Xinxiang, China
Email: xfwang11@yahoo.com.cn

Zhiyong Du

Henan Mechanical and Electrical Engineering College/ Department of Computer, Xinxiang, China
Email: zhuydu@126.com

Jindong Chen and Feng Pan

Jiangnan University / School of Communication and Control Engineering, Wuxi, China
Email: pan_feng_63@163.com

Abstract—Due to the complexity and high non-linearity of biotechnical process, most simple mathematical models cannot describe the behavior of biochemistry systems very well. Therefore, dynamic modeling of biotechnical process is indispensable. Support vector machine (SVM) is a novel machine learning method, which is powerful for the problem characterized by small sample, non-linearity, high dimension and local minima, and has high generalization. But currently most support vector machine regression (SVR) training algorithms are offline, which could not be suit for time-variant system. So an improved SVM called online support vector machine was presented to modeling for the dynamic feature of fermentation process. The model based on the modified SVM was developed and demonstrated using simulation experiments. Some models based on SVM were also presented. The result shows that the modeling based online SVM is superior to modeling based on SVW.

Index Terms—dynamic modeling, biotechnical process, online support vector machine, on-line estimation

I. INTRODUCTION

Biotechnical process is a kind of complicated reaction processes. Control of biotechnical process that was in nearly empty for twenty years has recently gained an unprecedented development because the methods of system identification and optimized control tend to mature. An accurate and effective mathematical model is a must for implementing control and optimization of biotechnical processes.

Three kinds of model are generally used for describing the characteristics of biotechnical processes and for the purpose of process control and optimization: the white-box, the black-box model, the gray-box. The white-box is one kind of unstructured dynamic model, for example, the Monod growth model and the Luedeking-Piret product formation model, uses ordinary differential equations to describe the time changes of those state

variables such as the concentrations of cells, substrates, metabolic products, etc. The unstructured model is mainly used for the purposes of off-line process prediction, control, and optimization. There are two major disadvantages generally accompany with the unstructured model^[1], one is that heavy works are required to identify a large amount of model parameters while the physiological meanings of those parameters is not so clear; the other is that the un-structured model can not really grasp the dynamic changes due to the changes of environmental factors, operating conditions, and the fermentation batch runs, therefore, the general predicting and control performance based on the model is limited. The black-box model is completely based on the input-output series data or the apparent behaviors of the processes, and the real mechanisms or natures of the metabolic reactions are not considered at all. The model parameters are not of any physical or chemical meanings. The artificial neural network (ANN) model is one kind of black-box model, which is quite popular in fermentation process modeling for its excellent ability dealing with the complex and non-linear process characteristics. However, a fetal shortcoming of the ANN model is that, a huge amount of experimental data pairs must be provided for training the ANN model, otherwise the general performance of the model will be significantly deteriorated^[2, 3]. The gray-box is one kind of hybrid-model, such as, fuzzy logic inference model is a human experience and knowledge-based qualitative model, its general prediction power and accuracy largely rely on the experiential fuzzy logic rules and membership functions, whereas the development and adaptive adjustment of fuzzy logic rules and membership functions are a time and labor consuming process^[4].

But, limited by available biological sensors, nowadays above method are almost off-line and manual have the drawbacks of complicated operation, great time lag and sample contamination^[5]. Although there are some kinds

of on-line measurement based on hardware sensors [6, 7], they have their own limitations, and the corresponding on-line instruments are expensive and difficult in maintenance.

Support vector machine (SVM) is a new and valid machine-learning algorithm and has been well used for classification, function regression, and time series prediction, etc. [8-11]. Compared with neural network, SVM has good generalization ability, and is especially suitable for machine learning in small sample condition. The training algorithm of SVM will not run into local minimum point. Also it can automatically construct the structure of system model. However, currently most support vector machine regression (SVR) training algorithms are offline, but online SVR algorithm is more useful when the system to be identified is time-variant, because this kind of algorithms can automatically track changes of system model with time-varying and time-lagging characteristics [12, 13].

According to the shortcoming in above generally modeling method in biotechnical processes, a dynamic modeling was built by utilizing the mechanism of fermentation and characteristics of online SVM in this paper. This method was adopted for glutamic acid fermentation process, the result shows that the dynamic modeling based on the online-SVR algorithm can get a good result.

The paper is organized as follows: In Section II the improved online SVR algorithm is introduced in detail. In Section III, two modeling method of glutamic acid concentration are designed based on SVM and online SVM respectively. Two simulation results are given to show the validity of two modeling methods. In Section IV, the comparisons of online SVR algorithm-based modeling with offline SVR algorithm for modeling biotechnical process. Some conclusions are obtained and put forward in Section V.

II. ONLINE SUPPORT VECTOR REGRESSION

A. Support Vector Machine

SVM is an excellent method aiming to finite data points based on statistical learning theory (SLT) [14, 15], advanced by Vapnik in 1990's. Given a training sample set: $T = \{x_i, y_i, i = 1, 2, \dots, l\}$, $x_i \in R^N$ and $y_i \in R$,

$$f(x) = \sum_{i=1}^l w_i \phi_i(x) + b \tag{1}$$

where $\{\phi_i(x)\}_{i=1}^l$ is the data in features space, $\{w_i(x)\}_{i=1}^l$ and b are coefficients. They can be estimated by minimizing the regularized risk function

$$R(C) = C \frac{1}{l} \sum_{i=1}^l L_\varepsilon(y, f(x_i)) + \frac{1}{2} \|w\|^2 \tag{2}$$

where $L_\varepsilon(y, f(x_i))$ is the so-called loss function measuring the approximate errors between expected output y_i and calculated output $f(x_i)$ and C is regularization constant determining the trade-off between the training error and the generalization performance. The

second term, $\frac{1}{2} \|w\|^2$, is used as a measurement of function flatness. Introduction of slack variables ξ, ξ^* leads (2) to the following constrained function

$$\min_{w, b, \xi} J = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \tag{3}$$

s.t.

$$\begin{cases} y_i - w\phi(x_i) - b \leq \varepsilon + \xi_i \\ w\phi(x_i) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i \geq 0 \\ \xi_i^* \geq 0 \end{cases} \quad i = 1, 2, \dots, l$$

Using the duality principle, the equation (3) could be changed equation (4)

$$\max_{\alpha, \alpha^*} \{L_D = -\frac{1}{2} \sum_{i, j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) x_i \cdot x_j \tag{4}$$

$$- \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*)\}$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i \leq C \\ 0 \leq \alpha_j \leq C \end{cases}$$

Although non-linear function ϕ is usually unknown all computations related to ϕ can be reduced to the form $\phi(x)^T \phi(y)$, which can be replaced with a so-called kernel function $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ that satisfies Mercer's condition [16, 17]. Then, Eq. (1) becomes the explicit form

$$f(x, \alpha_i, \alpha_i^*) = \sum_{x_i \in SV} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{5}$$

In equation (5), Lagrange multipliers α_i and α_i^* satisfy the equality $\alpha_i \times \alpha_i^* = 0, \alpha_i \geq 0, \alpha_i^* \geq 0, i = 1, \dots, l$. Those vectors with $\alpha_i \neq 0$ are called support vectors, which contribute to the final solution.

B. Online Support Vector Machine

The Online Support Vector Regression algorithm main purpose is to verify these conditions after that a new sample is added. The online SVR algorithm can be described as follows:

Let $\theta_i = \alpha_i - \alpha_i^*$, use equation (6), define a margin function $h(x_i)$ as,

$$h(x_i) = f(x_i) - y_i = \sum_{j=1}^l \theta_j K(x_j, x) + b - y_i \tag{6}$$

According to Lagrange Multipliers and Karush-Kuhn-Tucker (KKT) conditions, training samples can be separated into three sets [18], (as shown in Fig.1):

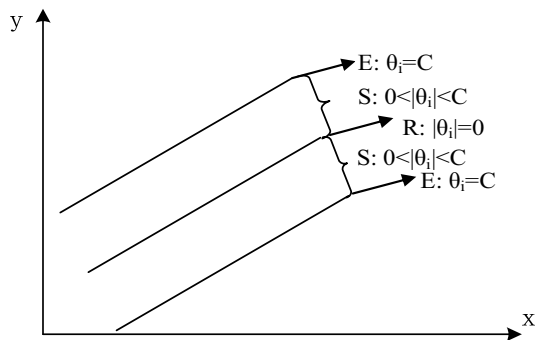


Figure 1. Three subsets gotten from training samples set

Set E: Error support vectors,

$$E = \{i \mid |\theta_i| = C, |h(x_i)| \geq \varepsilon\},$$

Set S: Margin support vectors,

$$S = \{i \mid 0 < |\theta_i| < C, |h(x_i)| = \varepsilon\},$$

Set R: Remaining samples,

$$R = \{i \mid |\theta_i| = 0, |h(x_i)| \leq \varepsilon\}$$

When a new sample x_c is added, the goal is to let x_c enter one of the three sets, while KKT condition still can be satisfied automatically.

At first, let $\theta_c = 0$, from equation (6), we can obtain:

$$h(x_c) = f(x_c) - y_c = \sum_{j=1}^l \theta_j K(x_c, x_j) + b - y_c \quad (7)$$

then we gradually change the value of θ_c and at the same time make all other samples satisfy KKT condition. The change of θ_c may change the values of θ_i and $h(x_i)$ of other samples. It satisfies the formula:

$$\Delta h(x_i) = K(x_i, x_c) \Delta \theta_c + \sum_{j=1}^l K(x_i, x_j) \Delta \theta_j + \Delta b \quad (8)$$

$$\theta_c + \sum_{i=1}^l \theta_i = 0 \quad (9)$$

the relation among $\Delta \theta_c, \Delta \theta_i$ and b could be obtained:

$$\begin{bmatrix} \Delta b \\ \Delta \theta_{s_1} \\ \vdots \\ \Delta \theta_{s_{l_s}} \end{bmatrix} = \beta \Delta \theta_c \quad (10)$$

where $\theta_{s_i} \in S$, l_s is sample number of set S, and

$$\beta = \begin{bmatrix} \beta \\ \beta_{s_1} \\ \vdots \\ \beta_{s_{l_s}} \end{bmatrix} = -R \begin{bmatrix} 1 \\ K(x_{s_1}, x_c) \\ \vdots \\ K(x_{s_{l_s}}, x_c) \end{bmatrix} \quad (11)$$

where

$$R = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K(x_{s_1}, x_{s_1}) & \cdots & K(x_{s_1}, x_{s_{l_s}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_{s_{l_s}}, x_{s_1}) & \cdots & K(x_{s_{l_s}}, x_{s_{l_s}}) \end{bmatrix} \quad (12)$$

The relation among $\Delta \theta_c$ and $\Delta h(x_i)$ can be obtained equation (13):

$$\begin{bmatrix} \Delta h(x_{n_1}) \\ \Delta h(x_{n_2}) \\ \vdots \\ \Delta h(x_{n_{l_n}}) \end{bmatrix} = \gamma \Delta \theta_c \quad (13)$$

where $x_{n_i} \in E \cup R \cup C$, l_n is the sample number of set $E \cup R \cup C$, and

$$\gamma = \begin{bmatrix} K(x_{n_1}, x_c) \\ K(x_{n_2}, x_c) \\ \vdots \\ K(x_{n_{l_n}}, x_c) \end{bmatrix} + \begin{bmatrix} 1 & K(x_{n_1}, x_{s_1}) & \cdots & K(x_{n_1}, x_{s_{l_s}}) \\ 1 & K(x_{n_2}, x_{s_1}) & \cdots & K(x_{n_2}, x_{s_{l_s}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_{n_{l_n}}, x_{s_1}) & \cdots & K(x_{n_{l_n}}, x_{s_{l_s}}) \end{bmatrix} \beta \quad (14)$$

Matrix R can be updated with a recursive algorithm according to following formula when a new sample x_i is added to set S:

$$R = \begin{bmatrix} & & 0 \\ & R & \vdots \\ & & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{1}{\gamma_i} \begin{bmatrix} \beta \\ 1 \end{bmatrix} [\beta^T \quad 1] \quad (15)$$

where

$$\beta = -R \begin{bmatrix} 1 \\ K(x_{s_i}, x_{s_i}) \\ K(x_{s_{l_s}}, x_{s_i}) \end{bmatrix} \quad (16)$$

$$\gamma_i = K(x_i, x_i) + [1 \quad K(x_{s_1}, x_i) \quad \cdots \quad K(x_{s_{l_s}}, x_i)] \quad (17)$$

when a sample x_i is removed from R, matrix R is updated as

$$R_{i,j} = R_{i,j} - \frac{R_{i,k} R_{k,j}}{R_{k,k}} \quad (18)$$

where $i, j \in [1, \dots, k, k+2, \dots, s_{l_s} + 1]$.

From above, online SVM regression training algorithm is consisted of two sub-algorithms: incremental algorithm and decremental algorithm.

The main idea of incremental algorithm is that: when a sample x_c is added to training sample set, we gradually change its θ_c and $h(x_c)$ until x_c enter into of one of three sets, and during the process some other samples are also updated from one of sets S, R, and E to another because they are influenced by the change of sample x_c . The goal of the incremental algorithm is to make all samples satisfy the KKT condition while a new sample is added to training sample set.

The main idea of decremental algorithm is that: a sample with zero value of θ_c can be removed safely since it has no influence to SVM model; when a sample x_c is to be removed from training sample set, we gradually change its θ_c and $h(x_c)$ until θ_c change to zero, and during the process some other samples are also updated from one of sets S, R, and E to another because they are

influenced by the change of sample x_c . The goal of the decremental algorithm is to make all samples satisfy the KKT condition while a useless sample is removed out of training sample set.

After long math proofs and analysis the algorithm is finally shown. It's in a pseudocode format. The input requested from an Online Support Vector Regression is the following:

INPUTS

1. Training Set $\{x_i, y_i, i = 1 \dots l\}$
2. Weights $\theta_i, i = 1 \dots l$
3. Bias b
4. Training Set partition into Supposed(S), Error Set (E) and Remaining Set(R)
5. Params: ε, C , Kernel Type and Kernel Params
6. R Matrix
7. New Sample $c = (x_c, y_c)$

OUTPUTS

1. New Training Set $\{x_i, y_i, i = 1 \dots l + 1\}$
2. New Coefficients $\theta_i, i = 1 \dots l + 1$
3. New Bias b
4. New Training Set partition
5. New R Matrix

NEW TRAINING ALGORITHM

1. Add (x_c, y_c) at the Training Set
2. Set $\theta_c = 0$
3. Compute $f(x_c)$ and $h(x_c)$
4. If $(|h(x_c)| < \varepsilon)$
 - 4.1 Add New Sample to the Remaining Set and Exit
5. Compute $h(x_i), i = 1 \dots l$
6. While (New Sample is not added into a set)
 - 6.1 Update the values β and γ
 - 6.2 Find Least Variations ($L_{c1}, L_{c2}, L_s, L_e, L_r$)
 - 6.3 Find Min Variation

$$\Delta\theta_c = \min(L_{c1}, L_{c2}, L_s, L_e, L_r)$$
 - 6.4 Let Flag the case number that determinates $\Delta\theta_c$

$$(L_{c1} = 1, L_{c2} = 2, L_s = 3, L_e = 4, L_r = 5)$$
 - 6.5 Let x_i the sample that determines $\Delta\theta_c$
 - 6.6 Update $\Delta\theta_c, \Delta\theta_i, i = 1 \dots l$ and b
 - 6.7 Update $h(x_i), i \in E \cup R$
 - 6.8 Switch Flag
 - 6.8.1 (Flag = 1)
 - 6.8.1.1 Add New Sample to Support Set
 - 6.8.1.2 Add New Sample to R Matrix
 - 6.8.1.3 Exit
 - 6.8.2 (Flag = 2)
 - 6.8.2.1 Add New Sample to Error Set
 - 6.8.2.2 Exit
 - 6.8.3 (Flag = 3)
 - 6.8.3.1 If $(_1 = 0)$
 - 6.8.3.1.1 Move Sample l from Support to Remaining Set
 - 6.8.3.1.2 Remove Sample l from R Matrix
 - 6.8.3.2 Else $[\theta_i = |C|]$

6.8.3.2.1 Move Sample l from Support to Error Set

6.8.3.2.2 Remove Sample l from R Matrix

6.8.4 (Flag = 4)

6.8.4.1 Move Sample l from Error to Support

6.8.4.2 Add Sample l to R Matrix

6.8.5 (Flag = 5)

6.8.5.1 Move Sample l from Remaining to Support Set

6.8.5.2 Add Sample l to R Matrix

At the beginning, Online SVR tries to check if the new sample can be inserted in the Remaining Set. This is the best case, because it is fast and does not increase the complexity of the machine. If this does not happen, it starts a cycle that finishes only when the new sample is added to Support or Error set (cases 1 and 2). At each iteration, a sample migrates from one set to another. Each migration passes through Support Set, so that at every iteration the R matrix (that is correlated to support samples) changes. Because this is a common and the most expensive operation, in the next sections it will be explained a method to efficiently update the R Matrix.

III. MODELING OF GLUTAMIC ACID FERMENTATION PROCESS

A. Modeling of glutamic acid concentration

We are using a batch glutamic acid fermentation process, which consists of a fermentor and its peripherals: air compressor, agitation system, heat exchanger and control equipment for pH, temperature, dissolved oxygen and foam leave. The fermentor is 30 m³ (Food Industry Institute, Tianjin, China) with pH, DO, and temperature optimal control. Above methods were applied in constructing the glutamic acid concentration soft-sensor model of the fed-batch fermentation for Glutathione production by *Saccharomyces cerevisiae*. According to the mechanism of fermentation procedure, we select the temperature $x_T(i)$, pH value $x_{pH}(i)$, dissolved oxygen concentration $x_{DO}(i)$, CO₂ partial pressure of exhaust gas $x_{CO_2}(i)$, and glutamic acid concentration $x_G(i)$ as the input variables of the glutamic acid concentration soft-sensor model because these parameters are related closely with the glutamic acid concentration, select $x_G(i+1)$ as the output variable, The model of the glutamic acid concentration is shown in Fig. 2.

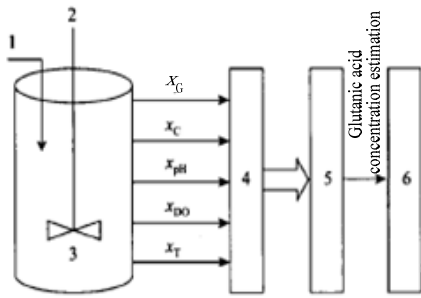


Figure 2. Modeling of glutamic acid concentration
 1-glucose; 2-agitation; 3-fermenter; 4-data process;
 5- glutamic acid concentration soft-sensor; 6-control system

There are 10 batches of production data, each batch express a complete fermentative process. Every batch is divided into 15 samples. 7 batches data as training data and the other regard as test data.

B. Modeling based on SVM

For the reflection fermentative process’s misalignment relations, use the radial direction base nuclear function,

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Therefore the SVM

parameter mainly has the insensitive coefficient ϵ , penalty coefficient C and the breadth factor σ . Uses various sample points relative error's average value (the mean error ratio) the computation model training error and the prediction error. Using different penalty coefficient C , the insensitive coefficient ϵ and the breadth factor σ , the computation training error and the prediction error as shown in Table I.

TABLE I.
 INFLUENCE OF PARAMETER ABOUT MODELING QUALITY

C	ϵ	σ	Training MSE
100	0.5	0.6	0.2355
100	0.5	1	0.2354
100	0.5	10	0.2469
100	0.5	20	0.4575
100	0.5	25	1.8397
100	0.1	0.6	0.0096
100	0.9	0.6	0.7626
100	1.1	0.6	1.1391
100	1.5	0.6	2.1179
100	2.0	0.6	3.7649
50	0.5	0.6	0.2355
40	0.5	0.6	0.2378
30	0.5	0.6	46.5735

From Table I data analysis, it can be found clearly that C is bigger, the penalty is stricter to the error, i.e. to fits the accuracy requirement to be high, causes the training

becomes difficult, very spends the time. Therefore along with C enlargement, the fitting error and the prediction error will reduce, when after C increases to the certain extent, the fitting error tends to be stable, when C is oversized, will present “the fitting” the phenomenon, this time the prediction error instead will increase.

ϵ is the reflection of the noise sensitive degree to the input variable, uses for to control the model fitting precision, ϵ is bigger, the fitting precision is lower, derives the support vector number are few, the model complex degree is low, its pan-ability is strong; Otherwise, ϵ is smaller, the fitting precision is high, supports the vector number to increase, the model order of complexity is high, its pan-ability is low.

When σ is small, the model fitting performance is good, but too small will create the pan-ability variation; Otherwise, σ is too big, supports the vector the influence to be excessively strong, the model achieves the enough precision with difficulty, easy to produce “owes the fitting”.

In order to judge the performance of the model, the mean square error (*MSE*) is selected. When the training $MSE=0.0096$, then parameter $\sigma=0.6$, $C=100$, $\epsilon=0.1$ are selected. The training result of Glutamic acid concentration based on SVM is shown in Fig.3.

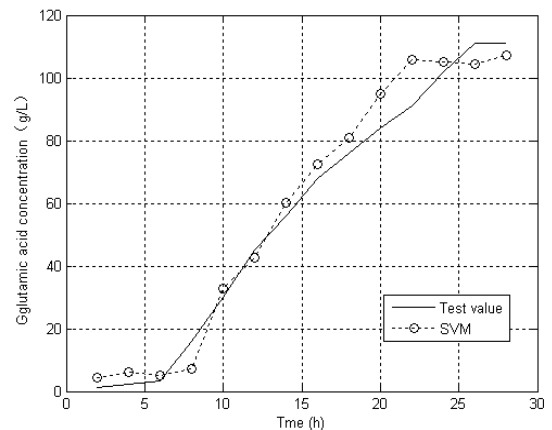


Figure 3. Predicting result of modeling based on SVM

From Figure 3, we can see that the predicting result of generalization of this method is not ideal, so it need to be improved.

C. Modeling based on online SVM

In order to improve the precision and ability of generalization of the soft-sensor, the modified model based on the online SVM is developed. The same inputs and architecture of model are used as that in above section.

One of the more interesting characteristics of Online SVR is stabilization. It corrects floating-point errors, but has another important application: if a parameter changes, with stabilization it is possible to stabilize the SVR without restarting the training. Is it faster to restart the training or use stabilization algorithm? To answer to this question, we prepared a test where, starting from a trained Online SVR, we changed all the parameters and compared the results.

The Table II shows the parameters of the SVR used during the stabilization test.

TABLE II.
TRAINING PARAMETERS OF THE STABLIZATION SET

Training	ϵ	C	σ
Default	0.01	3	50
Increasing C	0.01	10	50
Decreasing C	0.01	1	50
Increasing ϵ	0.1	3	50
Decreasing ϵ	0.0001	3	50
Increasing Kernel Param	0.01	3	100
Decreasing Kernel Param	0.01	3	10

The C parameter influences Support and Error Set. From these results, it seems that it should be used with an increment of speed. When used in an application, it could be changed to avoid overfitting and underfitting. ϵ is more critical than C because it influences all the samples. In fact, the number of iterations needed is higher and, sometimes the algorithm speed is lower than if we were to restart the training.

The detailed process of modeling is as following:

Step 1 Construct a new SVM training set

$X\{X_i = \{x_{T_i}, x_{pH_i}, x_{CO_2}, x_{DO_i}, x_{E_i}\}\}$ have summarized l samples.

Step 2 Use incremental algorithm of online SVR algorithm to train system model.

Step 3 If number of training samples is out of threshold max_sp , use decremental algorithm to remove redundant samples.

Step 4 Compute predicting error $e = x_G(i+1) - x_G(i)$, if e is less than a threshold min_error , end training process, and go to step 5, else return to step 1.

Step 5 Using the model to predict the glutamic acid concentration $x_G(i+1)$.

Step 6 Return to step 4.

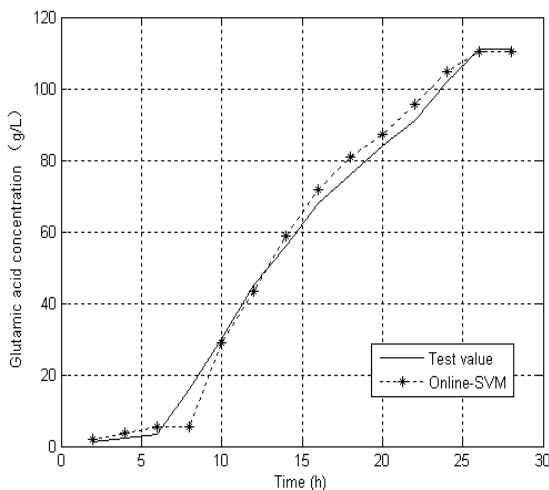


Figure 4. Predicting results of modeling based on online SVM

IV. COMPARING OF RESULTS BETWEEN TWO METHODS

The new proposed modeling of biotechnical process is easy to use and has good real-time processing ability and adaptively which come from the adoption of online SVR algorithm. Here we compare online SVR algorithm-based system identification with offline SVR algorithm for modeling biotechnical process.

A.. Usability

The Comparisons of predicting error between the two models are shown in Fig. 5

Fig. 5 shows that the modeling based on online SVM achieved more excellent capability of generalization than the one based on standard SVM. The performance index of the improved model is: MSE=0.0023. It shows a higher precision of approximation than modeling based on standard SVM.

From comparison of predicting errors, it is concluded that the improved model shows outstanding performance than the one built on the standard SVM because this modified algorithm possesses both advantages of sparseness and robustness.

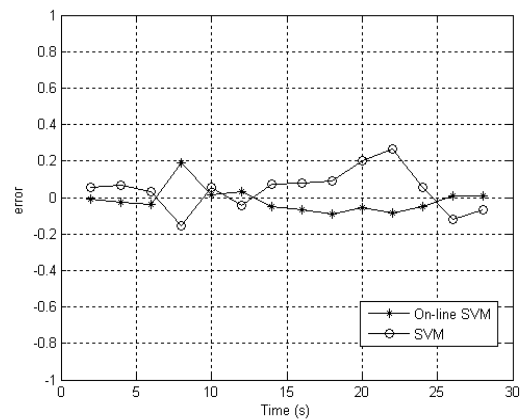


Figure 5. Comparisons of predicting error between two models

Online and offline SVR algorithms are all easy to use. They have uniform structure of model, and only have several adjust able parameters. While the structure of NN must be selected, including the tier number of NN, inner node number of each tier, and initial values of weight coefficients. If bad parameters are selected, it will lead to a NN model with bad performance or very long training time. So SVR algorithms can be more easily used to construct system model than NN.

B. Speed

Speed of online SVR algorithms is faster than that offline one. The comparison of training time and testing time between modeling SVM and online SVM method is shown in Table II.

In MATLAB simulation experiments, training processes of online SVR algorithm as shown in Table II take less than 2.9843 s.

Using SVM MATLAB Toolbox to simulate offline SVR algorithm, the process of 105 training samples takes about 3.5126 s (the kernel algorithm of the toolbox is

written in c code, while our algorithm is written in MATLAB code).

TABLE III.
COMPARING OF THE TRAINING AND TESTING TIME

Algorithm	Training sample number	Training Time (s)	Testing sample number	Testing Time (s)
SVM	105	3.5126	15	0.0150
Online SVM	105	2.9843	15	0.0102

From Table III, it is clearly found that the training and the testing time of modeling based on online SVM are shorter than the SVM. We can see that Online SVR is quicker than SVM in training.

From above analyze, it is illuminated that the modeling based on online SVR for biotechnical process has very good performance.

C. Adaptivity

Online SVR algorithm can automatically track variance of indented plant model, so it can be used to construct dynamic model of time-varying system. Offline SVR algorithm can not be used to construct model of time-varying system.

V. CONCLUSION

Modeling of biotechnical process has received much attention in recent years, and the requirement of the online predicting models is practical in real-world applications. This paper develops an online SVM model to predict the concentration of production, and it can provide promising prediction results.

In order to achieve the aim of on-line estimating the glutamic acid concentration, the method of SVM is introduced as the soft-sensor, and the SVM based model obtains a certain precision and generalization ability. Nevertheless, an improved SVM called online SVM is developed as the soft-sensor model for the fermentation process and better performance is achieved. Compared with the conventional SVM model, it can not only receive data in sequence and determine dynamically the optimal prediction model, but also can lead to high prediction performance; therefore, the applications of the online SVM method in biotechnical process is a good interesting attempt and it may be worthy to test its value in more areas. Additionally, the computational problem in the proposed approach with the numerical optimization in a high-dimensional space may suffer from the course of dimensionality. How to solve this problem will be our future research work.

ACKNOWLEDGMENT

This work is supported by the National High-tech Research and Development of China (863 project) (No.2006AA020301-11). Therefore, it is necessary for the stability conditions to be investigated in the multi-regions.

REFERENCES

- [1] K. Shimizu, K. Furuya, Optimal operation derived by Green's theorem for the cell-recycle filter fermentation focusing on the efficient use of the medium, *Biotechnol. Prog.* 10 (1994), pp. 258–262.
- [2] J.F. Pollard, M.R. Broussard, D.B. Garrison, K.Y. San, Process identification using neural networks, *Comput. Chem. Eng.* 16 (4) (1992), pp. 253–270.
- [3] L.H. Ungar, B.A. Powell, S.N. Kamens, Adaptive networks for fault diagnosis and process control, *Comput. Chem. Eng.* 14 (4/5) (1990), pp.561–572.
- [4] M. Kishimoto, Application of fuzzy logic theory to bioprocesses and its problem, *Biosci. Ind. (Jpn.)* 49 (7) (1991), pp.18–22.
- [5] Olsson, L., Nielson, J., "Online monitoring of biomass in submerged cultivation", *Trends Biotechnol.*, 15(11), (1997), pp.5017-522.
- [6] Wang, Y.J., Fan, Y., "Studies of on-line and in-situ measuring method for biomass concentration", *Prog. Biochem.Biophys.*, 27(4) (2000), pp:387-390. (in Chinese)
- [7] Wang, W., Yang, H.L., Lu. X.F., Yang, S.L., "The application of ultrasonic in fermentation engineering", *J. Wuxi Univ. Light Ind.*, 21(3), (2002), pp.322-326. (in Chinese)
- [8] P.M.L. Drezet, R.F. Harrison, Support vector machines for system identification, in: UKACC International Conference on CONTROL' 98, UK, 1998, pp. 668 - 692.
- [9] K.R. Müller, A.J. Smola, Predicting time series with support vector machines, in: *Proceedings of ICANN'97, Lecture Notes in Computer Science*, vol. 1327, Springer, Berlin, 1997. pp. 999–1004.
- [10] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Neurocolt Technical Report, Royal Holloway College, University of London, 1998.
- [11] J.A.K. Suykens, Nonlinear modeling and support vector machines, In: *IEEE Instrumentation and Measurement Technology Conference*, Budapest, Hungary, 2001, pp. 287–294.
- [12] M. Martin, On-line support vector machines for function approximation, Technical Report LSI-02-11-R, Software Department, Universitat Politècnica de Catalunya, Spain, 2002.
- [13] D.C. Wang, et al., Support vector machines regression on-line modeling and its application, *Control Decision* 18 (1) (2003), pp. 89–91,95.
- [14] Vapnik, V.N., *Statistical Learning Theory*, Wiley, New York (1998).
- [15] Vapnik, V.N., *The Nature of Statistical Learning Theory*, 2nd Edition, Springer-Verlag, New York (1999).
- [16] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.
- [17] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.
- [18] J.S.Ma, T.James, P.Simon, Accurate on-line support vector regression, *Neural Comput.* 15(11)(2003), pp.2683-2704

Xianfang Wang, 1969.08, doctoral student, associate professor; Jiangnan University, China, her research interests include machine learning, modeling and optimized controlling of biotechnical process process, etc. Tel: 13961868919; E-mail: xfwang11@yahoo.com.cn

Zhiyong Du, 1967.05, associate professor; Henan Mechanical and Electrical Engineering College, China, his research interests include machine learning, modeling and optimized controlling.
E-mail: zhydu@126.com

Jindong Chen, 1983.10, doctoral student, Jiangnan University, China, his research fields include modeling and optimized controlling of biotechnical process process, etc.
E-mail: chenjindong417@yahoo.com.cn

Feng Pan, 1963.10, doctor, professor; Jiangnan University, China, his current research interests include machine learning, modelling and control of complex industrial processes, etc. E-mail: pan_feng_63@163.com.