International Journal of Control and Automation
Vol.8, No.3 (2015), pp.171-184
http://dx.doi.org/10.14257/ijca.2015.8.3.19

Detecting Syrup Brix in Cane Sugar Crystallization Using an Improved Least Squares Support Vector Machine

Yanmei Meng, Jian Chen, Guancheng Lu and Zhihong Tang

[1]*Department of Mechanical Engineering, Guangxi University, China*
gxu_mengyun@163.com

Abstract

The cane sugar boiling was a process involved syrup-sugar physical transformation, which had still been operated manually in the sugar production fields across China. The major reason for this was that the syrup brix in the process had not yet been measured accurately on-line. Aimed to solve this issue, an improved LS-SVM based soft sensor model for it was proposed. This model was able to predict the syrup brix based on five auxiliary variables. Since this was a statistical model, the unusual samples in the training set had to be removed, which could be achieved by adding weighting coefficients to the empirical errors of samples. Also, the optimization function of LS-SVM was improved to ensure the generalization capacity of the model. Then, this model was verified in a self-regulating comprehensive experimental platform, which proved that the predicted brix was close to the real brix where the maximal relative error was nearly 2.5%.

Keywords: *Syrup brix, Soft-sensor, LS-SVM*

1. Introduction

The regular cane sugar process can be described as follows: juice squeezing out of canes; cane juice clarification; evaporation and concentration (sugar cane boiling); crystal separation etc. With heat and mass transmission, sugar boiling process is featured with nonlinearity, large inertia, strong coupling and delaying. To achieve automatic sugar production, it was essential to measure key parameters measurement in sugar process [1]. Therefore, a lot of researchers had dedicated their efforts in putting forward various advanced control methods for automation in the cane sugar process field. Nevertheless, even if advanced control methods had been proposed (nonlinear model based predictive control; genetic model control; HN control etc) for both continuous and batch crystallization processes [2-7], the lack of advanced sensors remain a major difficulty to the implementation of relevant control strategies [8].

Syrup brix was one of the most important variables and its measurement determined the efficiency and quality of sugar production. However, the surface of brix sensor was easily accumulated with filth during the process, which made the on-line measurement unstable and discontinuous. In addition, the brix sensor was a very expensive device [9]. Therefore, it was highly demanded to develop a method for on-line brix measurement with stability and affordability.

This paper defined five easy-to-measured auxiliary variables of the sugar process as the inputs, and these variables included vacuum degree, syrup temperature, steam pressure, steam temperature and flow of material. Brix was considered as the output of the model. An improved LS-SVM, called proximal least squares support vector machine (PLS-SVM) was

ISSN: 2005-4297 IJCA

utilized to build the model because it featured with high stability, great generalization and excellent model accuracy.

2. Soft-sensor Algorithm

2.1. LS-SVM based Soft Sensor Algorithm

The sample set of soft sensor was $S=\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where m was the number of the samples, $\mathbf{x}_i=(\mathbf{x}_i=[x_{i1},x_{i2},x_{i3},x_{i4},x_{i5}], x_{i1}\in R, x_{i2}\in R, x_{i3}\in R)$ was a vector that represents the auxiliary variables. Vacuum, syrup temperature, steam pressure, steam temperature and feed flow of the ***i***-th sample were denoted by x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5} respectively. The syrup brix was defined as y_i. According to the principle of LS-SVM, the relationship between syrup brix and the auxiliary variables was described as:

$$y(\mathbf{x}_i)=\mathbf{w}^T\phi(\mathbf{x}_i)+b, b\in R \tag{1}$$

where **w** represented the weight vector, b represented the offset value.

Obviously, the relationship between the syrup brix and the auxiliary variables was nonlinear. In Eq. (1), $\phi(\mathbf{x}_i)$ mapped the nonlinearity to a high dimensional feature space, in a way to turn the nonlinearity into the linear feature space. The LS-SVM based soft sensor algorithm for syrup brix could be expressed as an optimization approach (2) with constraint equation (3), which was defined as follows:

$$F(\mathbf{w},e)=\frac{1}{2}\mathbf{w}^T\mathbf{w}+\frac{1}{2}C\sum_{i=1}^{m}e_i^2 \tag{2}$$

The constraint condition was:

$$y_i=\mathrm{w}^T\phi(\mathbf{x}_i)+b+e_i,\ \mathrm{i}=1,\cdots,m \tag{3}$$

where e_i was the empirical error variable, C denoted the penalty factor that expressed the extent of empirical error. According to Eq. (2) and (3), a Lagrange function was constructed as:

$$\begin{aligned}L(\mathbf{w},b,e,\alpha)&=F(\mathbf{w},e)-\sum_{i=1}^{m}\alpha_i\{\mathbf{w}^T\phi(\mathbf{x}_i)+b+e_i-y_i\}\\&=\frac{1}{2}\mathbf{w}^T\mathbf{w}+\frac{1}{2}C\sum_{i=1}^{m}e_i^2-\sum_{i=1}^{m}\alpha_i\{\mathbf{w}^T\phi(\mathbf{x}_i)+b+e_i-y_i\}\end{aligned} \tag{4}$$

where α_i represented Lagrange multiplier. In order to obtain the optimal solution, the partial derivatives of **w**, b, e, α obtained from $L(\mathbf{w},b,e,\alpha)$ in Eq. (4) were set to equal to 0, which was described as Eq. (5).

$$\begin{cases}\dfrac{\partial L}{\partial \mathbf{w}}=0\rightarrow \mathbf{w}=\sum_{i=1}^{m}\alpha_i\phi(\mathbf{x}_i)\\ \dfrac{\partial L}{\partial b}=0\rightarrow \sum_{i=1}^{m}\alpha_i=0\\ \dfrac{\partial L}{\partial e_i}=0\rightarrow \alpha_i=Ce_i\\ \dfrac{\partial L}{\partial \alpha_i}=0\rightarrow \mathbf{w}^T\phi(\mathbf{x}_i)+b+e_i-y_i=0\end{cases} ,i=1,\cdots,m \tag{5}$$

The inner product of vector in high dimensional space could be replaced by kernel function if it met Mercer conditions [12-13]. Hence, the calculation of the specific form of $\phi(\mathbf{x}_i)$ was avoided, all we needed to do was to solve the kernel function values in the original input space.

The kernel function was set as $K(\mathbf{x}_i,\mathbf{x}_j)$, and we had $\phi(\mathbf{x}_i)\phi(\mathbf{x}_j)=K(\mathbf{x}_i,\mathbf{x}_j)$. The Gaussian RBF $K(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\gamma|x_i-x_j|)$ was selected as the kernel function of the soft sensor algorithm, where γ represented the kernel function parameter. Hence, the optimal solution of Eq. (5) was defined as:

$$\begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K(\mathbf{x}_i,\mathbf{x}_j)+\frac{1}{C} & \cdots & K(\mathbf{x}_i,\mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(\mathbf{x}_m,\mathbf{x}_j) & \cdots & K(\mathbf{x}_m,\mathbf{x}_m)+\frac{1}{C} \end{bmatrix} \times \begin{bmatrix} b \\ \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} 0 \\ y_1 \\ \vdots \\ y_m \end{bmatrix}, i, j=1,\cdots,m \tag{6}$$

It could be solved to obtain the α and b using numerical solution method for linear system of equations. Finally, with Eq. (1) and $\mathbf{w}=\sum_{i=1}^{m}\alpha_i\phi(\mathbf{x}_i)$ in Eq. (5), the relationship between the syrup brix and the auxiliary variables through the soft sensor was described as:

$$\hat{y}=\sum_{i=1}^{n}\alpha_i K(\mathbf{x},\mathbf{x}_i)+b=\sum_{i=1}^{n}\alpha_i \exp(-\gamma|\mathbf{x}-\mathbf{x}_i|^2)+b \tag{7}$$

where n represented the number of support vectors, vector $\mathbf{x}$ represented the auxiliary variables, $\mathbf{x}_i$ was the i-th support vector and y was the syrup brix.

The soft sensor algorithm modeled by the principle as described above could obtain the syrup brix value from the auxiliary variables while it involved several challenges, which were described as follows:

1). The coefficient matrix in Eq. (6) was semi-definite and the its inverse may not exist, which would make its calculation complicated.

2). The penalty factor parameter in every sample was set as the same value for the soft sensor algorithm. Namely, the extent of the empirical error in a sample was consistent with others, and it would compromise the performance of the soft sensor model for the syrup brix because of the unusual samples.

3).γ and C were obtained from the offline optimization while the soft sensor algorithm model development was constructed online, which couldn't dynamically reflect the effects the online samples brought to. The auxiliary variables of soft sensor were nonlinear and the auxiliary variables detected from sensors had uncertain components, which would decrease the accuracy of the soft sensor for syrup brix when the online dynamic feedback was not possible.

2.2. Stability Improvement for the Soft Sensor Algorithm

To improve the stability of soft sensor algorithm, b was input into the optimization objective function to turn the coefficient matrix positive [14], which ensured the calculation of the coefficient matrix much easier. The soft sensor algorithm for the syrup brix with improved optimization objective function was described as follows.

The optimization objective function was:

$$y=\sum_{i=1}^{n}\alpha_i K(\mathbf{x},\mathbf{x}_i)+b=\sum_{i=1}^{n}\alpha_i \exp(-\gamma|\mathbf{x}-\mathbf{x}_i|^2)+b \tag{8}$$

The constrained condition was:

$$y_i = \mathbf{w}^T \phi(\mathbf{x}_i) + b + e_i, i = 1, \cdots, m \tag{9}$$

According to Eq. (8) and (9), a Lagrange function was deduced as:

$$\begin{aligned} L(\mathbf{w}, b, e, \alpha) &= F(\mathbf{w}, e) - \sum_{i=1}^{m} \alpha_i \left\{ \mathbf{w}^T \phi(x_i) + b + e_i - y_i \right\} \\ &= \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}b^2 + \frac{1}{2}C\sum_{i=1}^{m} e_i^2 - \sum_{i=1}^{m} \alpha_i \left\{ \mathbf{w}^T \phi(x_i) + b + e_i - y_i \right\} \end{aligned} \tag{10}$$

The conditions for the optimal solution were described as:

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = 0 \to \mathbf{w} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i) \\ \dfrac{\partial L}{\partial b} = 0 \to b = \sum_{i=1}^{m} \alpha_i \\ \dfrac{\partial L}{\partial e_i} = 0 \to \alpha_i = C e_i \\ \dfrac{\partial L}{\partial \alpha_i} = 0 \to \mathbf{w}^T \phi(\mathbf{x}_i) + b + e_i - y_i = 0 \end{cases} , i = 1, \cdots, m \tag{11}$$

Hence the optimal solutions based on Eq. (11) were as follows:

$$\begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) + 1 + \frac{1}{C} & \cdots & K(\mathbf{x}_1, \mathbf{x}_i) + 1 & \cdots & K(\mathbf{x}_1, \mathbf{x}_m) + 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_i, \mathbf{x}_1) + 1 & \cdots & K(\mathbf{x}_i, \mathbf{x}_i) + 1 + \frac{1}{C} & \cdots & K(\mathbf{x}_i, \mathbf{x}_m) + 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_m, \mathbf{x}_1) + 1 & \cdots & K(\mathbf{x}_m, \mathbf{x}_i) + 1 & \cdots & K(\mathbf{x}_m, \mathbf{x}_m) + 1 + \frac{1}{C} \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_i \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_m \end{bmatrix} , i, j = 1, \cdots, m \tag{12}$$

Finally, with Eq. (1) and xx and xxx in Eq. (11), the relationship between the syrup brix and the auxiliary variables was described as follows:

$$y = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{n} \alpha_i \exp(-\gamma \,|\, \mathbf{x} - \mathbf{x}_i \,|^2) + \sum_{i=1}^{m} \alpha_i \tag{13}$$

With the introduction of improved optimization objective function, the coefficient matrix was positive definite. As we know, every positive definite matrix was reversible. Hence, the solution of Eq. (12) could be obtained smoothly.

2.3. Performance Improvement for Soft Sensor

For the existence of the unusual samples, the soft sensor model for syrup brix was affected by some extent. To solve this problem, the empirical errors were attached with weights [15-16]. The weights attachment was proceeded on the purpose of getting rid of the unusual sample, so as to improve the soft sensor model performance.

The weight coefficient of empirical error was presumed as z_i and Eq. (8) become as follows:

$$F\left(\mathbf{w},e\right)=\frac{1}{2}\mathbf{w}^T\mathbf{w}+\frac{1}{2}b^2+\frac{1}{2}C\sum_{i=1}^{m}z_ie_i^2 \tag{14}$$

According to Eq. (9) and (14), a Lagrange function was formed as:

$$\begin{aligned}L(\mathbf{w},b,e,\alpha)&=F(\mathbf{w},e)-\sum_{i=1}^{m}\alpha_i\left\{\mathbf{w}^T\phi(\mathbf{x}_i)+b+e_i-y_i\right\}\\&=\frac{1}{2}\mathbf{w}^T\mathbf{w}+\frac{1}{2}b^2+\frac{1}{2}C\sum_{i=1}^{m}z_ie_i^2-\sum_{i=1}^{m}\alpha_i\left\{\mathbf{w}^T\phi(\mathbf{x}_i)+b+e_i-y_i\right\}\end{aligned} \tag{15}$$

Then the conditions of the optimal solution were defined as follows:

$$\begin{cases}\dfrac{\partial L}{\partial \mathbf{w}}=0\rightarrow \mathbf{w}=\sum_{i=1}^{m}\alpha_i\phi(\mathbf{x}_i)\\ \dfrac{\partial L}{\partial b}=0\rightarrow b=\sum_{i=1}^{m}\alpha_i\\ \dfrac{\partial L}{\partial e_i}=0\rightarrow \alpha_i=Cz_ie_i\\ \dfrac{\partial L}{\partial \alpha_i}=0\rightarrow \mathbf{w}^T\phi(\mathbf{x}_i)+b+e_i-y_i=0\end{cases} ,i=1,\cdots,m \tag{16}$$

And the optimal solution equations based on Eq. (16) were described as:

$$\begin{bmatrix}K(\mathbf{x}_1,\mathbf{x}_1)+1+\frac{1}{Cz_1} & \cdots & K(\mathbf{x}_1,\mathbf{x}_i)+1 & \cdots & K(\mathbf{x}_1,\mathbf{x}_m)+1\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ K(\mathbf{x}_i,\mathbf{x}_1)+1 & \cdots & K(\mathbf{x}_i,\mathbf{x}_i)+\frac{1}{Cz_i} & \cdots & K(\mathbf{x}_i,\mathbf{x}_m)+1\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ K(\mathbf{x}_m,\mathbf{x}_1)+1 & \cdots & K(\mathbf{x}_m,\mathbf{x}_i)+1 & \cdots & K(\mathbf{x}_m,\mathbf{x}_m)+1+\frac{1}{Cz_m}\end{bmatrix}\times\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_i\\ \vdots\\ \alpha_m\end{bmatrix}=\begin{bmatrix}y_1\\ \vdots\\ y_i\\ \vdots\\ y_m\end{bmatrix},i,j=1,\cdots,m \tag{17}$$

Every absolute empirical error value was obtained as $error_i=|y_i-\hat{y}_i|,1\le i\le m$ using cross validation (CV) method under condition of z_i=1. And the average value of the absolute empirical error was $\overline{error}=\frac{\sum_{i=1}^{m}error_i}{m}$ and the variance was $Var=\frac{\sum_{i=1}^{m}(error_i-\overline{error})^2}{m-1}$. Hence the weight coefficient was $z_i = \begin{cases} 1.0 & ,error_i \le \overline{error} \\ \exp(-\frac{(error_i - \overline{error})^2}{Var}) & ,\overline{error} < error_i \le v \cdot \overline{error} \\ 10^{-4} & ,error_i > v \cdot \overline{error} \end{cases}$, where v could be confirmed based on the distribution of actual samples (in this paper v=1.65). And every weight coefficient would be obtained and substituted into Eq. (17). Finally, with Eq. (1) and $\mathbf{w} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i)$ and $b = \sum_{i=1}^{m} \alpha_i$ in Eq. (16), the syrup brix and the auxiliary variables constructed the following relationship:

$$y = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{n} \alpha_i \exp(-\gamma |\mathbf{x} - \mathbf{x}_i|^2) + \sum_{i=1}^{m} \alpha_i \tag{18}$$

According to the calculation process of weight coefficient, the absolute empirical error value was involved. When the absolute empirical error of a sample deviated from its average value at a larger scale, the probability of the sample being unusual was greater and the weight coefficient became smaller. Before the absolute empirical weight coefficient was attached, the lagrange multiplier with respect to the unusual sample was defined as Ce_i. Then it became Cz_ie_i after the weight coefficient was attached. Based on Eq. (18), when the absolute value of α_i of the support vector became smaller, it had less effect on the soft sensor model. Hence, the empirical error coefficient was able to weaken the influence the unusual sample had on the model. To some extent, it improved the stability and enhanced the performance of the soft sensor model.

2.4. Calculation Process Improvement

The improved soft sensor algorithm turned the coefficient matrix in Eq. (17) positive definite. According to the definition of positive definite, application of a square root method for the solution of Eq. (17) would lead to a higher accuracy level. Its specific solving process was described as follows:

1).

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{im} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mm} \end{bmatrix}$$

$$= \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) + 1 + \frac{1}{Cz_1} & \cdots & K(\mathbf{x}_1, \mathbf{x}_i) + 1 & \cdots & K(\mathbf{x}_1, \mathbf{x}_m) + 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_i, \mathbf{x}_1) + 1 & \cdots & K(\mathbf{x}_i, \mathbf{x}_i) + \frac{1}{Cz_i} & \cdots & K(\mathbf{x}_i, \mathbf{x}_m) + 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_m, \mathbf{x}_1) + 1 & \cdots & K(\mathbf{x}_m, \mathbf{x}_i) + 1 & \cdots & K(\mathbf{x}_m, \mathbf{x}_m) + 1 + \frac{1}{Cz_m} \end{bmatrix}, i, j = 1, \cdots, m \tag{19}$$

where:

$$\mathbf{a}=\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_i\\ \vdots\\ \alpha_m\end{bmatrix}, i,j=1,\cdots,m \qquad \mathbf{q}=\begin{bmatrix}y_1\\ \vdots\\ y_i\\ \vdots\\ y_m\end{bmatrix}, i,j=1,\cdots,m$$

2). Decomposed **A** as **A=LL**^{T}, where $\mathbf{L}=\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{m1} & l_{m2} & \cdots & l_{mm}\end{pmatrix}$. The calculation steps for **L** were described as follows:

Step 1:

$$l_{ii}=\sqrt{a_{ii}-\sum_{k=1}^{i-1} l_{ik}^2} \tag{20}$$

Step 2:

$$l_{ji}=\frac{a_{ji}-\sum_{k=1}^{i-1} l_{jk}l_{ik}}{l_{ii}},\ j=i+1,\ i+2,\ldots,m \tag{21}$$

Step 3: Set **Lp=q**, **L**^{T}**a=p**, where **P** was a temporary vector.

Step 4:

$$p_i=\frac{\left(q_i-\sum_{k=1}^{i-1} l_{ik}p_i\right)}{l_{ii}}, i=1,2,\cdots,n \tag{22}$$

5).

$$\alpha_i=\frac{\left(y_i-\sum_{k=i+1}^{n} l_{ki}\alpha_k\right)}{l_{ii}}, i=n,n-1,\cdots,1 \tag{23}$$

2.5. Improvement for Parameter Optimization

K fold cross validation (K-CV) method was widely used to estimate generalization error. Generally the estimated generalization error was a biased estimation when compared with the actual predicted error. But leave-one-out cross validation (LOO-CV), as one of K-CV methods, could yield an unbiased estimation [15-16]. It took one sample from the sample set as the verification object and trained the rest of sample set repeatedly until all the samples were verified. However, LOO-CV was rather time-consuming and not suitable for on-line estimating generalization error because all samples needed to be verified one by one. References [16-17] proposed an improved LOO-CV method which simplified its own solving process. Based on the improved LOO-CV method, a CV process of parameter optimization applied in the soft sensor algorithm was further improved, making it more suitable for the soft sensor algorithm. The improvement was described as follows.

1). Rewrote the coefficient matrix of system of Eq. (18):

$$\mathbf{A}=\begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{im} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mm} \end{bmatrix}$$

$$=\begin{bmatrix} K(\mathbf{x}_1,\mathbf{x}_1)+1+\frac{1}{Cz_1} & \cdots & K(\mathbf{x}_1,\mathbf{x}_i)+1 & \cdots & K(\mathbf{x}_1,\mathbf{x}_m)+1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_i,\mathbf{x}_1)+1 & \cdots & K(\mathbf{x}_i,\mathbf{x}_i)+\frac{1}{Cz_i} & \cdots & K(\mathbf{x}_i,\mathbf{x}_m)+1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_m,\mathbf{x}_1)+1 & \cdots & K(\mathbf{x}_m,\mathbf{x}_i)+1 & \cdots & K(\mathbf{x}_m,\mathbf{x}_m)+1+\frac{1}{Cz_m} \end{bmatrix} \tag{24}$$

$i,j=1,\cdots,m$

2). Blocked the Eq. (24):

$$\mathbf{A}=\begin{pmatrix} a_{11} & \mathbf{a}_1^T \\ \mathbf{a}_1 & \mathbf{A}_1 \end{pmatrix} \tag{25}$$

where:

$$\mathbf{a}_1=[a_{21},\cdots,a_{i1},\cdots,a_{m1}]^T=[K(\mathbf{x}_2,\mathbf{x}_1)+1,\cdots,K(\mathbf{x}_i,\mathbf{x}_1)+1,\cdots,K(\mathbf{x}_m,\mathbf{x}_1)+1]^T,\ i=2,\cdots,m$$

$$\mathbf{A}_1=\begin{bmatrix} a_{22} & \cdots & a_{2i} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i2} & \cdots & a_{ii} & \cdots & a_{im} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m2} & \cdots & a_{mi} & \cdots & a_{mm} \end{bmatrix},\ i=2,\cdots,m$$

3). Rewrote Eq. (17):

$$p_i=\frac{\left(q_i-\sum_{k=1}^{i-1} l_{ik}\,p_i\right)}{l_{ii}},i=1,2,\cdots,n \tag{26}$$

where $\boldsymbol{\alpha}_{-1}=[\alpha_2,\cdots,\alpha_m]^T$, $\mathbf{y}_{-1}=[y_2,\cdots,y_m]^T$

4). Rewrote Eq. (26):

$$\begin{cases} a_{11}\alpha_1+\mathbf{a}_1^T\boldsymbol{\alpha}_{-1}=y_1 \\ \mathbf{a}_1\alpha_1+\mathbf{A}_1\boldsymbol{\alpha}_{-1}=\mathbf{y}_{-1} \end{cases} \tag{27}$$

5). Took the first sample as test set, the rest as training set and had Eq. (27) go through LOO-CV:

$$\mathbf{A}_1\boldsymbol{\alpha}_{-1}=\mathbf{y}_{-1} \tag{28}$$

$$\boldsymbol{\alpha}_{-1} = \mathbf{A}_1^{-1}\mathbf{y}_{-1} \tag{29}$$

Premultiply both sides of Eq. (29) with $\mathbf{a}_1^T$:

$$\mathbf{a}_1^T\boldsymbol{\alpha}_{-1} = \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{y}_{-1} \tag{30}$$

6). Obtained the syrup brix with the auxiliary variables via Eq. (17):

$$\hat{y}_1 = \sum_{i=2}^{n}\alpha_i K(\mathbf{x},\mathbf{x}_i) + \sum_{i=2}^{m}\alpha_i = \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{y}_{-1} \tag{31}$$

Substituted Eq. (16) into Eq. (31):

$$\begin{aligned}\hat{y}_1 &= \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{y}_{-1} = \mathbf{a}_1^T\mathbf{A}_1^{-1}(\mathbf{a}_1\alpha_1 + \mathbf{A}_1\boldsymbol{\alpha}_{-1}) \\ &= \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1\alpha_1 + \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{A}_1\boldsymbol{\alpha}_{-1} \\ &= \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1\alpha_1 + \mathbf{a}_1^T\boldsymbol{\alpha}_{-1} = \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1\alpha_1 + y_1 - a_{11}\alpha_1 \\ &= y_1 - (a_{11}\alpha_1 - \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1\alpha_1)\end{aligned} \tag{32}$$

7). The inverse of Eq. (25):

$$\mathbf{A}^{-1} = \begin{pmatrix} a_{11} & \mathbf{a}_1^T \\ \mathbf{a}_1 & \mathbf{A}_1 \end{pmatrix}^{-1} = \begin{pmatrix} (a_{11} - \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1)^{-1} & (a_{11} - \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1)^{-1}\mathbf{a}_1\mathbf{A}_1^{-1} \\ \mathbf{A}_1^{-1} + (a_{11} - \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1)^{-1}\mathbf{A}_1^{-1}\mathbf{a}_1^T\mathbf{a}_1\mathbf{A}_1^{-1} & (a_{11} - \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1)^{-1}\mathbf{A}_1^{-1}\mathbf{a}_1^T \end{pmatrix} = \begin{pmatrix} g_{11} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \tag{33}$$

where g_{11} was the element of $\mathbf{A}^{-1}$ in the first row and the first column. **B, C** and **D** were sub matrixes of $\mathbf{A}^{-1}$. The output of the soft sensor model with respect to the first sample was described as:

$$e_1 = y_1 - \hat{y}_1 \tag{34}$$

where y_1 was the syrup brix obtained from the first sample through Eq. (32):

$$y_1 - \hat{y}_1 = (a_{11}\alpha_1 - \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1\alpha_1) \tag{35}$$

Therefore, the test error of the first sample was:

$$\begin{aligned} e_1 &= y_1 - \hat{y}_1 = (a_{11}\alpha_1 - \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1\alpha_1) \\ &= \alpha_1(a_{11} - \mathbf{a}_1^T\mathbf{A}_1^{-1}\mathbf{a}_1) = \alpha_1 g_{11}^{-1} \\ &= \frac{\alpha_1}{g_{11}}\end{aligned} \tag{36}$$

Since the *i*-th sample was defined as test set and the rest was considered as the training set, then the test error of the *i*-th sample was:

$$e_i = \frac{\alpha_i}{g_{ii}} \tag{37}$$

where g_{ii} represented the element of $\mathbf{A}^{-1}$ with respect to the *i*-th row and the *i*-th column.

To sum up, it needed *k* steps to solve Eq. (17) when using K-CV method to optimize parameters of the soft sensor algorithm for syrup brix. And the generalization error obtained was biased estimation of an actual soft sensor error. However, the improved K-CV method only needed one step calculation to obtain the inverse of the coefficient matrix in Eq. (17) and the test error of every sample. Besides, it was easy to carry out the improved K-CV method and its performance was better than the original K-CV method.

2.6. On-line Optimization for Parameters

The on-line optimization method applied in the soft sensor model was used to optimize the kernel function and penalty factor based on on-line samples. The parameters of LS-SVM were optimized quickly by a three-step search (TSS) method [19-23]. A TSS and improved LOO-CV based method was applied to speed up and simplify the calculation of this soft sensor model.

The mean square error (MSE) of the training samples was defined as $MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_1 - y_i)^2$,

where $\hat{y}_1$ and y_i denoted the soft sensor output and the original value. m represented the capacity of the sample dictionary. The auxiliary variables and syrup brix at time t could be put into the sample dictionary. The coefficient matrix in Eq. (17) was described as follows after adding a new sample:

$$\mathbf{K}_t = \begin{bmatrix} K(\mathbf{x}_1,\mathbf{x}_1)+1+\frac{1}{Cz_1} & \cdots & K(\mathbf{x}_1,\mathbf{x}_i)+1 & \cdots & K(\mathbf{x}_1,\mathbf{x}_{m+1})+1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_i,\mathbf{x}_1)+1 & \cdots & K(\mathbf{x}_i,\mathbf{x}_i)+\frac{1}{Cz_i} & \cdots & K(\mathbf{x}_i,\mathbf{x}_{m+1})+1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_{m+1},\mathbf{x}_1)+1 & \cdots & K(\mathbf{x}_{m+1},\mathbf{x}_i)+1 & \cdots & K(\mathbf{x}_{m+1},\mathbf{x}_{m+1})+1+\frac{1}{Cz_{m+1}} \end{bmatrix}, i,j=1,\cdots,m+1 \tag{38}$$

The capacity of the sample dictionary needed to be controlled in order to avoid the limitless expansion. Hence the earliest sample was abandoned and the coefficient matrix became:

$$\mathbf{K}_t = \begin{bmatrix} K(\mathbf{x}_2,\mathbf{x}_2)+1+\frac{1}{Cz_2} & \cdots & K(\mathbf{x}_2,\mathbf{x}_i)+1 & \cdots & K(\mathbf{x}_2,\mathbf{x}_m)+1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_i,\mathbf{x}_2)+1 & \cdots & K(\mathbf{x}_i,\mathbf{x}_i)+\frac{1}{Cz_i} & \cdots & K(\mathbf{x}_i,\mathbf{x}_m)+1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_m,\mathbf{x}_2)+1 & \cdots & K(\mathbf{x}_m,\mathbf{x}_i)+1 & \cdots & K(\mathbf{x}_m,\mathbf{x}_m)+1+\frac{1}{Cz_m} \end{bmatrix}, i,j=1,\cdots,m \tag{39}$$

The MSE of the training samples based on the improved LOO-CV was expressed as $MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{m}\sum_{i=1}^{m}e_i^2 = \frac{1}{m}\sum_{i=1}^{m}(\frac{\alpha_i}{g_{ii}})^2$, where α_i was the Lagrange multiplier with respect to the i-th sample and g was the element of **K** in the i-th row and i-th column.

Then the parameters optimization, soft sensor model learning and development were repeated again based on the on-line samples at time t, in a way to prepare the soft sensor model for the next moment. To simplify the large calculation of the parameters optimization, an improved LOO-CV method with TSS was applied. Details of it were described as followed [17]:

(a)Define the searching accuracy as *precision* and the initial searching step size as *l*. Developed a two dimensional plane with *y* and *C* in the step size of *l* at the last moment.

(b)Calculated the MSE of every point with respect to the relevant parameters based on the improved LOO-CV; Obtain the optimal point on the current plane by taking $\frac{1}{MSE+1}$ as a standard.

(c)Develop a quadrangle via setting the optimal point as center and the 2*l* as the length of the quadrangle; took the four peaks of the quadrangle and the four midpoints of its four sides as the new points; obtained the MSEs of the eight new points; selected the point with the smallest MSE.

(d)If *l* was less greater than *precision*, jumped to (g) step, otherwise proceed the process.

(f)Cut half of *l*, namely, $l=\frac{l}{2}$, then jumped to (c).

(g)Saved the optimal parameters and exit the optimization process.

3. Model Simulation and Experimental Analysis

3.1. Model Simulation

We tested PLS-SVM using different number of samples, where the outputs of samples were taken as actual brix and the outputs of PLS-SVM as the predicted brix. The maximal relative errors (MRE) between actual brix and predicted brix were calculated out as described in Figure 1.

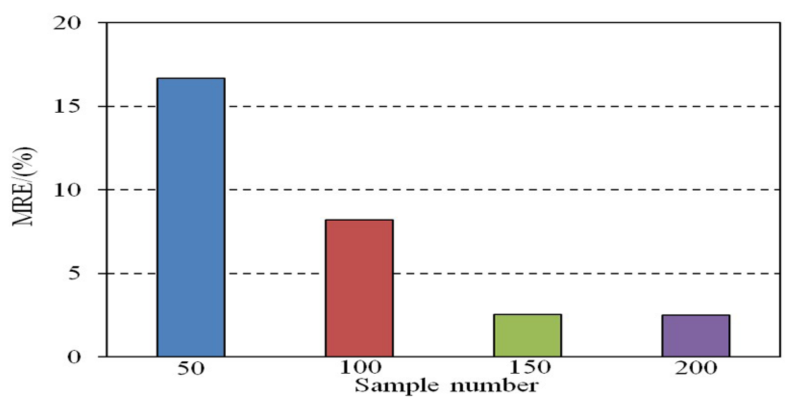

Figure 1. Simulated Brix Comparison under Different Sample Number Condition

The syrup brix predicted by model fall into the acceptable range when the capacity of the sample dictionary exceeded 150. Fig.1 showed that the MRE of the model was nearly 2.5% when the samples were between 150 and 200. Besides, the performance of PLS-SVM was compared with other algorithms (BP, SVM, LS-SVM) with 250 samples, where 200 samples were selected randomly as the training set and the rest (50 samples) was regarded as testing set.

The comparison of the actual brix and the brix predicted by different models with 50 testing samples was depicted in Figure 2. The maximal relative error (MRE) and mean square error (MSE) of each model was given respectively in Figure 3 and Figure 4 from Figure 2. Each model's consuming time was showed in Figure 5.

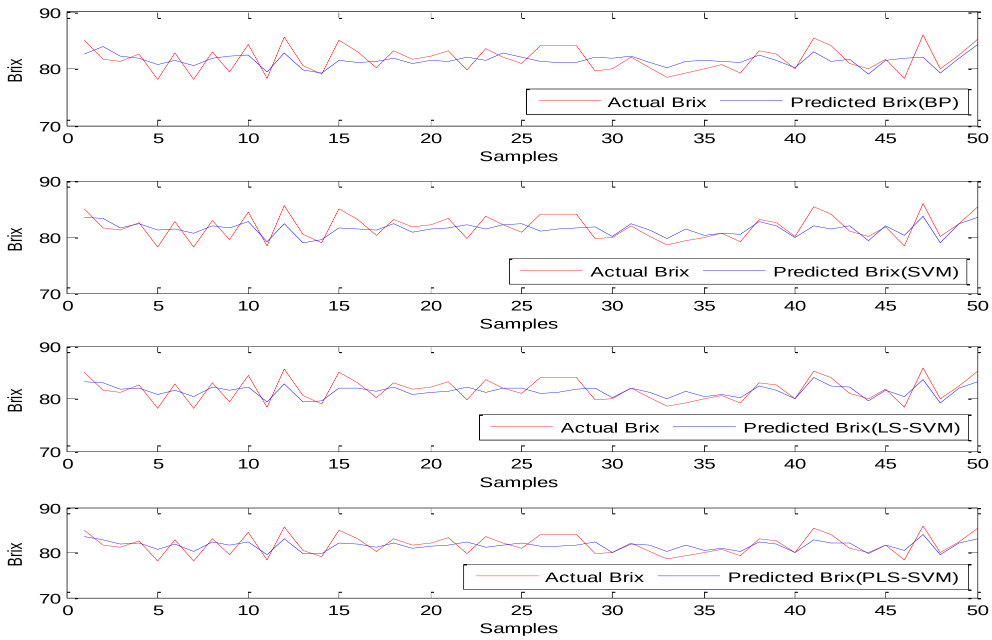

Figure 2. Test Results of Different Algorithms

From Figure 3 to Figure 5, the the MRE and MSE of PLS-SVM were both less than other models, which yielded the MRE and MSE by 2.5% and 1.17% respectively. In addition, the consuming time of PLS-SVM was obviously less than others.

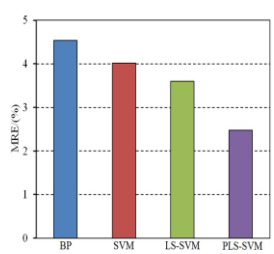

Figure 3. MRE Comparison

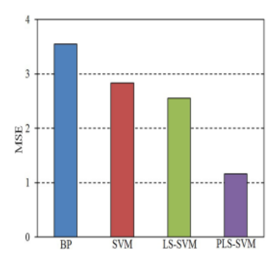

Figure 4. MSE Comparison

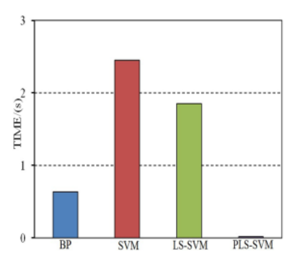

Figure 5. Time Consuming

3.2. Experimental Validation

The soft sensor model was validated on a self-regulating platform, a platform simulated the cane sugar process to an extensive level. And a refractometric brix sensor named MRP E-Scant was installed on the platform for brix measurement. In experiments, every cane sugar process took about 170 minutes, and the probe sensing surface of MRP E-Scant may be accumulated with some adhesive, which would compromise the performance of the sensor. Figiure 7 showed a fraction of the experiment results, in which the value measured by MRP E-Scant online was set as actual brix, the soft sensor model output was defined as predicted brix. It was found in Figure 7 that the MRE was close to 2.5%. The accuracy of MRP E-Scant was largely determined by its own surface. However, the soft sensor model was always feasible, reliable and highly accurate.

Figure 6. Overall Framework of Experimental Platform

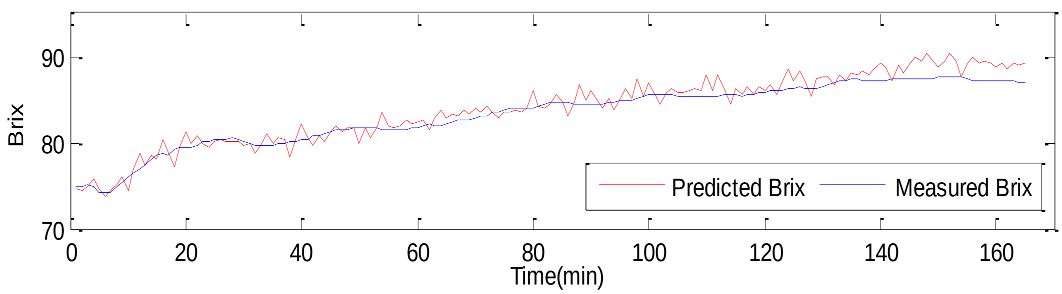

Figure 7. Experimental Test Results

4. Conclusions

We applied an improved LS-SVM in the soft sensor algorithm for predicting syrup brix, in this way to optimize the object function, add weighting to the empirical error and feed the online samples to the model. Also, the model was ensured to obtain solutions by removing the unusual samples, which improved the accuracy and generalization capacity of the model. Besides, a soft sensor system for the syrup brix was developed to complete the offline and online soft sensor under VC++ 6.0 environment. This system showed a good performance when run on the experiment platform.

Acknowledgements

This research is supported by The National Natural Science Foundation of China (No. 51465003), Guangxi Natural Science Foundation (No. 2011GXNSFA018168) and Nanning Scientific and Technological Project (No. 20131079).

References

[1] Y. M. Meng, H. P. He, and F. N. Lu, "Research of a real-time soft sensor system of syrup supersaturation in sugar crystallization process", Journal of Guangxi University (Natural Science Edition), vol. 1, **(2012)**, pp. 1-4.
[2] L. A. P. Suárez, P. Georgieva and F. de Azevedo, "Nonlinear MPC for fed-batch multiple stages sugar crystallization", Chemical Engineering Research and Design, vol. 89, **(2011)**, pp. 753-767.
[3] Q. Hu, S. Rohani, D. X. Wang, and A. Jutan, "Optimal control of a batch cooling seeded crystallizer", Powder Technology, vol. 156, **(2005)**, pp. 170-176.
[4] N. Moldovanyi, B. Lakatos and F. Szeifert, "Model predictive control of MSMPR crystallizers", Journal of Crystal Growth, vol. 275, **(2005)**, pp. 1349-1354.
[5] Z. K. Nagy, "Model based robust control approach for batch crystallization product design", Computers & Chemical Engineering, vol. 33, **(2009)**, pp.1685-1691.
[6] W. Paengjuntuek, A. Arpornwichanop and P. Kittisupakorn, "Product quality improvement of batch crystallizer by a batch-to-batch optimization and nonlinear control approach", Chemical Engineering Journal, vol. 139, **(2008)**, pp. 344-350.
[7] M. Sheikhzadeh, M. Trifkovic and S. Rohani, "Real-time optimal control of an anti-solvent is othermal semi-batch crystallization process", Chemical Engineering Science, vol. 63, **(2008)**, pp. 829-839.
[8] C. Damour, M. Benne, G. P. Brigitte and J. P. Chabriat, "Soft-sensor for industrial sugar crystallization: On-line mass of crystals, concentration and purity measurement", Control Engineering, vol. 18, **(2010)**, pp. 839-844.
[9] G. P. Brigitte, M. Benne and J. P. Chabriat, "Supervision of C crystallisation in Bois Rouge sugar mill using on-line crystal content estimation using synchronous microwave and refractometric brix measurements", Journal of Food Engineering, vol. 76, **(2006)**, pp. 639-645.
[10] J. A. K Suykens, "Least Squares Support Vector Machines", World Scientific, Singapore, **(2009)**.
[11] H. W. Wen, "Partial Least Squares Regression method and its application", National Defence Industry Press, Beijing, **(2008)**.
[12] H. Z. Wang and J. S. Yu, "Statistical learning theory and application of the process control based on support vector machine", Control and Instruments In Chemical Industry, vol. 31, **(2004)**, pp. 1-6.
[13] J. A. K Suykens, L. Lukas, and J. Vandwealle, "Sparse least squares support vecter machines for adaptive communication channel equalization", International Journal of Applied Science and Engineering, vol. 11, **(2005)**, pp. 51-59.
[14] J. Du and S. Y. Liu, "Direct support vector machine for regression", Systems Engineering and Electronics, vol. 31, **(2009)**, pp. 178-181.
[15] Y. G. Fan, P. Li and Z. H. Song, "Dynamic Weighted Least Squares Support Vector Machines', Control and Decision, vol. 21, **(2006)**, pp. 1129-1134.
[16] Y. P. Huang and H. Guo, "Method research of choosing multiple parameters for support vector machine", Fujian Computer, vol. 3, **(2007)**, pp. 53-54.
[17] R. Rosipal, "Kernel partial least squares for nonlinear regression and discrimination", Neural Network World, vol. 13, **(2003)**, pp. 291-300.
[18] L. J. Li, "The study of Modeling Algorithm Based on LS-SVM and Predictive Control Algorithm", M.D. thesis, Zhejiang University, **(2012)**.
[19] Y. Liu, H.Q. Wang and P. Li, "Adaptive local learning based least squares support vector regression with application to online modeling for fermentation processes", Journal of Chemical Industry and Engineering, vol. 2, **(2008)**, pp. 2053-2057.
[20] R. Resipal and L. J. Trejo, "Kernel partial least squares regression in reproducing kernel Hilbert space", Journal of Machine Learning Research, vol. 2, **(2001)**, pp. 91-124.
[21] A. Tenenhaus, A. Giron, E. Viennet, M. Béra, G. Saporta and B. Fertil, "Kernel logistic PLS: A tool supervised nonlinear dimensionality reduction and binary classification", Computational Statistics & Data Analysis, vol. 51, **(2007)**, pp. 4084-4100.
[22] H. Guo, H. P. Liu and L. Wang, "Method for Selecting Parameters of Least Squares Support Vector Machines and Application", Journal of System Simulation, vol. 18, **(2006)**, pp. 2034-2036.
[23] G. C. Lu, "Research of soft sensor for syrup supersaturation in cane sugar process", M.D. thesis, Guangxi University, (**2009)**.