Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 22 (2013) 580 – 589

Procedia Computer Science

17th International Conference in Knowledge-Based and Intelligent Information and Engineering Systems - KES2013

# Adaptive soft sensor model using online support vector regression with time variable and discussion of appropriate parameter settings

Hiromasa Kaneko, Kimito Funatsu*

Department of Chemical System Engineering, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan

**Abstract**

Soft sensors are used in chemical plants to estimate process variables that are difficult to measure online. However, the predictive accuracy of adaptive soft sensor models decreases when sudden process changes occur. An online support vector regression (OSVR) model with a time variable can adapt to rapid changes among process variables. One problem faced by the proposed model is finding appropriate hyperparameters for the OSVR model; we discussed three methods to select parameters based on predictive accuracy and computation time. The proposed method was applied to simulation data and industrial data, and achieved high predictive accuracy when time-varying changes occurred.



*Keywords:* Process control; soft sensor; degradation; online support vector machine; time variable

## 1. Introduction

Soft sensors are widely used to predict process variables that are difficult to measure online [1]. An inferential model is constructed between the variables that are easy to measure online and those that are not, and an objective variable, **y**, can then be predicted using that model. By using soft sensors, the values of **y** can be predicted with a high degree of accuracy.

Use of these sensors is, however, accompanied by some practical difficulties. One of the main difficulties is the degradation of the soft sensor models. The predictive accuracy of the soft sensors tends to decrease gradually for several reasons, including changes in the state of the chemical plant, catalyzing performance loss, and sensor and process drift.

A solution to this soft sensor model degradation problem is highly desirable. To reduce the degradation, the model can be reconstructed with the most recent data. The moving window (MW) model [2,3] and the recursive model [4] are categorized as sequentially updating type models, and the distance-based just-in-time (JIT) model [5], the correlation-based JIT model [6] and the locally-weighted partial least square model [7] are categorized as JIT type models. For example, a MW model is constructed with the data that were measured most recently, while a distance-based JIT model is constructed with data with distances to the prediction data that are smaller than those of other data. The problems of reconstructing a model, such as the incorporation of abnormal data with training data and the increased maintenance costs, have been studied, and a model based on the relationship between the change in **y** over a period of time and the corresponding change in the explanatory variables, **X**, over the same time period, was proposed [8-10]. This model is referred to as a time difference (TD) model. The effects of deterioration with age, such as sensor drift and the gradual changes in the state of the plant, can be handled using a TD model without reconstruction of the model. Models such as the MW, JIT, and TD models that can predict the **y**-values while adapting to the states of the plant are called adaptive models [11].

* Corresponding author. Tel.: +81-3-5841-7751; fax: +81-3-5841-7771.
*E-mail address:* funatsu@chemsys.t.u-tokyo.ac.jp .


doi:10.1016/j.procs.2013.09.138

There are no adaptive models with high predictive ability in all process states, and the prediction accuracy of each adaptive model depends on the process state [14]. Kaneko *et al*. categorized the degradation of the soft sensor model and discussed the characteristics of adaptive models such as the MW, JIT and TD models, based on the classification results, and confirmed their results through the analysis of both numerical simulation data and real industrial data [15]. The predictive abilities of the current MW, JIT and TD models are not entirely sufficient when rapid changes in the slope, i.e. time-varying changes in a process, occur, and novel techniques are thus required to solve this problem.

The online support vector regression (OSVR) [16] model with time variable was proposed to improve the predictive ability of soft sensor models under degradation conditions where the process changes, or the slope changes, are rapid and time-dependent, to handle any nonlinear relationships between **X** and **y**, and to cope with abrupt changes in the process characteristics. A support vector regression (SVR) [17] model is updated efficiently, while considering a variable representing the time in the **X**-variables. However, one of the main problems with the proposed model is the difficulty in finding the appropriate hyperparameters for the OSVR model, and we therefore discuss three methods, i.e. optimization of the predictive accuracy for validation data (OpV), cross-validation using the SVR method (CrV), and the theoretical decision (ThD) method, used to decide the parameters in terms of their predictive accuracy and computation time.

To verify the effectiveness of the proposed method, we analyze the simulation data where the relationships between **X** and **y** change from moment to moment. The performance of the proposed models is compared with that of other, more traditional adaptive models. The proposed method is then applied to industrial polymer process data.

## 2. Method

### *2.1. OSVR*

The SVR method applies a support vector machine (SVM) to regression analysis and can be used to construct nonlinear models by applying a kernel trick along with the SVM. The OSVR method is a method of efficiently updating a SVR model to meet the Karush-Kuhn-Tucker (KKT) conditions that the SVR model must fulfill when training data are added or deleted.

The primal form of the SVR can be shown to be the following optimization problem.

Minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\left|y_i - f\left(\mathbf{x}_i\right)\right|_{\varepsilon}, \tag{1}$$

where $y_i$, and $\mathbf{x}_i$ are training data, $f$ is the SVR model, $\mathbf{w}$ is a weight vector, $\varepsilon$ is a threshold, $N$ is the number of training data, and $C$ is a penalizing factor that controls the trade-off between the model complexity and the training errors. The second term of Eq. (1) is the $\varepsilon$-insensitive loss function and is given as follows:

$$\left|y_i - f\left(\mathbf{x}_i\right)\right|_{\varepsilon} = \max\left(0, \left|y_i - f\left(\mathbf{x}_i\right)\right| - \varepsilon\right) . \tag{2}$$

By minimization of Eq. (1), we can construct a regression model with a good balance between its generalization capabilities and its ability to adapt to the training data. A **y**-value predicted by inputting data **x** is represented as follows:

$$f\left(\mathbf{x}\right) = \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right)K\left(\mathbf{x}_i, \mathbf{x}\right) + b, \tag{3}$$

where $N$ is the number of training data, $b$ is a constant and $K$ is a kernel function. The kernel function in our application is a radial basis function:

$$K\left(\mathbf{x}_i, \mathbf{x}\right) = \exp\left(-\gamma\left\|\mathbf{x}_i - \mathbf{x}\right\|^2\right), \tag{4}$$

where $\gamma$ is a tuning parameter that controls the width of the kernel function. From Eqs. (1) and (2), $\alpha_i$ and $\alpha_i^*$ in Eq. (3) can be obtained by minimizing the equation given as:

$$\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}K_{ij}\left(\alpha_i-\alpha_i^*\right)\left(\alpha_j-\alpha_j^*\right)-\sum_{i=1}^{N}y_i\left(\alpha_i-\alpha_i^*\right)+\varepsilon\sum_{i=1}^{N}\left(\alpha_i+\alpha_i^*\right), \tag{5}$$

subject to

$$0\le\alpha_i,\alpha_i^*\le C \quad i=1,2,...,N, \tag{6}$$

$$\sum_{i=1}^{N}\left(\alpha_i-\alpha_i^*\right)=0. \tag{7}$$

$K_{ij}$ in Eq. (5) is represented as follows:

$$K_{ij}=K\left(\mathbf{x}_i,\mathbf{x}_j\right). \tag{8}$$

Next, we define $\theta_i$ as follows:

$$\theta_i=\alpha_i-\alpha_i^*. \tag{9}$$

From Eqs. (3), (4) and (8), the predicted **y**-value of data $\mathbf{x}_i$ is given as:

$$f\left(\mathbf{x}_i\right)=\sum_{j=1}^{N}K_{ij}\theta_j+b, \tag{10}$$

where $\theta_i$ satisfies the following equation:

$$\sum_{i=1}^{N}\theta_i=0. \tag{11}$$

The error function $h$ is defined as:

$$\begin{aligned}h\left(\mathbf{x}_i\right)&\equiv f\left(\mathbf{x}_i\right)-y_i\\&=\sum_{j=1}^{N}K_{ij}\theta_j+b-y_i\end{aligned}. \tag{12}$$

The KKT conditions can then be summarized as follows:

$$h\left(\mathbf{x}_i\right)>\varepsilon, \qquad \theta_i=-C, \tag{13}$$

$$h\left(\mathbf{x}_i\right)=\varepsilon, \qquad -C<\theta_i<0, \tag{14}$$

$$-\varepsilon<h\left(\mathbf{x}_i\right)<\varepsilon, \quad \theta_i=0, \tag{15}$$

$$h\left(\mathbf{x}_i\right)=-\varepsilon, \qquad 0<\theta_i<C, \tag{16}$$

$$h\left(\mathbf{x}_i\right)<-\varepsilon, \qquad \theta_i=C. \tag{17}$$

Each training datum must satisfy one of Eqs. (13)-(17). All training data can be divided into the following three sets: error support vectors (E), which satisfy Eq. (13) or Eq. (17); margin support vectors (S), which satisfy Eq. (14) or Eq. (16); and the remaining vectors (R), which satisfy Eq. (15).

When new data in the form $\mathbf{x}_c$, $y_c$ are added, there is no need to update $\theta_i$ and $b$ in the SVR model if $\mathbf{x}_c$ belongs to R. However, if $\mathbf{x}_c$ belongs to E or S, the initial value of $\theta_c$, which is $\theta_i$ corresponding to $\mathbf{x}_c$, is set as 0, and $\theta_c$, $\theta_i$ and $b$ are gradually changed to meet the KKT conditions. There is the possibility that each training datum moves to another region because of the changes, but, assuming no such movements, the variations of $h(\mathbf{x}_i)$, $\theta_c$, $\theta_i$ and $b$, and of $\Delta h(\mathbf{x}_i)$, $\Delta\theta_c$, $\Delta\theta_i$ and $\Delta b$, can be represented using Eqs. (11) and (12) , respectively, as follows:

$$\Delta h\left(\mathbf{x}_i\right) = K_{ic}\Delta\theta_c + \sum_{j=1}^{N} K_{ij}\Delta\theta_j + \Delta b\,, \tag{18}$$

$$\Delta\theta_c + \sum_{j=1}^{N}\Delta\theta_j = 0\,. \tag{19}$$

The $\theta_i$-values of the training data belonging to E and R did not change based on Eqs. (13), (15) and (17), and thus, Eq. (18) can be transformed as:

$$\Delta h\left(\mathbf{x}_i\right) = K_{ic}\Delta\theta_c + \sum_{j\in\mathrm{S}} K_{ij}\Delta\theta_j + \Delta b\,. \tag{20}$$

The $h(\mathbf{x}_i)$-values of the training data belonging to S are settled using Eqs. (14) and (16). Thus, Eqs. (19) and (20) can be changed to:

$$\sum_{j\in\mathrm{S}} K_{ij}\Delta\theta_j + \Delta b = -K_{ic}\Delta\theta_c \qquad \forall i\in\mathrm{S}\,, \tag{21}$$

$$\sum_{j\in\mathrm{S}}\Delta\theta_j = -\Delta\theta_c\,. \tag{22}$$

Then, $\Delta\theta_c$, $\Delta\theta_i$ and $\Delta b$ can be represented by:

$$\Delta b = \delta\Delta\theta_c\,, \tag{23}$$

$$\Delta\theta_i = \delta_i\Delta\theta_c \quad \forall i\in\mathrm{S}\,, \tag{24}$$

where

$$\begin{bmatrix}\delta\\ \delta_{\mathrm{S}_1}\\ \vdots\\ \delta_{\mathrm{S}_M}\end{bmatrix} = -\begin{bmatrix}0 & 1 & \cdots & 1\\ 1 & K_{\mathrm{S}_1\mathrm{S}_1} & \cdots & K_{\mathrm{S}_1\mathrm{S}_M}\\ \vdots & \vdots & \ddots & \vdots\\ 1 & K_{\mathrm{S}_M\mathrm{S}_1} & \cdots & K_{\mathrm{S}_M\mathrm{S}_M}\end{bmatrix}^{-1}\begin{bmatrix}1\\ K_{\mathrm{S}_1 c}\\ \vdots\\ K_{\mathrm{S}_M c}\end{bmatrix}, \tag{25}$$

$$\delta_i = 0 \quad \forall i\notin\mathrm{S}\,. \tag{26}$$

Here, $M$ is the number of training data that belong to S. From Eqs. (20), (23) and (24), $h(\mathbf{x}_i)$ for the training data belonging to E and R can be transformed into:

$$\begin{aligned}\Delta h(\mathbf{x}_i) &= K_{ic}\Delta\theta_c + \sum_{j\in \mathrm{S}} K_{ij}\Delta\theta_j + \Delta b \\ &= K_{ic}\Delta\theta_c + \sum_{j\in \mathrm{S}} K_{ij}\delta_j\Delta\theta_c + \delta\Delta\theta_c \\ &= \left( K_{ic} + \sum_{j\in \mathrm{S}} K_{ij}\delta_j + \delta \right)\Delta\theta_c \\ &= \gamma\Delta\theta_c\end{aligned}, \tag{27}$$

where

$$\gamma = K_{ic} + \sum_{j=1}^{N} K_{ij}\delta_j + \delta\,. \tag{28}$$

From Eqs. (24) and (27), $\Delta\theta_c$ for the movement of each training datum is represented by:

$$\Delta\theta_c = \delta_i^{-1}\Delta\theta_i \quad \forall i \in \mathrm{S}, \tag{29}$$

$$\Delta\theta_c = \gamma^{-1}\Delta h(\mathbf{x}_i) \quad \forall i \notin \mathrm{S}. \tag{30}$$

The absolute $\Delta\theta_i$-values for each training datum to move from their current region to another region, i.e. from E to S, from S to E or R and from R to S, are calculated using Eqs. (29) and (30). The minimum value of the absolute $\Delta\theta_i$-values calculated with all training data is selected, and the data with the minimum $\Delta\theta_i$-value are actually moved to a new region. The calculations of the absolute $\Delta\theta_i$-values and the movement of the data with the minimum of the absolute $\Delta\theta_i$-values are repeated until all the training data satisfy the KKT conditions, i.e., one of Eqs. (13)-(17). When one datum is deleted from the training data, the same iterative calculation is performed until all the data satisfy the KKT conditions.

### *2.2. Time variable*

By updating a nonlinear regression model, the model will be able to handle new nonlinear relationships between **X** and **y**. However, even the updated model cannot adapt to time-varying process changes. Therefore, a variable representing the process time is added to the **X**-variables. By modeling the changes in the process characteristics with this time variable, the model will be able to predict the subsequent **y**-values. In addition, even when the changes in the process are nonlinear in terms of time, an updated nonlinear model with the time variable can be adaptive for these changes.

The time variable is an interval scale and the zero point of the scale can be set arbitrarily.

### *2.3. Parameter setting for OSVR*

For OSVR, the hyperparameters $C$, $\varepsilon$ and $\gamma$ must be set first. In this paper, we discuss the three methods mentioned earlier, i.e. the OpV, CrV and ThD methods, that may be used to determine the parameters in terms of their predictive accuracy and computational time.

#### 2.3.1 Optimization of predictive accuracy for validation data (OpV)

When the number of training data is $N$, the window size, which means the number of data used for the model construction, was set as $m$, and the hyperparameters are selected to provide the best prediction performance based on the last ($N$–$m$) training data. The predictive ability of an OSVR model for new data can be considered as part of the parameter setting process, but a lot of time is required because the OSVR model must be updated many times.

#### 2.3.2 Cross validation using the SVR method (CrV)

SVR parameters optimized by cross validation (CV) on training data were used as the parameters of an OSVR model. CV is a technique that evaluates the predictive ability of statistical models by using only the training data. In an *M*-fold CV, the training data are randomly divided into *M* groups, in which the numbers of data are the same as far as possible. One group is then used as the data to validate the model constructed using the data from the other (*M*–1) groups. This procedure is repeated *M* times so that the data of each of the *M* groups are used once as the validation data. Finally, estimated rather than calculated values of **y** can be obtained. In this paper, *M* is set as 5, which is a standard value. The predictive ability of the SVR model can be considered in the parameter setting process, and the hyperparameters can be selected in reasonable time.

2.3.3 Theoretical decision (ThD)

Cherkassky and Ma proposed the theoretical decision method to decide the hyperparameters $C$ and $\varepsilon$ of a SVR model [18] and we use this method to set the OSVR hyperparameters. The values of $C$ and $\varepsilon$ are calculated as follows:

$$C = \max\left(\left|\mu_{\mathrm{y}} + 3\sigma_{\mathrm{y}}\right|, \left|\mu_{\mathrm{y}} - 3\sigma_{\mathrm{y}}\right|\right), \tag{31}$$

$$\varepsilon = 3\sigma\sqrt{\frac{\ln N}{N}}\,. \tag{32}$$

Here, $\mu_{\mathrm{y}}$ is the average of **y** and $\sigma_{\mathrm{y}}$ is the standard deviation of **y**. $\sigma$ in Eq. (32) is estimated with the *k*-nearest-neighbor's method as follows:

$$\sigma^2 = \frac{N^{1/5}k}{N^{1/5}k - 1} \cdot \frac{1}{n}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right). \tag{33}$$

Here $y_i$ with the marker above it is an estimated **y**-value. Only the hyperparameter $\gamma$ in the kernel function is optimized with CV on training data. This theoretical method takes less time to perform than the other methods.

## 3. Results and discussion

To verify the effectiveness of the proposed method, we analyzed the simulation data along with industrial polymer process data. The relationships between **X** and **y** change from moment to moment for the simulation data. The models to be compared were as follows:

- TDPLS: TD model constructed using the partial least squares (PLS) method [8,19]
- TDSVR: Nonlinear TD model constructed using the SVR method [10]
- JITPLS: PLS model constructed using data with Euclidian distances to new data that are smaller than those of the other data
- LWPLS: PLS model constructed by weighting data with a similarity matrix [20]
- MWPLS: PLS model constructed with the most recently measured data
- OSVR: Updated SVR model

Each model, including its time variable, was also used in the case studies. For OSVR, the OpV, CrV and ThD methods were used to decide on the hyperparameters.

### *3.1. Simulation data in which the relationship between X and y is time-varying*

The number of **X**-variables was set as two. First, $\mathbf{x}_1$ and $\mathbf{x}_2$ of the uniform pseudorandom numbers with ranges from 0 to 10 were prepared. Then, **y** was set as follows:

$$\mathbf{y} = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2\end{bmatrix}\begin{bmatrix}1 & b_2\end{bmatrix}^{\mathrm{T}} + \mathrm{N}(0, 0.1)\,, \tag{34}$$

where $b_2$ is the magnitude of the contribution of $\mathbf{x}_2$ to **y** and N(0, 0.1) represents random numbers from a normal distribution given a standard deviation of 0.1 and a mean of 0. We set $b_2$ as follows:

$$b_2 = 3\sin\left(0.01\pi t\right)+1, \tag{35}$$

$$b_2 = 3\sin\left(0.02\pi t\right)+1. \tag{36}$$

Eqs. (35) and (36) represent the gradual and rapid changes in $b_2$, respectively. $t$ was set as 1, 2, ..., 200 and the number of data was 200. The first 100 data were used for training and the next 100 data were used as the test data. This data set is the same as that of the "*Change of the Slope*" set in reference [15].

The hyperparameter for the LWPLS model was selected with five-fold cross-validation. The window sizes were set as 20 and 50 for the JITPLS, MWPLS and OSVR models. Table 1 shows the prediction results for the test data when only $\mathbf{x}_1$ and $\mathbf{x}_2$ were used as **X**-variables and the time variable was not used. The $r_{pred}^2$ parameter is the determination coefficient for the test data and can have negative values for the test data. The higher the $r_{pred}^2$-value of a model is, then the greater that model's predictive accuracy becomes. The $RMSE_P$ is the root-mean-square error for the test data. Lower $RMSE_P$-values of models indicate higher prediction accuracy in the models. In both cases of $b_2$ based on Eqs. (35) and (36), the prediction abilities of the TDPLS, TDSVR, JITPLS, and LWPLS models were low, because the $r_{pred}^2$-value was low and the $RMSE_P$-value was high for each model. These models could not adapt to the time-dependent process characteristics. The prediction results of the MWPLS models were better than those of the other models when the window size was 20, but, nevertheless, the $r_{pred}^2$-values were low and the $RMSE_P$-values were high for the MWPLS models. None of these models could accurately adapt to the time-varying changes in the relationships between **X** and **y**.

The prediction results are shown in Table 2 when $t$ was added to the **X**-variables ($\mathbf{x}_1$ and $\mathbf{x}_2$) as the time variable. In TDPLS, TDSVR, JITPLS and LWPLS modeling, some of the prediction results improved in comparison with the results without the time variable, but these improvements were still not enough. In both cases of $b_2$, the $r_{pred}^2$-values increased greatly and the $RMSE_P$-values decreased significantly when using the MWPLS models with a window size of 20 with the time variable. Furthermore, a significant improvement in the predictive ability of the OSVR models was confirmed by the addition of the time variable when the hyperparameters were decided based on OpV or CrV. When the number of the window size increased, then the predictive accuracy of the OSVR models with hyperparameters that were selected using CrV also increased. This is because the window size number became close to that of the training data with which the hyperparameters of the SVR model were selected using cross-validation. However, the OSVR models of the hyperparameters set using ThD did not have a high prediction performance, meaning that the hyperparameters should be selected by considering the predictive accuracy of the model. By setting the hyperparameters appropriately, the proposed models could then deal appropriately with the time-varying changes in the relationship between **X** and **y** by updating the SVR models and adding the time variable to the **X**-variables.

Fig. 1 shows the relationships between the simulated and predicted **y** values for the test data. From Fig. 1a and 1b, we can see that the data came close to the diagonal and that the prediction accuracy improved when we added the time variable

Table 1. Prediction results using the simulation data without the time variable.

| | $b_2$: Eq. (35) | | $b_2$: Eq. (36) | |
|---|---|---|---|---|
| Model | $r_{pred}^2$ | $RMSE_P$ | $r_{pred}^2$ | $RMSE_P$ |
| TDPLS | -6.1 | 15 | 0.376 | 9.0 |
| TDSVR | -9.9 | 19 | -0.19 | 12 |
| JITPLS, 20* | -6.0 | 15 | 0.009 | 11 |
| JITPLS, 50* | -6.0 | 15 | 0.053 | 11 |
| LWPLS | -5.9 | 15 | 0.045 | 11 |
| MWPLS, 20* | 0.546 | 3.8 | 0.574 | 7.3 |
| MWPLS, 50* | -1.3 | 8.7 | -0.44 | 13 |
| OSVR, OpV, 20* | 0.498 | 4.1 | 0.536 | 7.6 |
| OSVR, OpV, 50* | -1.1 | 8.3 | -0.20 | 12 |
| OSVR, CrV, 20* | 0.283 | 4.8 | 0.474 | 8.1 |
| OSVR, CrV, 50* | -1.5 | 9.1 | -3.6 | 13 |
| OSVR, ThD, 20* | 0.473 | 4.2 | 0.413 | 8.6 |
| OSVR, ThD, 50* | -1.2 | 8.5 | -0.30 | 13 |

* The number of the window size.

Table 2. Prediction results using the simulation data with the time variable.

| | $b_2$: Eq. (35) | | $b_2$: Eq. (36) | |
|---|---|---|---|---|
| Model | $r_{pred}^2$ | $RMSE_P$ | $r_{pred}^2$ | $RMSE_P$ |
| TDPLS | -6.1 | 15 | 0.376 | 9.0 |
| TDSVR | -19.2 | 26 | -20.1 | 52 |
| JITPLS, 20* | 0.407 | 4.4 | 0.261 | 9.6 |
| JITPLS, 50* | -0.25 | 6.4 | -0.19 | 12 |
| LWPLS | -0.30 | 6.5 | 0.265 | 9.6 |
| MWPLS, 20* | 0.848 | 2.2 | 0.848 | 4.4 |
| MWPLS, 50* | 0.147 | 5.3 | 0.239 | 9.8 |
| OSVR, OpV, 20* | 0.996 | 0.35 | 0.994 | 0.85 |
| OSVR, OpV, 50* | 0.995 | 0.42 | 0.993 | 0.93 |
| OSVR, CrV, 20* | 0.988 | 0.62 | 0.968 | 2.0 |
| OSVR, CrV, 50* | 0.997 | 0.29 | 0.991 | 1.0 |
| OSVR, ThD, 20* | 0.792 | 2.6 | 0.735 | 5.8 |
| OSVR, ThD, 50* | 0.723 | 3.0 | 0.698 | 6.2 |

* The number of the window size.

to the **X**-variables. Though relatively many data are far from the diagonal for the OSVR models with hyperparameters that were decided using ThD (Fig. 1d), by using OpV and CrV for selection of the hyperparameters, the plots show tight clusters of predicted values along the diagonals, meaning that the prediction accuracy of the models was suitably high. It was confirmed that the proposed models were able to deal with the sudden and time-varying changes in the processes and that the appropriate hyperparameters can be decided using CrV, which requires less computational time than OpV.

### *3.2. Application to Industrial Polymer Processes*

We applied the proposed methods to actual industrial data obtained from an industrial polymer process at Mitsui Chemicals, Inc., which is one of the largest Japanese chemical companies, to verify their prediction abilities. Industrial polymer processes generally produce many grades of products. Therefore, when the polymer grade changes, it is important to reduce the quantity of off-grade material produced. Thus, early and accurate judgment of whether the polymer quality is within the given specification limits or not is made using soft sensors, because it is impossible to perform online measurement of large numbers of polymer quality variables when using hard sensors. Construction of models with high prediction performance is difficult in the present circumstances. One reason for this is that there can be a nonlinear relationship between a polymer quality variable, **y**, and another process variable, **X** [21,22]. In addition, impurities can be mixed in with the raw materials and the relationships between **X** and **y** will depend on the amounts of these impurities.

We tried to construct nonlinear models between **X** and **y** for various polymer grades. The constructed models can be

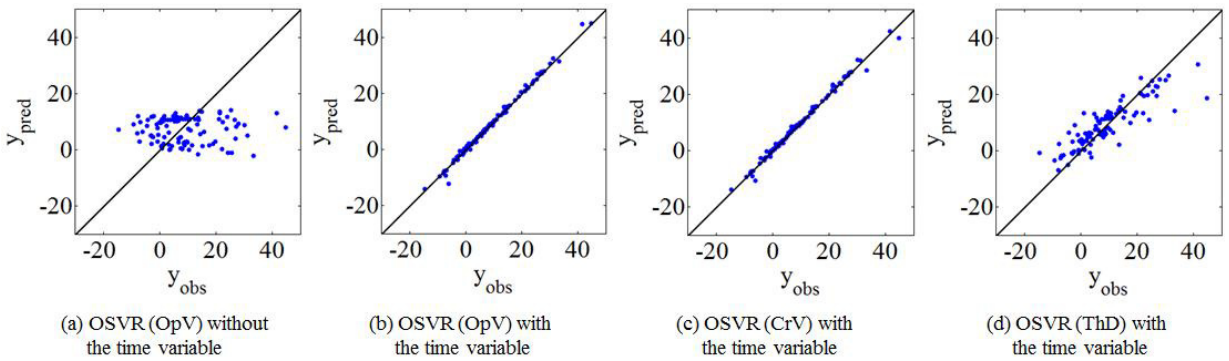


Fig. 1. Relationships between **y** and predicted **y** of test data when Eq. (36) was used as $b_2$. The window size number is 50 in each case.

applied to data from any kind of polymer grade if the relationship between **X** and **y** can be extrapolated to fit those data. Data measured in the steady state for many grades were collected in this study. The **y**-variable represents the density, and the **X**-variables represent 38 different variables, including the temperature in the reactor and the pressure and concentration of the monomer, the comonomer, and hydrogen. We used the data that were monitored from January 2005 to April 2007 as the training data, and data that were monitored from May 2007 to May 2008 as the test data.

The density prediction results are shown in Table 3. In this study, the window size was relatively large and was set as 300 for the JITPLS, MWPLS and OSVR models because of the large numbers of **X**-variables. The OSVR with OpV was not performed because it took too much computation time, but the OSVR with CrV should have almost the same performance as that of OpV, based on the results of section 3.1. In fact, the proposed OSVR model with the time variable achieved the highest density prediction performance when the hyperparameters were selected with CrV. The results of the proposed model were superior to those of the OSVR models without the time variable, indicating that the relationship between the **X**-variables and the density is time-varying, based on the results of the analysis with the simulation data. The relationship between the **X**-variables and the density may be time-varying and may also depend on the amount of impurities in the raw materials.

Fig. 2 shows the relationships between the measured and predicted density values with the test data. The plots of Fig. 2a, 2b and 2d showed the negative bias of the prediction errors when the actual density values range from 0.94 to 0.95 for the OSVR models without the time variable and the OSVR models with parameters that were set using ThD with the time variable. When we used the OSVR model with parameters that were selected using CrV with the time variable, this bias did not exist and much tighter clusters of the predicted values along the diagonal could be seen, reflecting the more accurate prediction of density. It was confirmed that the proposed model can adapt accurately to the nonlinear relationships between **X** and **y** and to the time-varying changes in the relationships, i.e. the changes in the process characteristics.

The hyperparameters and window sizes of the OSVR models were fixed for the predictions in all the case studies. However, the appropriate parameters must vary, depending on the states in a process. Therefore, further improvements in prediction accuracy of OSVR modeling will be achieved by selecting the appropriate parameters in response to the process characteristics.

Table 3. The density prediction results.

| | Without the time variable | | With the time variable | |
|---|---|---|---|---|
| Model | $r_{pred}^2$ | $RMSE_P(\times 10^{-3})$ | $r_{pred}^2$ | $RMSE_P(\times 10^{-3})$ |
| TDPLS | 0.933 | 3.12 | 0.933 | 3.12 |
| TDSVR | 0.960 | 2.43 | 0.954 | 2.58 |
| JITPLS, 300* | 0.960 | 2.41 | 0.960 | 2.44 |
| LWPLS | 0.968 | 2.15 | 0.962 | 2.37 |
| MWPLS, 300* | 0.960 | 2.42 | 0.959 | 2.45 |
| OSVR, CrV, 300* | 0.956 | 2.54 | 0.976 | 1.88 |
| OSVR, ThD, 300* | 0.962 | 2.36 | 0.960 | 2.43 |

* The number of the window size.

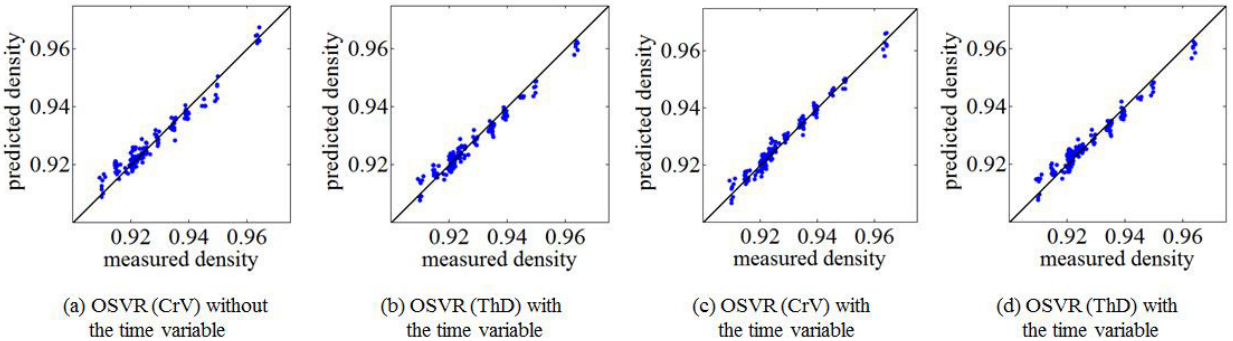


Fig. 2. Relationships between measured and predicted densities with test data.

## 4. Conclusion

In this paper, to accurately predict **y**-values even when the changes in the relationship between **X** and **y** are both nonlinear and time-varying, we proposed a new adaptive soft sensor model combining the OSVR method with the time variable. By adding the time variable to the **X**-variables and efficiently updating the SVR model, the proposed model can adapt to abrupt changes in the process characteristics. We also discussed three methods, i.e. OpV, CrV, and ThD, that were used to decide the parameters in terms of their predictive accuracy and their computation time.

Through the analysis of the simulation data, it was confirmed that the OSVR models with the time variable showed good prediction performance when the relationship between **X** and **y** changed from moment to moment. The OSVR hyperparameters could also be selected using CrV given a reasonable length of time. Then, the superiority of the proposed method was confirmed through the analysis of real industrial data. The appropriate selection of parameters for an OSVR model such as the window size depends on the process states, and ensemble prediction using multiple OSVR models will be able to improve the prediction accuracy.

The main issue with the proposed method is that the updated model specializes in predictions over a narrow data range when there is little variation in a process over a long period, because it is intended for sequentially updating models and JIT models. Appropriate database control is required to solve this issue.

We believe that by applying our proposed method to process control and adapting it to the process characteristics for highly accurate prediction, we will be able to operate chemical plants effectively and stably.

## Acknowledgements

H. Kaneko is grateful for the financial support of the Japan Society for the Promotion of Science (JSPS) through a Grant-in-Aid for Young Scientists (B) (grant no. 24760629). The authors also wish to acknowledge the support of Mitsui Chemicals, Inc., and the financial support of the Mizuho Foundation for the Promotion of Sciences.

## References

[1] Kadlec P, Gabrys B, Strandt S. Data-driven soft sensors in the process industry. *Comput Chem Eng*. 2009;**33**:795–814.
[2] Kaneko H, Arakawa M, Funatsu K. Development of a new soft sensor method using independent component analysis and partial least squares. AIChE J. 2009;55:87–98.
[3] Kadlec P, Gabrys B. Local learning-based adaptive soft sensor for catalyst activation prediction, AIChE J. 2010;57:1288–1301.
[4] Qin SJ. Recursive PLS algorithms for adaptive data modeling. Comput. Chem. Eng. 1998;22:503–514.
[5] Cheng C, Chiu MS. A new data-based methodology for nonlinear process modeling. Chem. Eng. Sci. 2004;59:2801–2810.
[6] Fujiwara K, Kano M, Hasebe S, Takinami A. Soft-sensor development using correlation-based just-in-time modeling. AIChE J. 2009;55:1754–1765.
[7] Schaal S, Atkeson CG, Vijayakumar S. Scalable techniques from onparametric statistics for real time robot learning. Appl. Intell. 2002;17:49–60.
[8] Kaneko H, Funatsu K. Maintenance-free soft sensor models with time difference of process variables. Chemom. Intell. Lab. Syst. 2011;107:312–317.
[9] Kaneko H, Funatsu K. A soft sensor method based on values predicted from multiple intervals of time difference for improvement and estimation of prediction accuracy. Chemom. Intell. Lab. Syst. 2011;109:197–206.
[10] Kaneko H, Funatsu K. Development of soft sensor models based on time difference of process variables with accounting for nonlinear relationship. Ind. Eng. Chem. Res. 2011;50:10643–10651.
[11] Kadlec P, Grbic R, Gabrys B. Review of adaptation mechanisms for data-driven soft sensors. Comput. Chem. Eng. 2011;35:1–24.
[12] Yu J. Multiway gaussian mixture model based adaptive kernel partial least squares regression method for soft sensor estimation and reliable quality prediction of nonlinear multiphase batch processes. Ind. Eng. Chem. Res. 2012;51:13227–13237.
[13] Yu J. A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses. Comput Chem Eng. 2012;41:134–144.
[14] Okada T, Kaneko H, Funatsu K. Development of a model selection method based on reliability of a soft sensor model, Songklanakarin J. Sci. Technol. 2012;34:217–222.
[15] Kaneko H, Funatsu K. Classification of the degradation of soft sensor models and discussion on adaptive models. AIChE J. in press.
[16] Ma J, Theliler J, Perkins S. Accurate on-line support vector regression. Neural Comput. 2003;15:2683–2703
[17] Bishop CM. *Pattern recognition and machine learning*. New York: Springer; 2006.
[18] Cherkassky V, Ma Y. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks* 2004;**17**:113–126.
[19] Wold S, Sjöström M, Eriksson L. PLS-regression: a basic tool of chemometrics. Chemom. Intell. Lab. Syst. 2001;58:109–130.
[20] Kim S, Kano M, Nakagawa H, Hasebe S. Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection. Int. J. Pharm. 2011;421:269–274.
[21] McAuley KB, MacGregor JF. On-line inference of polymer properties in an industrial polyethylene reactor. AIChE J. 1991;37:825–835.
[22] Kaneko H, Arakawa M, Funatsu K. Novel soft sensor method for detecting completion of transition in industrial polymer processes. Comput Chem Eng. 2011;35:1135–1142.