

Article

A Novel Just-in-Time Learning Strategy for Soft Sensing with Improved Similarity Measure Based on Mutual Information and PLS

Yueli Song ^{1,2}  and Minglun Ren ^{1,2,*}

¹ School of Management, Hefei University of Technology, Hefei 230009, China; song052414@mail.hfut.edu.cn

² Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei 230009, China

* Correspondence: renml@hfut.edu.cn; Tel.: +86-139-0569-3529

Received: 31 May 2020; Accepted: 4 July 2020; Published: 7 July 2020



Abstract: In modern industrial process control, just-in-time learning (JITL)-based soft sensors have been widely applied. An accurate similarity measure is crucial in JITL-based soft sensor modeling since it is not only the basis for selecting the nearest neighbor samples but also determines sample weights. In recent years, JITL similarity measure methods have been greatly enriched, including methods based on Euclidean distance, weighted Euclidean distance, correlation, etc. However, due to the different influence of input variables on output, the complex nonlinear relationship between input and output, the collinearity between input variables, and other complex factors, the above similarity measure methods may become inaccurate. In this paper, a new similarity measure method is proposed by combining mutual information (MI) and partial least squares (PLS). A two-stage calculation framework, including a training stage and a prediction stage, was designed in this study to reduce the online computational burden. In the prediction stage, to establish the local model, an improved locally weighted PLS (LWPLS) with variables and samples double-weighted was adopted. The above operations constitute a novel JITL modeling strategy, which is named MI-PLS-LWPLS. By comparison with other related JITL methods, the effectiveness of the MI-PLS-LWPLS method was verified through case studies on both a synthetic Friedman dataset and a real industrial dataset.

Keywords: just-in-time learning; locally weighted partial least squares; mutual information; similarity measure; soft sensor

1. Introduction

Data-driven soft sensors are usually models built based on a large quantity of data generated in production processes and can be used to monitor key indicators that are difficult to measure [1,2]. With the development of big data and other information technologies, data acquisition and processing in industrial processes have become easier, which makes data-driven soft sensors very popular in industrial process monitoring, quality prediction, and other process control-related tasks [3–5].

Traditional soft sensor models usually adopt some linear or non-linear methods, including partial least squares [6,7], support vector machine [8,9], artificial neural network [10,11], etc. These are basic kinds of global modeling methods since the models are built offline based on historical data. A characteristic feature of a global model is that once the model is established, it is difficult to adaptively adjust to a change of processes, so its performance may gradually degrade in practical application. To solve this issue, adaptive methods such as moving window [12–14], time difference [15,16], and recursive methods [17] are proposed. Although these methods can adapt to slow changes in processes to some extent, they cannot deal with abrupt changes [18,19].

Just-in-time learning (JITL) provides a new way to deal with the changing characteristics of processes. Different from the global methods, the JITL-based method does not build models in advance but only stores the historical sample data. When the prediction is needed, the JITL-based method performs the following three steps [19–21]: first, the nearest neighbor samples for a query point based on the similarity measure are selected; second, the selected nearest neighbor samples are used to establish a local model; third, the predicted value according to the model is calculated, and then the model is discarded; and the above three steps are repeated when the next query comes. It can be seen that the selection of the nearest neighbor samples and the establishment of the local model are two important steps. Only by selecting the sample points that are close to the query point can we establish an effective local model and obtain high prediction accuracy. The selection of sample points depends on the similarity index, so the similarity between samples should be defined carefully.

At present, there are many methods to measure the similarity between samples in JITL. Euclidean distance (ED) [22], weighted Euclidean distance (WED) [23–25], and distance and angle methods are some usual methods [21]. Correlation-based similarity measurement methods were also proposed by Japanese scholars in recent years [26,27]. These methods have been proved to be effective in some applications. However, the output information of samples is ignored completely in a similarity measure, which makes these methods inaccurate and causes a larger deviation in the nearest neighbor sample selection. By using a global model to estimate the query output first, the output information of historical data and query point can both be used to define the similarity [28,29]. However, the accuracy of the global model used in this method is usually not high, and the estimation error also has a negative impact on the selection of nearest neighbor samples. In order to make full use of information of samples, and avoid the introduction of additional estimation errors, Yuan [19] proposes a supervised latent structure method, which uses partial least squares (PLS) to project the historical samples and the query point into a low-dimensional latent variable space and uses Euclidean distance to calculate the similarity in the latent variable space. However, this method relies on the PLS model, which cannot deal with nonlinearity or non-Gaussian distribution. Considering different effects of input variables on output, a WED method based on mutual information (MI) has been proposed in the literature [30,31]. MI can not only capture the linear or nonlinear correlation between variables but is also not limited by the data distribution characteristics, so the above MI-based WED method gives a more accurate similarity measure than traditional methods. However, when input variables are multicollinear, the MI-based method may not be satisfactory. Especially in the case of high-dimensional input, collinearity will not only make the calculation of the whole model complex but also affect the accuracy of the similarity measure.

Through the above literature review, it can be seen that the different contributions of input variables to output, the correlation between input variables, the redundancy of input variables, and other factors can affect the similarity measure between samples. Especially in the case of high-dimensional input, the collinearity or redundancy between input variables can also reduce the efficiency of similarity calculation. To overcome the shortcomings of the above methods in the similarity measure, a novel MI-PLS-based similarity measure is proposed in this paper by skillfully combining MI and PLS. By calculating mutual information between input and output variables and realizing variable weighted based on MI, the correlation between input and output variables can be accurately described, and some uncorrelated redundant variables can be removed at the same time. Then, by using PLS to project the weighted variables into a low dimensional space, we can eliminate the influence of collinearity between variables on the similarity measure, and the computational efficiency can also be improved by reducing the dimension. In order to apply the proposed similarity measure method to soft sensor modeling in the JITL framework, a two-stage strategy was also designed to improve the computational efficiency of online prediction by avoiding repeated calculation as much as possible. The main calculation steps are as follows: in the training stage, MI between input and output variables is firstly calculated, then each variable is weighted based on its MI value. After that, the irrelevant variables with zero weights are removed. Finally, PLS is used to project the remaining weighted variables into latent variable space.

In the prediction stage, for a query point, the similarity and sample weights are calculated by the ED-based method in the latent variable space. Then, locally weighted partial least squares (LWPLS), with variables and samples double-weighted, are used for local modeling and answering a prediction query. All the above calculation processes constitute a novel JITL soft sensor modeling strategy, which is named MI-PLS-LWPLS. By using MI, correlation information between input and output is described accurately in a similarity calculation. At the same time, PLS is used to overcome the influence of collinearity, and the double weighting method is used to describe the different importance of variables to output and historical samples to query a sample in local modeling [19]. Therefore, the proposed method can achieve high prediction accuracy. The effectiveness of MI-PLS-LWPLS was verified by using both numerical and industrial cases. The proposed similarity measure method is generally applicable to soft sensor modeling in the JITL framework. Although LWPLS was chosen to build the local model in this study, when the process has strong non-linear characteristics, it may be necessary to select methods such as Gaussian process regression (GPR) or support vector regression (SVR) to build a local model. At this time, the proposed similarity measure method can still be considered in combination with these methods by selecting accurate neighbor samples to achieve a prediction model with good performance.

This paper is arranged as follows: Section 2 briefly reviews MI and LWPLS. Section 3 introduces the proposed method in detail. Section 4 verifies the effectiveness of the proposed method through numerical and industrial cases. Conclusions are made in Section 5.

2. Preliminaries

2.1. Mutual Information

MI between two random variables can represent the degree of their interdependence. The larger the MI value is, the more relevant the two variables are. Compared with traditional correlation criteria, such as correlation coefficient, cross-correlogram, etc., MI can describe the correlation among variables more comprehensively, including linear, periodic, or nonlinear correlation [32].

Given two random variables X and Y , the MI between them is defined as follows [33]:

$$I(X; Y) = \iint \mu(x, y) \log \frac{\mu(x, y)}{\mu_X(x)\mu_Y(y)} dx dy \quad (1)$$

Here, $\mu_X(x)$ and $\mu_Y(y)$ are marginal probability distributions, and $\mu(x, y)$ is the joint probability distribution. To calculate MI, the probability density functions (PDFs) need to be estimated first. The commonly used methods to calculate MI based on PDF estimation are histogram and kernel-based estimators. However, it is not easy to estimate the PDFs of random variables accurately in practical applications. Kraskov et al. [33] proposed a K-nearest neighbor (K-NN) method to directly calculate MI from data samples and avoid PDF estimation. This greatly reduces the complexity of the MI calculation.

Consider a new space $Z = (X, Y)$, which is built from the original variables X and Y ; for any point $z_i = (x_i, y_i)$, $i = 1, 2, \dots, N$, the distance from point z_i to its K-nearest neighbor $z_k = (x_k, y_k)$ is defined as follows:

$$D_k = \frac{\varepsilon(i)}{2} = \max(\|x_i - x_k\|, \|y_i - y_k\|) \quad (2)$$

Then, for other points, $z_j = (x_j, y_j)$, ($j \neq i$) in Z space, count $n_x(i)$ the number of the points x_j that satisfy $\|x_i - x_j\| \leq \varepsilon(i)/2$ and $n_y(i)$ the number of the points y_j that satisfy $\|y_i - y_j\| \leq \varepsilon(i)/2$. Then, MI can be calculated by the following formula:

$$I(X; Y) = \psi(K) - \frac{1}{N} \sum_{i=1}^N (\psi(n_x(i) + 1) + \psi(n_y(i) + 1)) + \psi(N) \quad (3)$$

where $\psi(x)$ is the digamma function satisfying $\psi(x + 1) = \psi(x) + 1/x$, $\psi(1) = -C$, $C = 0.5772156 \dots$. Parameter K denotes the number of neighbors and is usually set to be an integer in the range of 2 to 8 [33,34].

2.2. Locally Weighted PLS

LWPLS, which combines PLS with local learning, can deal with nonlinearity and collinearity and is a very commonly used JITL modeling method [35,36].

For the input and output variables $\mathbf{X} \in \mathbb{R}^{N \times p}$ and $\mathbf{Y} \in \mathbb{R}^{N \times 1}$, N is the sample number, and p represents the input variable number. The n th sample is denoted by (x_n, y_n) , $y_n \in \mathbb{R}$, $x_n \in \mathbb{R}^{1 \times p}$ defined by:

$$\mathbf{x}_n = [x_{n1}, x_{n2}, \dots, x_{np}] \quad (4)$$

To estimate the output of a new sample $\mathbf{x}_q \in \mathbb{R}^{1 \times p}$, firstly, d_n representing ED between \mathbf{x}_q and \mathbf{x}_n is calculated. Based on this, the sample weight or sample similarity s_n is defined as follows:

$$d_n = \sqrt{(\mathbf{x}_n - \mathbf{x}_q)(\mathbf{x}_n - \mathbf{x}_q)^T} \quad (5)$$

$$s_n = \exp\left(-\frac{d_n}{h \times \sigma_d}\right) \quad (6)$$

where σ_d is the standard deviation of $\mathbf{D} = [d_1, d_2, \dots, d_N]$, and h is called bandwidth, which can control the speed of weight attenuation. The smaller h is, the faster the weight decays; on the contrary, the larger h is, the slower the weight decays [35–37]. Then, an $N \times N$ matrix $\mathbf{\Omega}$ is built as follows:

$$\mathbf{\Omega} = \text{diag}[s_1, s_2, \dots, s_N] \quad (7)$$

Generally, the output estimation of \mathbf{x}_q is obtained by the following calculation steps 1–11 [37,38]:

- 1: Set the number of latent variables R and the tuning parameter h ;
- 2: Calculate $\mathbf{\Omega}$;
- 3: Calculate \mathbf{X}_0 , \mathbf{Y}_0 , and $\mathbf{x}_{q,0}$;

$$\mathbf{X}_0 = \mathbf{X} - \mathbf{1}_N[\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p] \quad (8)$$

$$\mathbf{Y}_0 = \mathbf{Y} - \mathbf{1}_{N \times 1} \bar{y} \quad (9)$$

$$\mathbf{x}_{q,0} = \mathbf{x}_q - [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p] \quad (10)$$

$$\bar{x}_i = \frac{\sum_{n=1}^N s_n x_{ni}}{\sum_{n=1}^N s_n} \quad (11)$$

$$\bar{y} = \frac{\sum_{n=1}^N s_n y_n}{\sum_{n=1}^N s_n} \quad (12)$$

- 4: Initialize: $\mathbf{X}_r = \mathbf{X}_0$, $\mathbf{Y}_r = \mathbf{Y}_0$, $\mathbf{x}_{q,r} = \mathbf{x}_{q,0}$, $\hat{y}_q = \bar{y}$;
- 5: For $r = 1: R$;
- 6: Calculate the weight loading \mathbf{W}_r ;

$$\mathbf{W}_r = \frac{\mathbf{X}_r^T \mathbf{\Omega} \mathbf{Y}_r}{\|\mathbf{X}_r^T \mathbf{\Omega} \mathbf{Y}_r\|} \quad (13)$$

Derive the r th latent variables.

$$\mathbf{t}_r = \mathbf{X}_r \mathbf{W}_r, \mathbf{t}_{q,r} = \mathbf{x}_{q,r} \mathbf{W}_r \quad (14)$$

7: Derive X-loading vector \mathbf{p}_r and Y-regression coefficient q_r ;

$$\mathbf{p}_r = \mathbf{X}_r^T \mathbf{\Omega} \mathbf{t}_r / \mathbf{t}_r^T \mathbf{\Omega} \mathbf{t}_r, q_r = \mathbf{Y}_r^T \mathbf{\Omega} \mathbf{t}_r / \mathbf{t}_r^T \mathbf{\Omega} \mathbf{t}_r \quad (15)$$

8: Update $\hat{y}_q = \hat{y}_q + t_{q,r} q_r$;

9: Update \mathbf{X}_{r+1} , \mathbf{Y}_{r+1} , and $\mathbf{x}_{q,r+1}$;

$$\mathbf{X}_{r+1} = \mathbf{X}_r - \mathbf{t}_r \mathbf{p}_r^T \quad (16)$$

$$\mathbf{Y}_{r+1} = \mathbf{Y}_r - \mathbf{t}_r q_r \quad (17)$$

$$\mathbf{x}_{q,r+1} = \mathbf{x}_{q,r} - t_{q,r} \mathbf{p}_r^T \quad (18)$$

10: End for;

11: Output \hat{y}_q .

3. The Proposed Method

Firstly, the similarity measure based on PLS latent structure proposed by Yuan [19] is briefly described. On this basis, the proposed JITL method with the MI-PLS-based similarity measure is introduced in detail.

3.1. PLS-Based Similarity Measure

The PLS-based similarity measure method calculates the similarity by using ED in latent variable space. Suppose that $\mathbf{X} \in \mathbb{R}^{N \times p}$ and $\mathbf{Y} \in \mathbb{R}^{N \times 1}$ are input and output variables. The calculation formulas of the PLS algorithm are defined as:

$$\begin{aligned} \mathbf{X} &= \mathbf{T}^T \mathbf{P} + \mathbf{E} \\ \mathbf{Y} &= \mathbf{U}^T \mathbf{Q} + \mathbf{F} \end{aligned} \quad (19)$$

$\mathbf{T} \in \mathbb{R}^{N \times R}$ ($1 \leq R \leq p$) represents the latent variable score matrix of input space. Let $\mathbf{T}_j \in \mathbb{R}^{N \times 1}$ ($j = 1, 2, \dots, R$) represent the j th latent variable and $\mathbf{t}_n \in \mathbb{R}^{1 \times R}$ ($n = 1, 2, \dots, N$) represent the n th sample in latent variable space, i.e., $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_R] = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N]^T$, and \mathbf{t}_q represents the query sample. Then, the ED between \mathbf{t}_q and \mathbf{t}_n can be calculated as follows:

$$d_{n,LV} = \sqrt{(\mathbf{t}_n - \mathbf{t}_q)(\mathbf{t}_n - \mathbf{t}_q)^T} \quad (20)$$

On this basis, the weight of the n th sample is defined as:

$$S_{n,LV} = \exp\left(\frac{-d_{n,LV}}{h\sigma_d}\right) \quad (21)$$

Here, h is the tuning parameter, which is also known as bandwidth. σ_d is the standard deviation of $d_{n,LV}$ ($n = 1, 2, \dots, N$).

3.2. The Proposed MI-PLS-LWPLS Method

Compared with traditional similarity measure methods, which only use input information, the PLS-based similarity measure introduced in Section 3.1 can select nearest neighbor samples more accurately by using supervised latent structure. However, PLS cannot describe a nonlinear correlation between input and output. Instead, mutual information can express both linear and nonlinear

correlation at the same time, and it cannot be affected by data distribution. References [30,31] adopted an MI-based similarity measure, and the results showed that this method can obtain better prediction accuracy than the traditional similarity measure. However, MI cannot deal well with the redundancy caused by the correlation between input variables [39]. Therefore, in the case of multiple collinearities between input variables, the prediction results are not ideal.

To develop a JITL-based soft sensor with good performance in the case of nonlinearity and collinearity, a novel similarity measure method combining MI and PLS is proposed. In order to fully consider the different importance of variables to the output and samples to the query sample, a two-stage strategy was designed to realize double weighting variables and samples in building an LWPLS-based local model. Figure 1 gives the two-stage flow chart of the proposed method, which is termed MI-PLS-LWPLS.

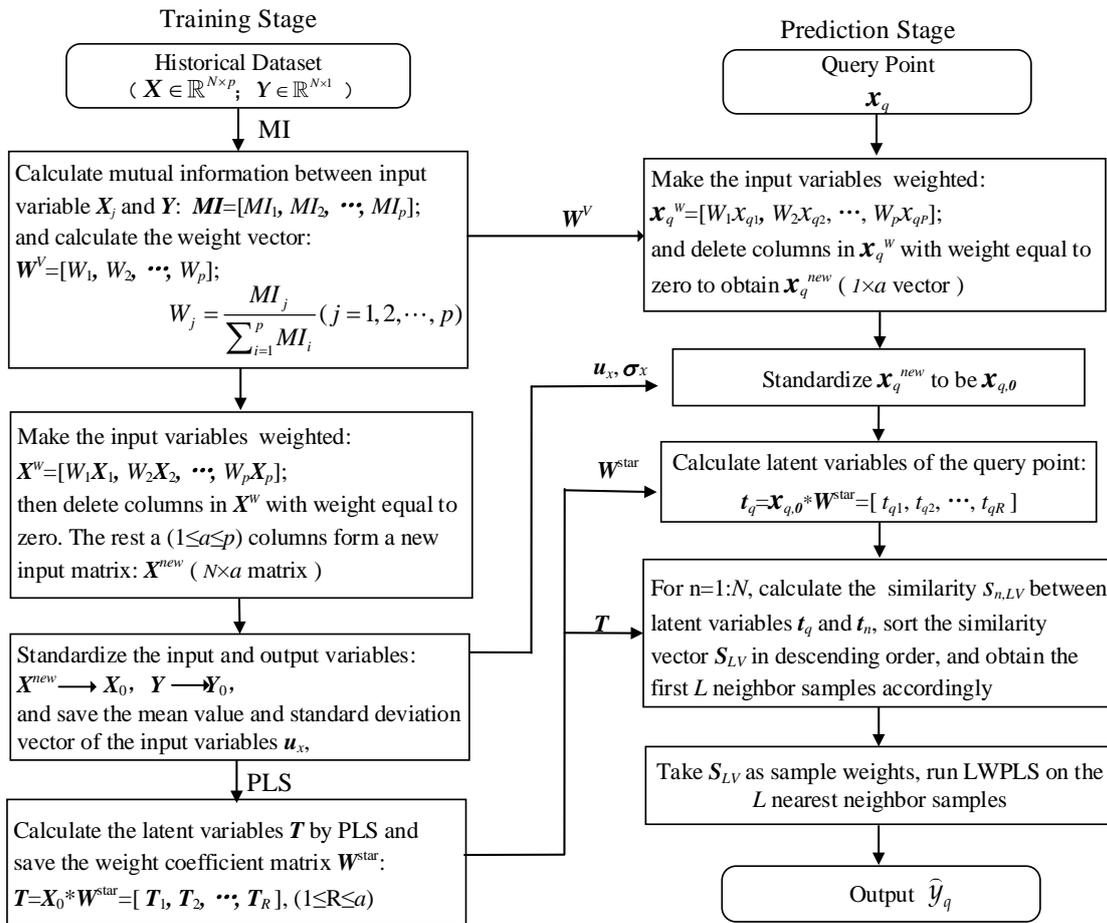


Figure 1. Two-stage flow chart of the proposed method.

3.2.1. Training Stage

In the training stage, some important variables and parameters, such as MI between input variables and output, latent variables, and weight matrix are obtained by offline computing based on the historical dataset so as to prepare for online prediction. Detailed computing steps are given below.

Step 1: Calculate MI between each input variable X_j and the output Y to obtain a mutual information vector $MI = [MI_1, MI_2, \dots, MI_p]$. Then calculate the variable weight vector $W^V = [W_1, W_2, \dots, W_p]$, here W_j ($j = 1, 2, \dots, p$), as follows:

$$W_j = \frac{MI_j}{\sum_{i=1}^p MI_i} \quad (22)$$

Step 2: Weight the input variables by using the weight vector \mathbf{W}^V and record the weighted input matrix as $\mathbf{X}^W = [W_1 \times 1, W_2 \times 2, \dots, W_p \times p]$. Then, remove all zero data columns caused by zero weight in \mathbf{X}^W , and form a new input matrix $\mathbf{X}^{new} \in \mathbb{R}^{N \times a}$ with the remaining a columns.

Step 3: Standardize input matrix \mathbf{X}^{new} and output variable \mathbf{Y} , then record them as \mathbf{X}_0 and \mathbf{Y}_0 , respectively. The formula is as follows:

$$\begin{aligned} x_{0,nj} &= \frac{x_{nj} - u_{x,j}}{\sigma_{x,j}} \\ y_{0,n} &= \frac{y_n - u_y}{\sigma_y} \end{aligned} \quad (23)$$

where x_{nj} represents the element of row n , column j of input matrix \mathbf{X}^{new} . $u_{x,j}$ and $\sigma_{x,j}$ respectively represent the element of column j of mean vector \mathbf{u}_x and standard deviation vector σ_x of input matrix \mathbf{X}^{new} . $x_{0,nj}$ is the value of the standardized x_{nj} . u_y and σ_y are the mean and standard deviation of output \mathbf{Y} , respectively. The n th element y_n of output \mathbf{Y} is expressed as $y_{0,n}$ after standardization.

Step 4: Take \mathbf{X}_0 and \mathbf{Y}_0 as input and output variables, respectively. Then, the input latent variable matrix \mathbf{T} is obtained by running PLS, and save the transformation weight coefficient matrix \mathbf{W}^{star} , where $\mathbf{T} = \mathbf{X}_0 * \mathbf{W}^{star} = [T_1, T_2, \dots, T_R]$, $1 \leq R \leq a$.

3.2.2. Prediction Phase

In the prediction stage, a query is responded to by the following calculation procedure. Firstly, parameters obtained in the training stage are used to complete the transformation calculation of the query sample, then a locally weighted model is established by using selected nearest neighbor samples to obtain predicted output value. Detailed computing steps are given below.

Step 1: Transform the query sample $\mathbf{x}_q = [x_{q1}, x_{q2}, \dots, x_{qp}]$ ($\mathbf{x}_q \in \mathbb{R}^{1 \times p}$) into $\mathbf{x}_q^W = [W_1 x_{q1}, W_2 x_{q2}, \dots, W_p x_{qp}]$ by using the weight vector \mathbf{W}^V obtained in step 1 of the training stage. Then, remove zero data columns caused by zero weight in \mathbf{x}_q^W and record the processed query sample vector as \mathbf{x}_q^{new} ($\mathbf{x}_q^{new} \in \mathbb{R}^{1 \times a}$).

Step 2: Standardize query sample \mathbf{x}_q^{new} to $\mathbf{x}_{q,0}$ according to the following equation:

$$\mathbf{x}_{q,0} = \frac{\mathbf{x}_q - \mathbf{u}_x}{\sigma_x} \quad (24)$$

Step 3: Project the query $\mathbf{x}_{q,0}$ into latent variable space to obtain \mathbf{t}_q by using the transformation weight coefficient matrix \mathbf{W}^{star} obtained in step 4 of the training stage. R is the latent variable number, $1 \leq R \leq a$.

$$\mathbf{t}_q = \mathbf{x}_{q,0} * \mathbf{W}^{star} = [t_{q1}, t_{q2}, \dots, t_{qR}] \quad (25)$$

Step 4: In the latent variable space, calculate Euclidean distance between the query sample \mathbf{t}_q ($1 \times R$ vector) and each training sample \mathbf{t}_n ($n = 1, 2, \dots, N$). Sample weights are also obtained according to the ED. Taking the n th sample as an example, the Euclidean distance $d_{n,LV}$ and the similarity $s_{n,LV}$ between the n th sample and the query sample are calculated by Equations (20) and (21).

Step 5: Sort the similarity vector S_{LV} in descending order and save the order index vector recorded as \mathbf{Ind} , and sort the training input matrix \mathbf{X}^{new} ($N \times a$) obtained in step 2 of the training stage according to the \mathbf{Ind} . Then the first L samples in \mathbf{X}^{new} with the largest similarity value are selected as the nearest neighbor samples. Finally, an LWPLS-based model with a sample weighted by S_{LV} is established, and the predicted output \hat{y}_q is then obtained by taking \mathbf{x}_q^{new} as the query input.

It can be seen that in the above calculation process, the input variables in LWPLS are weighted by MI, and the nearest neighbor samples used for local modeling are also weighted by their similarity indexes. By performing double weighting operations, both variable importance and sample importance are considered [19]. Therefore, the proposed modeling method can accurately describe the complex relationship between input and output variables and achieve high accuracy.

4. Case Studies

In this section, the effectiveness of the proposed MI-PLS-LWPLS modeling method is verified through a numerical case on a Friedman dataset [40,41] and an industrial debutanizer column process (DCP) case. Three other LWPLS methods based on different similarity measures are used to compare with MI-PLS-LWPLS. The four modeling methods are as follows:

ED-LWPLS: Traditional Euclidean distance-based LWPLS (calculating sample similarity and weight in original input space).

PLS-LWPLS: PLS latent structure-based LWPLS (calculating sample similarity and weight in latent variable space).

MI-LWPLS: MI weighted Euclidean distance-based LWPLS (calculating sample similarity by using MI weighted ED in original input space and assigning sample weight accordingly).

MI-PLS-LWPLS: The proposed MI-PLS-based LWPLS (combining MI and PLS together in the similarity measure and weight assignment).

The prediction accuracy is measured by the criteria mean absolute relative error (MARE) and root mean square error (RMSE), defined as follows:

$$\text{MARE} = \frac{1}{M} \sum_{m=1}^M \left| \frac{y_m - \hat{y}_m}{y_m} \right| \times 100\% \quad (26)$$

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{m=1}^M (y_m - \hat{y}_m)^2} \quad (27)$$

Here, y_m and \hat{y}_m respectively represent the real and predicted values of the m th test point, and M represents the total sample number of the test dataset.

4.1. Numerical Experiment on Friedman Dataset

4.1.1. Experimental Design

The Friedman dataset is defined by the equation below [40,41]:

$$Y = 10 \sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \varepsilon \quad (28)$$

Here, $X_1 \sim X_{10}$ are random variables uniformly distributed in the interval [0,1], and ε is white noise of standard normal distribution. One can see that the output Y is related to the input variables $X_1 \sim X_5$, but not to $X_6 \sim X_{10}$.

The two cases below were investigated.

Case 1: Generate Friedman data based on the above Equation (28) and take $X_1 \sim X_{10}$ as the input and Y as the output to form a dataset.

Case 2: On the basis of case 1, add two input variables X_{11} and X_{12} , which are determined by X_1 , X_2 , and X_3 as follows:

$$\begin{aligned} X_{11} &= 0.5(X_1 + X_2) \\ X_{12} &= 0.5X_3 \end{aligned} \quad (29)$$

We took $X_1 \sim X_{12}$ as the input and y as the output to form a new dataset. One can observe that in case 2, there are not only uncorrelated input variables $X_6 \sim X_{10}$ but also redundant variables X_{11} and X_{12} , which are collinear with X_1 , X_2 , and X_3 .

For the above two cases, 400 data samples were randomly generated, 300 of which were taken as training data and the remaining 100 as test data. The four modeling methods mentioned above were used in the experiment. The following parameters needed to be determined in the application of the four methods:

L : Number of neighbor samples used for local modeling in LWPLS;

R : Number of latent variables in LWPLS;

h : Tuning parameter in sample weight calculation;

K : Number of nearest neighbor samples used in K -NN-based MI estimation. K is usually an integer in the range of 2 to 8 [33,34].

To determine parameters L and R , the influence of the changes of L and R on RMSE was studied by the cross-validation method. Figure 2 shows the results. The value of L varies from 10 to 100 with a step size of 10, and the value of R is an integer between 1 and 12.

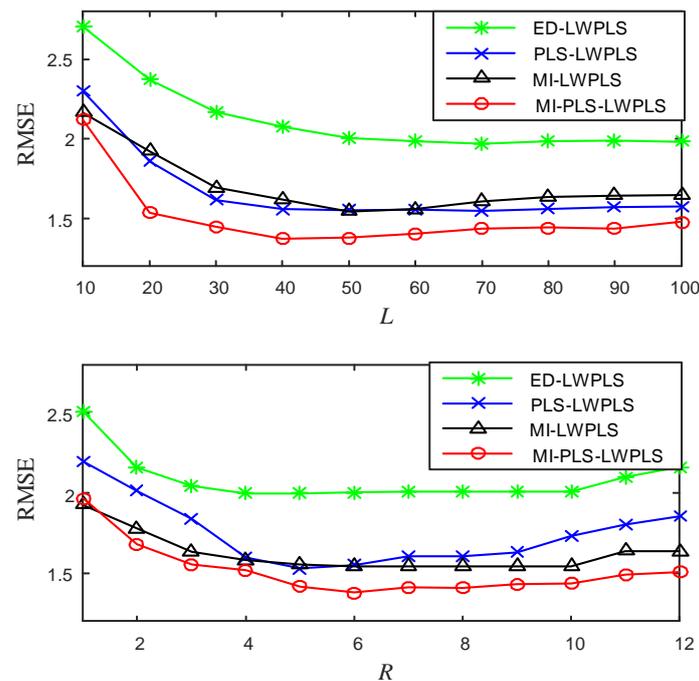


Figure 2. Influence of the changes of L and R on root mean square errors (RMSEs) of the four methods.

From Figure 2, one can see that when the value of L is in the interval 40–100, the RMSEs of the four methods change very little. When $L = 50$, the four methods can obtain their own smaller RMSEs, so the L values of the four methods are all set to 50. The changes of RMSEs with R are similar to the above situation. When R changes between interval 5–10, the value of RMSEs fluctuates very little. When R is greater than 10, RMSEs tend to increase, indicating that the prediction results become worse. Therefore, R was set to six in this study according to the result shown in Figure 2.

Bandwidth parameter h was selected by trial-and-error experiments. Firstly, the initial value set of h was set as {0.01, 0.05, 0.1, 0.3, 0.6, 0.8, 1, 1.3, 1.6, 2, 5, 10, 20, 30, 50}. By minimizing the RMSE of the cross-validation experiment, an initial optimal h value could be obtained. Then, by further narrowing the selection range, a new value was set around the initial optimal h value, and finally, the optimal bandwidth parameter h was obtained by constantly narrowing the selection range and step size.

For parameter K , in order to avoid the inaccuracy in MI estimation caused by taking a specific K value, the following strategy was adopted: the mutual information with K set to be each integer in the interval 2–8 was calculated at first, and then the average of all MI values was taken as the final MI value.

4.1.2. Results and Discussion

Table 1 gives statistical analysis results of prediction errors of the four modeling methods. It is observed that the MI-PLS-LWPLS method achieves minimum RMSE and MARE in both cases, which means that it has the best prediction performance. By further observing the results of the first method

ED-LWPLS and the third method MI-LWPLS, their RMSE and MARE values in case 2 are both greater than those in case 1, which means the performance of these two methods in case 2 is worse than in case 1, while PLS-LWPLS and MI-PLS-LWPLS both perform better in case 2 than in case 1.

Table 1. Prediction errors of the four methods.

Method	Case 1		Case 2	
	RMSE	MARE (%)	RMSE	MARE (%)
ED-LWPLS	1.98	13.28	2.01	13.58
PLS-LWPLS	1.61	11.13	1.55	10.15
MI-LWPLS	1.50	9.99	1.54	10.11
MI-PLS-LWPLS	1.42	9.70	1.38	9.47

This is because two other collinear inputs X_{11} and X_{12} related to input X_1 , X_2 , and X_3 are added in case 2. ED-LWPLS and MI-LWPLS select samples and define weights based on ED and MI weighted ED, respectively, both in the original sample space. These two methods cannot deal with collinear redundancy of input variables in the sample selection procedure, so their performance gets worse in case 2 than in case 1. PLS-LWPLS and MI-PLS-LWPLS calculate the sample similarity and weights in latent variable space based on PLS transformation, which can overcome the influence of collinearity, so they are more effective in case 2. In addition, the proposed method considers the different correlations between input and output variables in similarity calculation by weighting input variables based on MI, so more accurate neighbor samples are selected. In the local modeling phase, the variable and sample double-weighted modeling scheme is adopted. Therefore, MI-PLS-LWPLS achieves the best performance among the four methods in both cases.

Figure 3 shows the scatter plots between real and predicted values of the four methods on the test set in case 2. Figure 3a shows the result of ED-LWPLS. It is observed that data points in Figure 3a are the most scattered among the four scatter plots, indicating that deviation between the predicted and real values is the largest. Prediction results of Figure 3b,c are close and both better than that of Figure 3a. Figure 3d shows the best prediction result since data points in Figure 3d are most concentrated near the diagonal line among the four plots. This also proves that the prediction accuracy of the MI-PLS-LWPLS method is the highest.

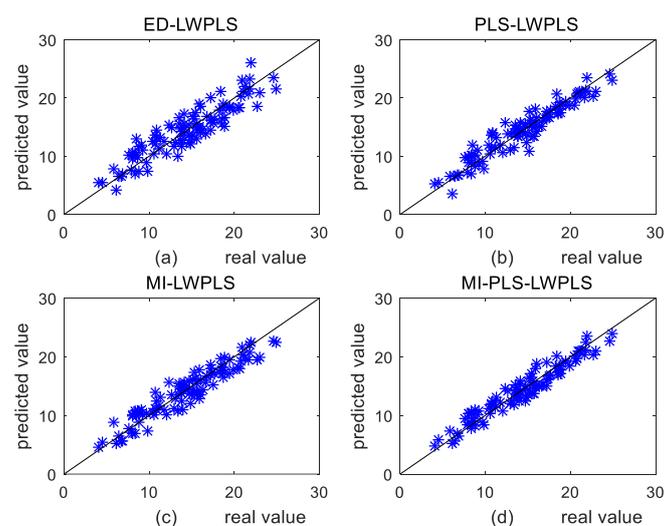


Figure 3. Comparison of prediction scatter plots in case 2: (a) Euclidean distance-based locally weighted partial least squares (ED-LWPLS); (b) PLS-LWPLS; (c) mutual information (MI)-LWPLS; (d) MI-PLS-LWPLS.

4.2. Industrial Case

4.2.1. Debutanizer Column Process

A debutanizer is used in the process of desulfurization and naphtha separation. Butane concentration at the bottom of the tower is an important index to ensure the quality of process control, so it needs to be monitored in real-time [1,19]. However, traditional online measurement using meteorological chromatography is very time-consuming and does not meet the needs of real-time control.

Therefore, a butane concentration measurement based on a soft sensor is an important alternative solution. To establish a soft sensing model for butane concentration measurement, seven variables that are easy to detect in the debutanizer were selected as auxiliary variables. The flow chart of the debutanizer column process (DCP) is shown in Figure 4, in which U1 ~ U7 are the installation locations of real-time monitoring devices for the seven auxiliary variables. An explanation of these seven variables is given in Table 2.

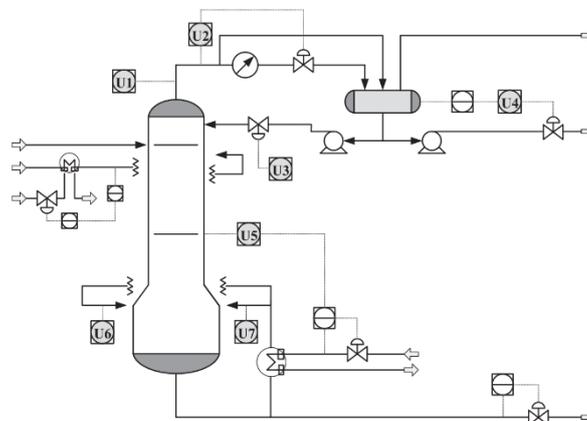


Figure 4. Debutanizer column process.

Table 2. Auxiliary variables for the debutanizer column process.

Case 1	Case 2
U1	Top temperature
U2	Top pressure
U3	Reflux flow
U4	Flow to next process
U5	6th tray temperature
U6	Bottom temperature
U7	Bottom pressure

The DCP dataset, provided by [1], contains 2394 samples obtained from a DCP and has been a popular benchmark for evaluating various soft sensors [42–45]. The first half of samples was chosen as a training set, the remaining half was divided into two parts including a validation set for parameter optimization and a test set. The following model structure was adopted, in which the input variables were expanded according to the experience of experts [1,45].

$$\hat{y}(t) = f_{\text{DCP}} \left(\begin{array}{c} U_1(t), U_2(t), U_3(t), U_4(t), U_5(t), U_5(t-1), U_5(t-2), \\ U_5(t-3), \frac{U_6(t)+U_7(t)}{2}, y(t-4), y(t-5), y(t-6) \end{array} \right) \quad (30)$$

where t is the current sampling time, $y(t)$ represents the actual butane concentration, $U_i(t)$ ($i = 1, 2, \dots, 7$) represents the sampling value of the i th input variable, and $\hat{y}(t)$ is obtained by the soft sensing model, representing the predicted butane concentration.

For ease of description, the 12 expanded input variables are noted as X_1, X_2, \dots, X_{12} . Firstly, the correlation between X_i ($i = 1, 2, \dots, 7$) and y was examined by calculating MI between them. Figure 5 shows the histogram of MI values between 12 input variables and y . One can see that MI values of different input variables vary greatly, indicating that they have different correlations with the output variable.

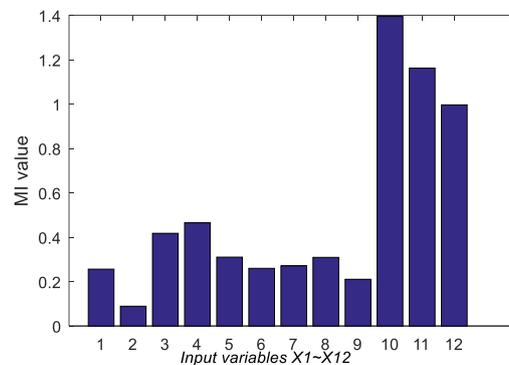


Figure 5. MI values between 12 input variables and y .

In this study, multicollinearity between input variables was also examined by using the common variance inflation factor (VIF) method. In this method, one of the input variables X_i is taken as the output, then the other variables are used for regression to obtain an estimated value \hat{X}_i , and then the variance inflation factor is calculated as follows:

$$\text{VIF}_i = \frac{1}{1 - R_i^2} \quad (31)$$

where R_i^2 is the determination coefficient obtained according to the regression result.

Generally, if there is at least one VIF_i ($i = 1, 2, \dots, 12$) greater than 10, it is considered that the input variables are multi-collinear. The VIF value of each input variable is shown in Table 3. One can see that there are several VIF values greater than 10, so multicollinearity does exist in DCP data.

Table 3. Variance inflation factor (VIF) values of input variables.

Input Variables	X_1	X_2	X_3	X_4	X_5	X_6
VIF	1.6	1.2	1.5	1.3	38.6	118.7
Input Variables	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
VIF	119.3	36.4	3.4	1078.5	3972.6	1020.7

To verify the effectiveness of MI-PLS-LWPLS, four modeling methods, ED-LWPLS, PLS-LWPLS, MI-LWPLS, and MI-PLS-LWPLS were used for soft sensor modeling on the DCP dataset, and their prediction results were compared. First of all, the values of parameters L (number of local modeling samples) and R (number of latent variables) in LWPLS needed to be determined. In this study, we selected the optimized parameter values by investigating RMSEs in the validation set. Figure 6 shows the change curves of RMSEs with different parameter values in the four methods. One can see that when the values of L and R are small, RMSEs of the four methods decrease with the increase of both L and R values. However, when L is greater than 60 or R is greater than 8, the changes of RMSEs are not significant. Therefore, the values of L and R in the four methods are determined as $L = 60$ and $R = 8$.

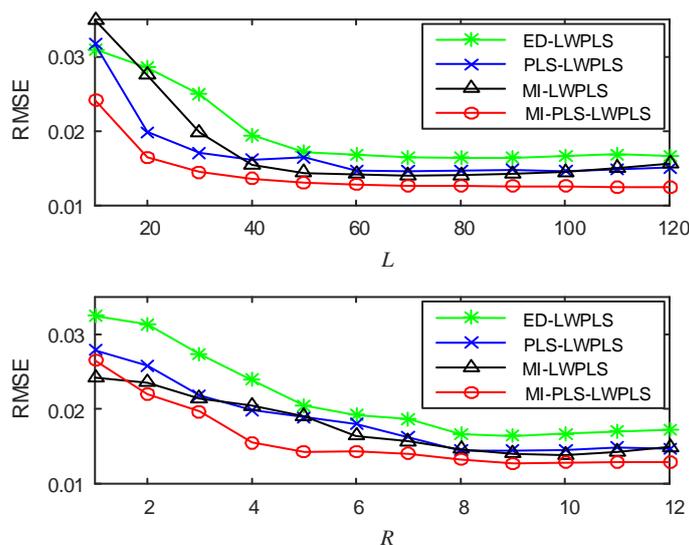


Figure 6. Changes of RMSEs with L and R in the validation set in the four methods.

The tuning parameter h is determined by selecting an initial optimal value from the value set $\{0.01, 0.05, 0.1, 0.3, 0.6, 0.8, 1, 1.3, 1.6, 2, 5, 10, 20, 30, 50\}$ to minimize the RMSE of the validation set, further reducing the selection range near the optimal value and finally determining the optimized h value. In this study, the optimal h values of the four methods ED-LWPLS, PLS-LWPLS, MI-LWPLS, and PLS-MI-LWPLS were finally selected as 0.05, 0.3, 0.01, and 2.6, respectively.

4.2.2. DCP Experimental Results and Analysis

Table 4 gives statistical prediction errors of the four methods on both the validation dataset and test dataset. It is observed that the MI-PLS-LWPLS method achieves the minimum RMSE and MARE, indicating that its prediction result is the best. The RMSE and MARE of ED-LWPLS have the largest values among the four methods, indicating the worst prediction result. This is because the ED-LWPLS method only uses the input information of historical samples to calculate similarity by ED in the original variable space, and it does not consider the different correlations between input and output variables. PLS-LWPLS and MI-LWPLS both use the input and output information of historical samples to calculate the similarity, which greatly improves the prediction accuracy. However, the PLS-based similarity measure method ignores the nonlinear correlation between input and output, and the MI-based similarity measure method cannot deal with the collinear redundancy of input variables, so the prediction performance of these two methods needs to be further improved. The proposed MI-PLS-LWPLS method combines the advantages of PLS and MI in similarity calculation, so as to deal with collinearity of input variables and the nonlinear correlation between the input and output. Therefore, it has the best performance on the DCP dataset with multicollinearity.

Table 4. Statistical analysis of prediction errors of the debutanizer column process (DCP) dataset.

Method	Validation Dataset		Test Dataset	
	RMSE	MARE (%)	RMSE	MARE (%)
ED-LWPLS	0.0164	5.81	0.0188	6.20
PLS-LWPLS	0.0146	5.27	0.0155	5.47
MI-LWPLS	0.0140	5.16	0.0153	5.42
MI-PLS-LWPLS	0.0129	4.10	0.0135	4.73

Figure 7 shows scatter plots of prediction results for the test set of the four methods, (a) ED-LWPLS, (b) PLS-LWPLS, (c) MI-LWPLS, and (d) MI-PLS-LWPLS. By comparing the four scatter plots in Figure 7,

one can see that the plot points in Figure 7d are most concentrated near the diagonal line, which proves that the MI-PLS-LWPLS method achieves the highest prediction accuracy. The performance of the other three methods needs to be improved on the complex DCP dataset with both nonlinearity and collinearity.

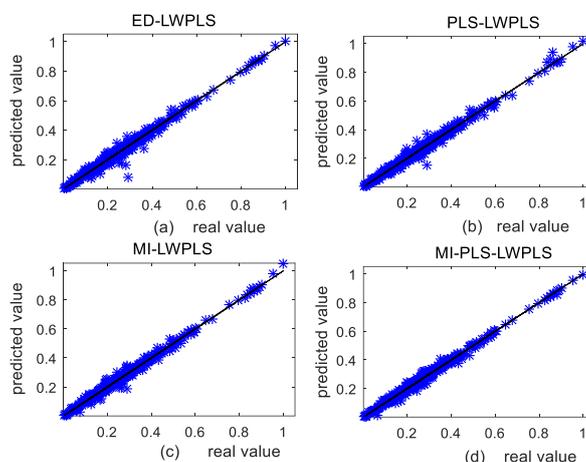


Figure 7. Comparison of prediction scatter plots for the test set of the four methods: (a) ED-LWPLS; (b) PLS-LWPLS; (c) MI-LWPLS; (d) MI-PLS-LWPLS.

In order to investigate the computational efficiency of the proposed MI-PLS-LWPLS modeling method, we compared the predicted response time of the four methods. For each method, the time required to respond to the entire test set was recorded. Each method was run 20 times, and then the average value of the response time was taken, as shown in Table 5. It can be seen that compared with other LWPLS modeling methods based on different similarity measures, the prediction response time of the proposed method is only slightly increased. This is because the two-stage calculation strategy designed in this paper puts the operation related to historical samples in the training stage as much as possible, while the online calculation only aims at the transformation and operation closely related to query samples, which guarantees a fast response speed. Compared with the slow dynamic characteristics of the chemical process and low sampling frequency of quality parameters, the response speed can meet the requirements of process control.

Table 5. Comparison of the response time of the four methods.

Method	Prediction Time (s)
ED-LWPLS	6.41
PLS-LWPLS	6.22
MI-LWPLS	7.19
MI-PLS-LWPLS	7.32

5. Conclusions

This paper mainly focuses on the similarity measure in the JITL framework for soft sensor modeling. Firstly, several representative traditional similarity measure methods were analyzed. Through the analysis, it is known that in order to accurately calculate the similarity between samples, some key factors need to be considered comprehensively, including both consideration of input and output information, the different effects of input on output, the redundancy and collinearity of input variables, and the complexity of calculation. Based on the analysis of the shortcomings of current similarity measure methods, a new similarity measure method combining MI and PLS is proposed. The main contribution of the proposed method in solving the similarity measure problem is as follows:

(1) MI is used to calculate the correlation between input variables and the output, and the input variables are weighted by the MI value, so that the input and output information, as well as the different contribution of input to output, can be both considered in the similarity measure, and the uncorrelated redundant variables are eliminated.

(2) The weighted input variables are projected by the PLS algorithm, and the sample similarity is calculated in the latent variable space. This allows the influence of collinearity between input variables on the similarity measure to be eliminated. In the case of high-dimensional input, dimension reduction by PLS can also alleviate the complexity of calculation.

In order to use the above MI-PLS-based similarity measure method to develop a soft sensor under the JITL framework, we used LWPLS, which is commonly used in soft sensor modeling to build the local model. A two-stage modeling strategy was designed to reduce the online computing burden as much as possible, including a training stage and a prediction stage, so as to ensure a fast response to queries. In addition, in order to fully describe the relationship between the input and output, we adopted the following double weighting strategy, which considers the importance of both variables and samples.

(3) Weighted by MI, variables with high correlation with the output get larger weights, and weighted by similarity, samples more similar to the query have larger weights in building the local model. By giving larger weights to the more relevant variables and samples, the mapping relationship between process input and output can be better described, so the accuracy of the model is improved.

Finally, the effectiveness of the proposed method was verified by both numerical and industrial cases.

Author Contributions: Conceptualization, M.R. and Y.S.; Data curation, Y.S.; Funding acquisition, M.R.; Investigation, Y.S.; Methodology, Y.S.; Project administration, M.R.; Supervision, M.R.; Writing—original draft, Y.S.; Writing—review & editing, Y.S. and M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No.71531008, No.71521001, No.71490720).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fortuna, L.; Graziani, S.; Rizzo, A.; Xibilia, M.G. *Soft Sensors for Monitoring and Control of Industrial Processes*; Springer: London, UK, 2007.
2. Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven Soft Sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. [[CrossRef](#)]
3. Zhang, S.; Chu, F.; Deng, G.; Wang, F. Soft Sensor Model Development for Cobalt Oxalate Synthesis Process Based on Adaptive Gaussian Mixture Regression. *IEEE Access* **2019**, *7*, 118749–118763. [[CrossRef](#)]
4. Grbic, R.; Sliskovic, D.; Kadlec, P. Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models. *Comput. Chem. Eng.* **2013**, *58*, 84–97. [[CrossRef](#)]
5. He, Y.; Zhu, B.; Liu, C.; Zeng, J. Quality-Related Locally Weighted Non-Gaussian Regression Based Soft Sensing for Multimode Processes. *Ind. Eng. Chem. Res.* **2018**, *57*, 17452–17461. [[CrossRef](#)]
6. Facco, P.; Doplicher, F.; Bezzo, F.; Barolo, M. Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process. *J. Process Control* **2009**, *19*, 520–529. [[CrossRef](#)]
7. Camacho, J.; Pico, J.; Ferrer, A. Bilinear modelling of batch processes. Part II: A comparison of PLS soft-sensors. *J. Chemom.* **2008**, *22*, 533–547. [[CrossRef](#)]
8. Jiang, H.; Yan, Z.; Liu, X. Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm. *Neurocomputing* **2013**, *119*, 469–477. [[CrossRef](#)]
9. Chang, Y.; Wang, F.; Wang, X.; Lv, Z. Soft sensor modeling based on support vector machines and its applications to fermentation process. *Chin J. Sci. Instrum.* **2006**, *27*, 241–244, 271.
10. Yan, X. Hybrid artificial neural network based on BP-PLSR and its application in development of soft sensors. *Chemom. Intell. Lab. Syst.* **2010**, *103*, 152–159. [[CrossRef](#)]

11. Pisa, I.; Santin, I.; Lopez Vicario, J.; Morell, A.; Vilanova, R. ANN-Based Soft Sensor to Predict Effluent Violations in Wastewater Treatment Plants. *Sensors* **2019**, *19*, 1280. [[CrossRef](#)]
12. Liu, J.; Chen, D.-S.; Shen, J.-F. Development of Self-Validating Soft Sensors Using Fast Moving Window Partial Least Squares. *Ind. Eng. Chem. Res.* **2010**, *49*, 11530–11546. [[CrossRef](#)]
13. Yao, L.; Ge, Z. Online Updating Soft Sensor Modeling and Industrial Application Based on Selectively Integrated Moving Window Approach. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 1985–1993. [[CrossRef](#)]
14. Wang, X.; Kruger, U.; Irwin, G.W. Process monitoring approach using fast moving window PCA. *Ind. Eng. Chem. Res.* **2005**, *44*, 5691–5702. [[CrossRef](#)]
15. 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. [[CrossRef](#)]
16. Kaneko, H.; Funatsu, K. Discussion on Time Difference Models and Intervals of Time Difference for Application of Soft Sensors. *Ind. Eng. Chem. Res.* **2013**, *52*, 1322–1334. [[CrossRef](#)]
17. Ahmed, F.; Nazir, S.; Yeo, Y.K. A recursive PLS-based soft sensor for prediction of the melt index during grade change operations in HDPE plant. *Korean J. Chem. Eng.* **2009**, *26*, 14–20. [[CrossRef](#)]
18. Yuan, X.; Ge, Z.; Huang, B.; Song, Z.; Wang, Y. Semisupervised JITL Framework for Nonlinear Industrial Soft Sensing Based on Locally Semisupervised Weighted PCR. *IEEE Trans. Ind. Informat.* **2017**, *13*, 532–541. [[CrossRef](#)]
19. Yuan, X.; Huang, B.; Ge, Z.; Song, Z. Double locally weighted principal component regression for soft sensor with sample selection under supervised latent structure. *Chemom. Intell. Lab. Syst.* **2016**, *153*, 116–125. [[CrossRef](#)]
20. Cheng, C.; Chiu, M.S. A new data-based methodology for nonlinear process modeling. *Chem. Eng. Sci.* **2004**, *59*, 2801–2810. [[CrossRef](#)]
21. Zhang, X.; Li, Y.; Kano, M. Quality Prediction in Complex Batch Processes with Just-in-Time Learning Model Based on Non-Gaussian Dissimilarity Measure. *Ind. Eng. Chem. Res.* **2015**, *54*, 7694–7705. [[CrossRef](#)]
22. Ge, Z.; Song, Z. A comparative study of just-in-time-learning based methods for online soft sensor modeling. *Chemom. Intell. Lab. Syst.* **2010**, *104*, 306–317. [[CrossRef](#)]
23. Hazama, K.; Kano, M. Covariance-based locally weighted partial least squares for high-performance adaptive modeling. *Chemom. Intell. Lab. Syst.* **2015**, *146*, 55–62. [[CrossRef](#)]
24. Kim, S.; Okajima, R.; Kano, M.; Hasebe, S. Development of soft-sensor using locally weighted PLS with adaptive similarity measure. *Chemom. Intell. Lab. Syst.* **2013**, *124*, 43–49. [[CrossRef](#)]
25. Shigemori, H.; Kano, M.; Hasebe, S. Optimum quality design system for steel products through locally weighted regression model. *J. Process Control* **2011**, *21*, 293–301. [[CrossRef](#)]
26. Fujiwara, K.; Kano, M.; Hasebe, S. Development of Correlation-Based Pattern Recognition and Its Application to Adaptive Soft-Sensor Design. In Proceedings of the ICCAS-SICE 2009, Fukuoka, Japan, 18–21 August 2009; pp. 1990–1995.
27. Fujiwara, K.; Kano, M.; Hasebe, S.; Takinami, A. Soft-Sensor Development Using Correlation-Based Just-in-Time Modeling. *Aiche J.* **2009**, *55*, 1754–1765. [[CrossRef](#)]
28. Chang, S.Y.; Ernie, H.B.; Bruce, C.M. Implementation of Locally Weighted Regression to Maintain Calibrations on FT-NIR Analyzers for Industrial Processes. *Appl. Spectrosc.* **2001**, *55*, 1199–1206. [[CrossRef](#)]
29. Wang, Z.; Isaksson, T.; BR, K. New approach for distance measurement in locally weighted regression. *Anal. Chem.* **1994**, *66*, 249–260. [[CrossRef](#)]
30. Zhao, D.; Pan, T.H.; Sheng, B.Q. Just-in-time Learning Algorithm Using the Improved Similarity Index. In Proceedings of the 35th Chinese Control Conference 2016, Chengdu, China, 27–29 July 2016; pp. 9065–9068.
31. Jin, H.; Chen, X.; Yang, J.; Wang, L.; Wu, L. Online local learning based adaptive soft sensor and its application to an industrial fed-batch chlortetracycline fermentation process. *Chemom. Intell. Lab. Syst.* **2015**, *143*, 58–78. [[CrossRef](#)]
32. Khan, S.; Bandyopadhyay, S.; Ganguly, A.R.; Saigal, S.; Erickson, D.J., 3rd; Protopopescu, V.; Ostrouchov, G. Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* **2007**, *76*, 026209. [[CrossRef](#)]
33. Kraskov, A.; Stogbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* **2004**, *69*, 066138. [[CrossRef](#)]
34. Gao, W.; Oh, S.; Viswanath, P. Demystifying Fixed k-Nearest Neighbor Information Estimators. *IEEE Trans. Inf. Theory* **2018**, *64*, 5629–5661. [[CrossRef](#)]

35. Kano, M.; Kim, S.; Okajima, R.; Hasebe, S. Industrial Applications of Locally Weighted PLS to Realize Maintenance-Free High-Performance Virtual Sensing. In Proceedings of the 2012 12th International Conference on Control, Automation and Systems, Jeju Island, Korea, 17–21 October 2012; pp. 545–548.
36. Kim, S.; Kano, M.; Hasebe, S.; Takinami, A.; Seki, T. Long-Term Industrial Applications of Inferential Control Based on Just-In-Time Soft-Sensors: Economical Impact and Challenges. *Ind. Eng. Chem. Res.* **2013**, *52*, 12346–12356. [[CrossRef](#)]
37. Ren, M.; Song, Y.; Chu, W. An Improved Locally Weighted PLS Based on Particle Swarm Optimization for Industrial Soft Sensor Modeling. *Sensors* **2019**, *19*, 4099. [[CrossRef](#)] [[PubMed](#)]
38. Kamata, K.; Fujiwara, K.; Kinoshita, T.; Kano, M. Missing RRI Interpolation Algorithm based on Locally Weighted Partial Least Squares for Precise Heart Rate Variability Analysis. *Sensors* **2018**, *18*, 3870. [[CrossRef](#)]
39. Chen, C.; Yan, X. Selection and transformation of input variables for RVM based on MIPCAMI and 4-CBA concentration model. *Asia-Pac. J. Chem. Eng.* **2013**, *8*, 69–76. [[CrossRef](#)]
40. Jerome, H.F. Multivariate adaptive regression splines. *Ann. Stat.* **1991**, *19*, 1–67.
41. Han, M.; Ren, W.J. Global mutual information-based feature selection approach using single-objective and multi-objective optimization. *Neurocomputing* **2015**, *168*, 47–54. [[CrossRef](#)]
42. Kadlec, P.; Grbić, R.; Gabrys, B. Review of adaptation mechanisms for data-driven soft sensors. *Comput. Chem. Eng.* **2011**, *35*, 1–24. [[CrossRef](#)]
43. Shao, W.; Tian, X. Semi-supervised selective ensemble learning based on distance to model for nonlinear soft sensor development. *Neurocomputing* **2017**, *222*, 91–104. [[CrossRef](#)]
44. Yao, L.; Ge, Z. Locally Weighted Prediction Methods for Latent Factor Analysis with Supervised and Semisupervised Process Data. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 126–138. [[CrossRef](#)]
45. Shao, W.; Tian, X.; Wang, P. Local Partial Least Squares Based Online Soft Sensing Method for Multi-output Processes with Adaptive Process States Division. *Chin. J. Chem. Eng.* **2014**, *22*, 828–836. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).