Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Quality-related locally weighted soft sensing for non-stationary processes by a supervised Bayesian network with latent variables*

Yuxue XU[1], Yun WANG[2], Tianhong YAN[†‡1], Yuchen HE[†‡1], Jun WANG[1], De GU[3], Haiping DU[4], Weihua LI[5]

[1]*College of Mechanical and Electrical Engineering, China Jiliang University, Hangzhou 310018, China*
[2]*Mechanical and Electrical Engineering Department, Zhejiang Tongji Vocational College of Science and Technology, Hangzhou 311231, China*
[3]*Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Institute of Automation, Jiangnan University, Wuxi 214122, China*
[4]*School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia*
[5]*School of Mechanical, Materials, Mechatronic, and Biomedical Engineering, University of Wollongong, Wollongong, NSW 2522, Australia*
[†]E-mail: thyan@cjlu.edu.cn; yche@cjlu.edu.cn

Received Aug. 24, 2020; Revision accepted Dec. 23, 2020; Crosschecked July 22, 2021

Abstract: Soft sensors are widely used to predict quality variables which are usually hard to measure. It is necessary to construct an adaptive model to cope with process non-stationaries. In this study, a novel quality-related locally weighted soft sensing method is designed for non-stationary processes based on a Bayesian network with latent variables. Specifically, a supervised Bayesian network is proposed where quality-oriented latent variables are extracted and further applied to a double-layer similarity measurement algorithm. The proposed soft sensing method tries to find a general approach for non-stationary processes via quality-related information where the concepts of local similarities and window confidence are explained in detail. The performance of the developed method is demonstrated by application to a numerical example and a debutanizer column. It is shown that the proposed method outperforms competitive methods in terms of the accuracy of predicting key quality variables.

Key words: Soft sensor; Supervised Bayesian network; Latent variables; Locally weighted modeling; Quality prediction
https://doi.org/10.1631/FITEE.2000426 **CLC number:** TP277

1 Introduction

Most industrial processes are well equipped with online process sensors, such as temperature, flow rate, and pressure sensors, for monitoring, control, and maintenance. When online sensors cannot be used to predict or estimate variables which are difficult to measure, soft sensors are key for successfully estimating product quality or other important variables (Kano and Fujiwara, 2012; Al-Jlibawi et al., 2019; Steurtewagen and van den Poel, 2020). In general, soft sensor techniques consist of two major types, namely, model- and data-driven methods. Model-driven soft sensors are traditionally developed based

‡ Corresponding authors
* Project supported by the National Key Research and Development Program of China (No. 2016YFC0301404), the National Natural Science Foundation of China (Nos. 51379198 and 61903352), the Natural Science Foundation of Zhejiang Province, China (No. LQ19F030007), the Natural Science Foundation of Jiangsu Province, China (No. BK20180594), the Project of Department of Education of Zhejiang Province, China (No. Y202044960), the China Postdoctoral Science Foundation (No. 2020M671721), the Foundation of Key Laboratory of Advanced Process Control for Light Industry (No. APCLI1803), and the Fundamental Research Funds for the Provincial Universities of Zhejiang, China (Nos. 2021YW18 and 2021YW80)
ORCID: Yuxue XU, https://orcid.org/0000-0003-0144-6015; Yun WANG, https://orcid.org/0000-0002-7512-0168; Tianhong YAN, https://orcid.org/0000-0003-3916-3926; Yuchen HE, https://orcid.org/0000-0002-0528-2778

on first-principle models (FPMs) and require in-depth knowledge of the target process (Yuan et al., 2018). However, they usually become invalid under complex non-stationary conditions. Because of the complexity of industrial processes, model-driven soft sensors are usually impractical, cost-prohibitive, and time-consuming. As a solution, data-driven soft sensors have gained ever-increasing popularity in the process industry. This method regards the object as a black box and builds a model based on real process data directly collected from plants. The advantage of the data-driven method is that it does not need to describe the physical or chemical background of the process, but needs only to obtain enough data to establish the model (Kadlec et al., 2009; Bidar et al., 2017; Ge et al., 2017; Liu Q et al., 2018; Montáns et al., 2019). The most popular data-driven modeling approaches are principal component analysis (PCA) and partial least squares (PLS). These are often used to reduce the dimension of process data and applied in linear regression (Vallejo et al., 2019).

Nonlinear relationships among different process variables are very common in the process industry which usually contains complex structures. Therefore, data-driven methods are designed to describe the nonlinear relationships between process variables and quality variables. As a well-established method of nonlinear function approximation, locally weighted learning (LWL) builds locally linear models in the neighborhood of the query sample and uses these local models to realize nonlinear estimation. Typical LWL methods include locally weighted linear regression (LWLR), locally weighted principal component analysis (LWPCA), and locally weighted partial least squares (LWPLS) (Atkeson et al., 1997). To build an accurate model with LWL, the similarity needs to be properly defined. More attention has been paid to using a priori information of inputs and outputs. In general, similarity can be simply defined on the basis of the Euclidean distance between training samples and query samples (Kim S et al., 2013). Clearly, this method may lead to the deviation of similarity calculations since it ignores the output information in training samples. A simple solution is to predict the output of the query samples through an appropriate model, so that the similarities can be obtained by the full content of training samples and the query ones in the original dataset space. However, an appropriate model is required to predict the output of query samples, which undoubtedly increases the complexity and uncertainty of the overall model (Atkeson et al., 1997; Chang et al., 2001). Hence, Yuan et al. (2016) proposed a new similarity calculation method in a supervised hidden space constructed from a PLS algorithm. The supervised data-driven model tries to build a relationship between fully labeled input and output datasets, which is suitable for the regression situation such as soft sensor development (Shao et al., 2015; Yao and Ge, 2017b). Then the latent variables closely related to inputs and outputs can be extracted in the low-dimensional hidden space. Finally, the traditional similarity measurement can be applied to the latent information. The above structure makes it possible to perform a similarity calculation by using all available information while little extra noise is added (Yuan et al., 2014). In addition, factor analysis (FA) can be used to construct a supervised hidden space and perform locally weighted learning (Yao and Ge, 2017a). However, it is true that out of the change of measurement and transmission interference, process data are often disturbed by noise and uncertainty. As common dimensionality reduction methods, PLS and FA extract latent variables linearly, so processing data with noise and uncertain interference cannot be handled well.

To cope with noise interference, process variables are reasonably treated as random variables to construct a model in a probability frame (Ge, 2018; Wang et al., 2019). Bayesian networks (BNs) are considered as combinations of probabilistic methods and graphical models. BNs have the advantages of dealing with noise interference and causal learning. In recent years, BNs have been widely used in medical diagnosis, statistical decision, expert systems, and other fields (Liu F et al., 2006; Chen and Yu, 2007; Tang and Liu, 2007; Nie et al., 2018). The locally weighted naive Bayesian model and distance measures have been applied in many ways (Frank et al., 2000; Jiang et al., 2013; Li et al., 2014). Recently, an adaptive soft sensor combining BN with a locally weighted method has been proposed (Liu ZW et al., 2018). The locally weighted BN (LWBN) uses the data from the original space for similarity calculation. From the previous discussion of similarity in LWL, this method of similarity calculation is not perfect. Fortunately, we can introduce hidden space in

BNs. The introduction of latent variables can improve the causal relationship explanation and make knowledge representation more effective. Consequently, the BN with latent variables has been applied in ranking (Kim JS and Jun, 2013), fault classification (Mohammadi et al., 2019), risk assessment (Masmoudi et al., 2019), medical diagnosis (Cain, 2016), etc. In addition, it can be used to interpret FA, PCA, and mixture of experts (MOE) according to the graphical models with different structures and parameters (Bishop, 1998). In this study, a supervised BN (SBN) is proposed to extract latent information in complex industrial processes. Different from traditional BNs, the SBN structure adds latent variable nodes between process variable nodes and quality variable nodes where the process variables $\boldsymbol{X}$ and the quality variables $\boldsymbol{Y}$ are considered as the input at the corresponding node. That is why the SBN is a quality-related structure. The supervision of the quality variables $\boldsymbol{Y}$ of training samples is included in the network, which guarantees the reliability of similarity measurement.

Most industrial processes also exhibit time-varying characteristics because of unmeasured changes in raw materials, product demands, catalyst deactivation, and seasonal effects. Operating conditions are constantly perturbed. However, the similarity measurement in the latent space is implemented under a strong assumption that latent models should be applied to the non-stationary system first, which seems infeasible to traditional BNs. To improve the soft sensing performance, the traditional moving window (MW) method is often employed (Kadlec et al., 2011). In this study, a locally weighted supervised BN (LWSBN) for soft sensors is proposed, and a double-layer similarity measurement strategy is also designed. The global similarity is composed of local similarity and corresponding window confidence to maintain variable-wise relationship within the locally weighted strategy.

Based on the above introduction, the contributions of this study are as follows:

1. To extract latent information in complex process variables, an SBN is proposed.

2. To improve the performance of soft sensing in non-stationary systems, a double-layer similarity measurement strategy based on latent variables is proposed.

2 Preliminaries

BN is one of the most popular probabilistic graphical models, often employed to represent knowledge about an uncertain domain. Each node in the graph represents a random variable where the edges between each pair of nodes indicate corresponding probabilistic dependencies (Ben-Gal, 2008). BNs can be parameterized by specifying all the local conditional probability distributions (CPDs) (Murphy, 2001).

The probabilistic relations among nodes can be clearly reflected through the networks. Theoretical research on BN includes mainly the learning method and inference mechanism. Bayesian learning includes two cases, i.e., parameter learning and structure learning, which refer to the process of obtaining a BN by analyzing data. If the structure of a BN is known, the expectation maximum (EM) algorithm is introduced to estimate the optimal parameters. The EM algorithm is an effective method for giving the maximum likelihood estimation (MLE) through multiple iterations. Then the BN obtains useful information by inference. The inference of BN refers to inquiring the posterior information of some nodes under given evidence conditions. One of the most widely used inference algorithms is named junction tree, which is famous for high accuracy and efficiency. In this study, junction tree inference is used to calculate the maximum posterior probability distribution.

The basic idea of the locally weighted method is to establish the model by BN on the neighborhood of the query sample rather than on the whole training sample. Local learning helps mitigate the effects of attribute dependencies that may exist in the data as a whole (Frank et al., 2002; Wu et al., 2015). Suppose that process variables $\boldsymbol{X}=(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)\in\mathbb{R}^{5\times n}$ and quality variables $\boldsymbol{Y}=(y_1, y_2, \ldots, y_n)\in\mathbb{R}^{1\times n}$ are used for soft sensing, where n is the number of samples in the dataset. Then the structure of the BN for soft measurement is as shown in Fig. 1, and CPDs for all nodes are linear Gaussian (Geiger and Heckerman, 1994).

To implement locally weighted soft sensing, the similarities between training samples and the query sample are measured by the Euclidean distance (d) defined as follows. These measures serve as weights assigned to the original training data.

$$d=\sqrt{(\boldsymbol{x}_{\mathrm{n}}-\boldsymbol{x}_{\mathrm{q}})^{\mathrm{T}}(\boldsymbol{x}_{\mathrm{n}}-\boldsymbol{x}_{\mathrm{q}})}, \tag{1}$$

$$w=\exp\left(-\frac{\varphi d}{\sigma_d}\right), \tag{2}$$

where $\boldsymbol{x}_{\mathrm{n}}$ and $\boldsymbol{x}_{\mathrm{q}}$ represent the training sample and query sample respectively, φ is a tuning parameter, and σ_d is the standard deviation of d in Eq. (1). Usually, training samples with the least distance from the newest query sample are selected to build the LWBN model (Liu ZW et al., 2018).

When a query variable $\boldsymbol{x}_{\mathrm{q}}$ arrives, the LWBN soft sensing for prediction of quality variable y_{q} is as follows:

Step 1: select the closest K training samples from the historical dataset based on Eq. (1), and assign weights according to Eq. (2).

Step 2: input the weighted data into the BN for parameter learning.

Step 3: calculate the posterior distribution of y_{q} according to the query sample $\boldsymbol{x}_{\mathrm{q}}$.

Step 4: discard the existing model.

When new query variables come, repeat the above steps until all query samples are predicted.

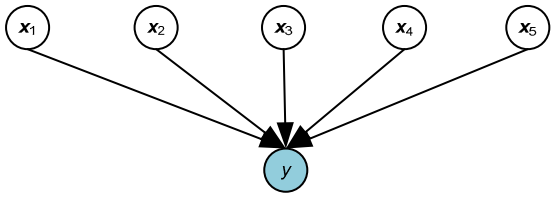

Fig. 1 The Bayesian network between process variables and quality variables in the soft sensor model

3 Bayesian network with latent variables

3.1 Latent variables

Latent variables are usually unobservable and can be inferred by mathematical models based on the observed data. This kind of variable is often introduced into a graphical model to describe the complex dependencies among variables. By introducing latent variables, the number of degrees of freedom within the model can be controlled, while still allowing correlations to be captured. In general, the goal of the latent variable model is to use a smaller number of latent variables $\boldsymbol{z}=(z_1, z_2, \ldots, z_m)$ to depict the distribution $P(\boldsymbol{u})$ of the variables $\boldsymbol{u}=(u_1, u_2, \ldots, u_n)$, where $m<n$. This is achieved by first decomposing the joint distribution $P(\boldsymbol{u}, \boldsymbol{z})$ into the product of the marginal distribution $P(\boldsymbol{z})$ of the latent variables and the conditional distribution $P(\boldsymbol{u}|\boldsymbol{z})$ of the data variables given the latent variables. It is often convenient to assume that the conditional distribution factorizes over the data variables, so that the joint distribution becomes (Bishop, 1998)

$$P(\boldsymbol{u},\boldsymbol{z})=P(\boldsymbol{z})P(\boldsymbol{u}\,|\,\boldsymbol{z})=p(\boldsymbol{z})\prod_{i=1}^{n}P(u_i\,|\,\boldsymbol{z}). \tag{3}$$

This factorization property can be graphically represented by a BN (Fig. 2), where the same graph is drawn using the plate notation at the right side of the arrow. In addition, the conditional distribution $P(\boldsymbol{u}|\boldsymbol{z})$ can be expressed in terms of a mapping from latent variables to data variables. One of the simplest latent variable models is called factor analysis and is based on linear mapping.

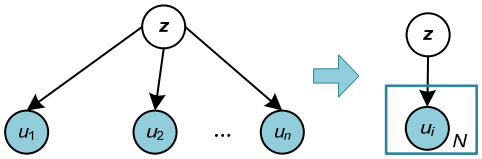

Fig. 2 Bayesian network representation of the latent variable distribution (N represents the number of times repeated by the same node)

3.2 Supervised Bayesian network

For quality-oriented information extraction, an SBN is designed in this study. The topology of an SBN including three Gaussian-distributed nodes can be represented as Fig. 3, where nodes $\boldsymbol{X}$, $\boldsymbol{T}$, and $\boldsymbol{Y}$ represent process variables, latent variables, and quality variables, respectively. The purpose of designing a quality-related BN is to solve the similarity calculation problem caused by the introduction of a locally weighted method. From Yuan et al. (2014), we know that the similarity calculated by latent variables usually contains more known information.

Unlike the structure of MOE (Bishop, 1998), node $\boldsymbol{T}$ in an SBN does not serve as the gate function for regression calculation. Instead, parameter learning for node $\boldsymbol{T}$ is designed to extract the hidden features.

In other words, node $\boldsymbol{T}$ in the SBN helps take quality-related information into hidden layer construction. The joint probability involved in this structure is as follows:

$$P(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{T}) = P(\boldsymbol{X})P(\boldsymbol{T} \mid \boldsymbol{X})P(\boldsymbol{Y} \mid \boldsymbol{X},\boldsymbol{T}). \quad (4)$$

Once the BN is set up, training samples put into the network are used to learn parameters of latent variables. Inference is included in the SBN model where marginal distribution or maximum probability of variables is inferred by new evidence. To obtain the latent variables and their corresponding nodes, both $\boldsymbol{X}$ and $\boldsymbol{Y}$ of the training samples are considered as the model input. That is why this structure is called supervised BN. Query samples are input for inferring the corresponding results so that node $\boldsymbol{T}$ contains the input and output information of the training samples, as well as the input information of query samples.

To improve the soft sensing performance of the SBN for the non-stationary process, a double-layer similarity measurement strategy is developed based on latent variables, where the original training samples are weighted for parameter learning.

3.3 Calculation of latent variables

From the structure of the SBN proposed above, latent variables can be extracted from process variables and quality variables of the training samples. For simplicity, process variables and quality variables are defined as $\boldsymbol{X}=(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)\in\mathbb{R}^{m\times n}$ and $\boldsymbol{Y}=(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n)\in\mathbb{R}^{k\times n}$ respectively, where m, k, and n are the numbers of process variables $\boldsymbol{X}$, quality variables $\boldsymbol{Y}$, and samples contained in the dataset, respectively. When a new query sample arrives, the training dataset including $\boldsymbol{X}$ and $\boldsymbol{Y}$ is preset in the moving window first as in other studies. If the number of samples contained in each window is w and the step of windows moving is s, then the number of samples traversed is $W=ws$; so, the training set corresponding to each query sample $\boldsymbol{x}_{\mathrm{q}}\in\boldsymbol{X}_{\mathrm{query}}$ is $\boldsymbol{D}_{\mathrm{train}}=(\boldsymbol{X}_{\mathrm{train}}, \boldsymbol{Y}_{\mathrm{train}})\in\mathbb{R}^{(m+k)\times W}$. Then the training set $\boldsymbol{D}_{\mathrm{train}}$ is used to train the quality-related BN, whose structure is explained in Section 3.2. Finally, latent variables are inferred by the quality-related BN with the EM algorithm. The schematic of latent variable calculation is shown in Fig. 4. Supposing that the r^{th} training sample belongs to the v^{th} moving window, the latent variables $\boldsymbol{t}_r$ can be calculated by reasoning after training sample $\boldsymbol{x}_r$ is input to the BN as evidence. Therefore, according to $\boldsymbol{X}_{\mathrm{train}}$, the latent variables $\boldsymbol{T}\in\mathbb{R}^{W\times i}$ corresponding to training dataset $\boldsymbol{D}_{\mathrm{train}}$ can be calculated, where i is the number of latent variables. Then new process variables $\boldsymbol{x}_{\mathrm{q}}$ (as query samples) are put into node $\boldsymbol{X}$ to obtain latent variables $\boldsymbol{t}_{\mathrm{new}}\in\mathbb{R}^{1\times i}$ corresponding to $\boldsymbol{x}_{\mathrm{q}}$.

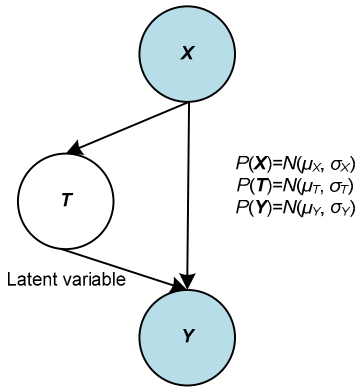

Fig. 3 Structure of a supervised Bayesian network with latent variables

4 Locally weighted soft sensing by supervised Bayesian network

4.1 Calculation of similarity

A locally weighted approach is introduced to cope with the time-varying characteristics in the process. Different from the similarity measurement mentioned in Section 2, latent variables are taken into consideration where quality-oriented information can be involved in soft sensor construction. For non-stationary process soft sensing, a moving window strategy is also adopted to build a local area where the SBN can be applied and further transferred into a non-stationary process application through the double-layer similarity measurement structure. This structure is composed of three major parts including sample local similarity, window confidence, and integration similarity. This will be explained in detail in the following. The schematic of similarity calculation is shown in Fig. 5.

1. Sample local similarity

In each moving window, the SBN is introduced

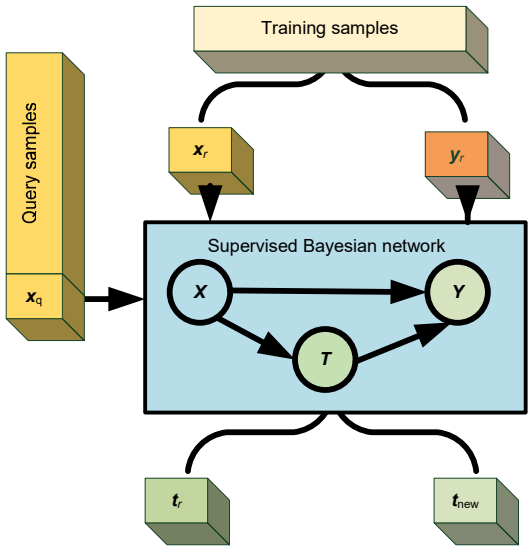

Fig. 4 Schematic of latent variable calculation

to construct a local latent space so that the quality-oriented information can be employed for sample similarity measurement. Here the Euclidean distance between the latent variables is considered as the sample similarity. In the v^{th} window, the sample local similarity can be represented as follows:

$$d_v = \sqrt{(\boldsymbol{t}_{new} - \boldsymbol{t}_r)^T (\boldsymbol{t}_{new} - \boldsymbol{t}_r)}, \tag{5}$$

$$S_v = \exp\left(-\frac{\varphi d_v}{\sigma_d}\right), \tag{6}$$

where φ is the tuning parameter with a value between 0 and 1 and σ_d is the standard deviation of d_v. Note that the value of similarity S_v is obtained mainly under the assumption of the SBN model based on the moving window data. It is a local value and cannot be used directly for the weighted structure. Hence, the weights of the moving window should also be taken into consideration to avoid incomplete data description.

2. Window confidence

A moving window strategy is always considered as a powerful solution for non-stationary process problems. It should be noted that one training sample may be assigned to different windows at the same time, which means that the sample local similarity is highly dependent on the confidence of each window. Local information can be well captured using the moving window strategy, and an overlapping structure can also guarantee the consistency of process

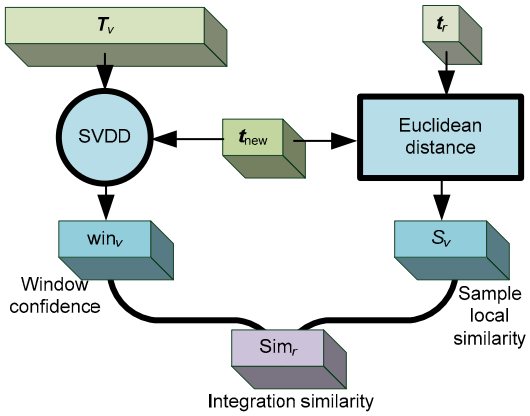

Fig. 5 Schematic of similarity calculation

modeling. Window confidence is therefore designed to evaluate the authenticity of the local similarity corresponding to one window. In this study, window confidence is defined by the support vector data description (SVDD) method, which maps the raw data to a higher-dimensional space and finds a hypersphere to envelope the raw data in the training set. The v^{th} window confidence is defined as

$$\text{win}_v = R_v \Big/ \left\| \boldsymbol{t}_v - \boldsymbol{a}_v \right\|^2, \tag{7}$$

where $\boldsymbol{a}_v$ and R_v represent the central coordinates and radius of the hypersphere constructed according to the latent variables $\boldsymbol{T}_v$ calculated in the v^{th} window, respectively. $\boldsymbol{t}_v$ represents the corresponding coordinates of $\boldsymbol{t}_{new}$ in the hypersphere.

3. Integration similarity

To obtain the final similarity of one training sample with respect to the current query sample, the local information of different windows should be integrated. As a result, integration similarity is applied by combining the information of local similarities and window confidences. In the v^{th} window, the final similarity of the r^{th} training sample can be represented as follows:

$$\text{Sim}_r = \sum_{v=1}^{s} \text{win}_v \cdot S_v. \tag{8}$$

The integration similarity vector of training samples with respect to the current query sample can be expressed as

$$\textbf{SIM} = (\text{Sim}_1, \text{Sim}_2, ..., \text{Sim}_r, ..., \text{Sim}_W). \tag{9}$$

The local similarity will be given a large weight when the corresponding window confidence is high. Conversely, the local similarity is not reliable if the window has a small confidence with the current query sample. The integration attempts to construct global similarity by combining the sample local similarity and window confidence. This enables the application of quality-related information extraction to non-stationary processes.

The detailed procedure of similarity measurement can be summarized as Table 1. After obtaining the latent variables by the SBN, the sample local similarity is first calculated by the Euclidean distance of the latent variables corresponding to the query samples and the training samples. Then SVDD is used to calculate window confidence based on query samples and training samples of each sliding window. Finally, the window confidence and sample local similarity are integrated by multiplication to ensure that the similarity contains both sample information and variable information.

Table 1 Procedure of similarity measurement

Name	Function
Sample local similarity	The Euclidean distance between the query sample and each training sample
Window confidence	The similarity of the query sample to the whole of the window by SVDD
Integration similarity	Used to assign weights to training data

4.2 Prediction of quality variables

After double-layer similarity is calculated, the similarity is assigned to the original training data ($\boldsymbol{X}_{\text{train}}$, $\boldsymbol{Y}_{\text{train}}$):

$$\begin{cases} \boldsymbol{X}_t = \mathbf{SIM} \cdot \boldsymbol{X}_{\text{train}}, \\ \boldsymbol{Y}_t = \mathbf{SIM} \cdot \boldsymbol{Y}_{\text{train}}. \end{cases} \tag{10}$$

The weighted data $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ are substituted into nodes $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. As shown in Fig. 6, $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ are used to learn parameters. Then the query sample is input to the SBN as evidence to predict the quality variables by inference.

In conclusion, the main steps of LWSBN for quality prediction can be listed as follows (Fig. 7):

Step 1: introduce the moving window strategy to split the dataset.

Step 2: establish an SBN based on local training samples and obtain the corresponding latent variables.

Step 3: introduce the query sample into the above SBN to obtain the corresponding latent variables.

Step 4: calculate the integration similarity based on latent variables according to Eq. (9) and assign it to the corresponding training samples.

Step 5: obtain a locally weighted SBN based on the weighted training samples.

Step 6: obtain the predicted value of the quality variable by introducing the query sample into the network as evidence.

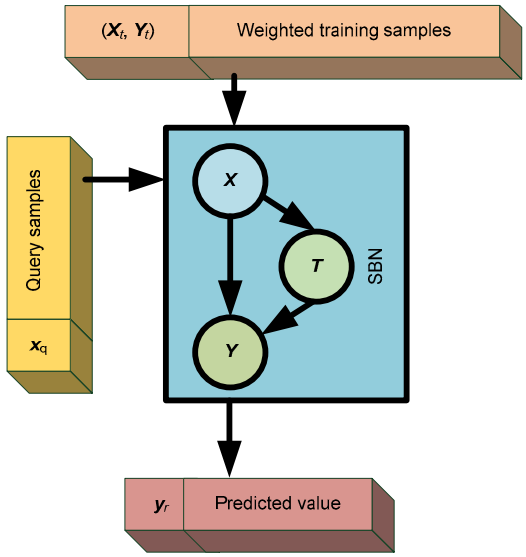

Fig. 6 Schematic of quality prediction

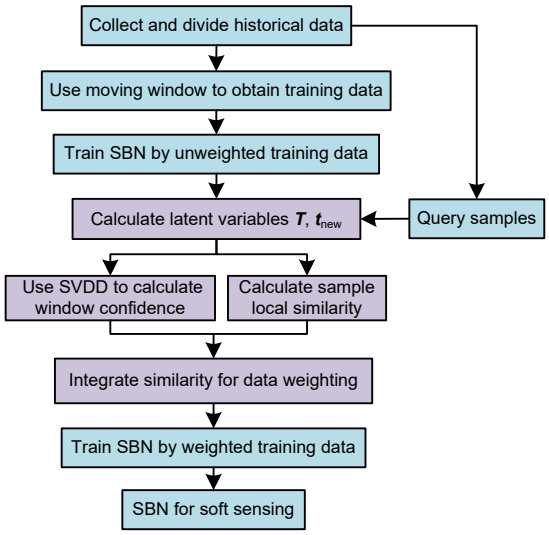

Fig. 7 Locally weighted soft sensing by a supervised Bayesian network (SBN)

Step 7: repeat steps 1–6 until quality prediction is completed for all samples.

5 Case studies

In this section, the proposed LWSBN is tested using a numerical example and a real industrial process. For comparison, the LWBN and LWLR mentioned above and a locally weighted supervised latent factor analysis model (LWSLFA) (Ge, 2016) are also tested.

5.1 Numerical example

In this numerical example, it is assumed that the dataset is generated by the following process:

$$\begin{cases} \boldsymbol{X} = \boldsymbol{A}\boldsymbol{T} + \boldsymbol{e}, \\ \boldsymbol{Y} = \boldsymbol{C}\boldsymbol{T} + \boldsymbol{f}, \end{cases} \tag{11}$$

where $\boldsymbol{A}$ is a random 6×3 matrix, $\boldsymbol{C}$ is a random 1×3 matrix, and the number of latent variables $\boldsymbol{T}$ is 3. To generate non-stationary data, $\boldsymbol{T}$ contains three modes, and the ratio of each mode is (0.20, 0.25, 0.55). In the three modes, $\boldsymbol{T}$ obeys the Gaussian distribution $\boldsymbol{T}\sim N(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$. The parameters of $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t$ are set as

$$\boldsymbol{\mu}_t = \begin{bmatrix} 0.03 & 0.1 & 0.5 \\ 2 & 6.5 & 12 \\ 1.5 & 3.3 & 1 \end{bmatrix}, \quad \boldsymbol{\sigma}_t = \begin{bmatrix} 0.5 & 0.33 & 0.24 \\ 0.2 & 0.4 & 0.61 \\ 1 & 0.32 & 0.21 \end{bmatrix}.$$

In addition, $\boldsymbol{e}\sim N(0.2, 0.1^2\boldsymbol{I})$ and $\boldsymbol{f}\sim N(0.1, \boldsymbol{I})$ are measurement noises of input and output data respectively, where $\boldsymbol{I}$ is the identity matrix. To verify the validity of the proposed soft sensor model, 1000 training samples and 200 query samples are generated according to the above rules. In every local model in LWSBN, 150 training samples are selected by the moving window. The number of samples contained in each window and the step of windows are set to be 15 and 10, respectively. In this study, we compare the experimental results of our method with those of three other methods (LWBN, LWSLFA, and LWLR). As the Euclidean distance is involved in all four models, the tuning parameter φ is set to 0.75. The prediction results of the four methods are shown in Fig. 8 with blue and red lines representing the actual values and prediction results, respectively.

Because of the absence of variable-wise correlation in the LWLR method, the corresponding performance is comparatively poor especially at some peak points. As indicated by arrows 1 and 2, there are large deviations in the prediction results of LWBN and LWSLFA, while the proposed method LWSBN can track quality variables better than the three other methods. In addition, it can be seen that some obvious errors and delays exist at the beginning of the new modes. To verify the influence of the latent variables mentioned above on the global similarity, three-dimensional (3D) graphs of the similarity are plotted in Fig. 9. Since different colors represent different similarity degrees, it is easy to see that LWSBN assigns more uniform weights to samples, so that more data information can be used. The predicted errors of the four methods are plotted in Fig. 10, in which the accuracy of the proposed method in soft sensor modeling can be demonstrated.

To compare the performance quantitatively, the root mean squared error (RMSE) and the R^2 statistic are considered. They are defined as follows:

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{\mathrm{real}}-y_{\mathrm{pred}})^2}, \tag{12}$$

$$R^2=1-\sum_{i=1}^{n}(y_{\mathrm{real}}-y_{\mathrm{pred}})^2 \Big/ \sum_{i=1}^{n}(y_{\mathrm{real}}-\bar{y}_{\mathrm{real}})^2, \tag{13}$$

where y_{real}, $\bar{y}_{\mathrm{real}}$, and y_{pred} are the real value, average value, and predicted value respectively, and n is the number of query samples. A lower RMSE value indicates a smaller prediction error, while a better

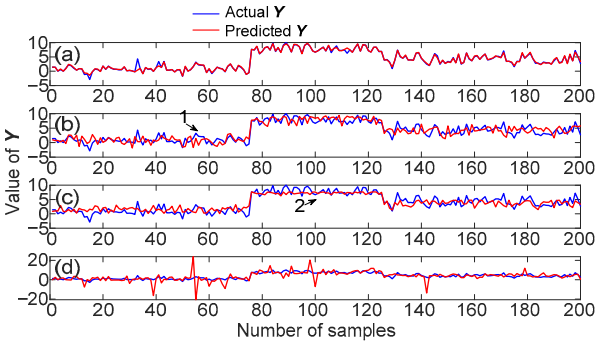

Fig. 8 Predicted Y of the four methods in a numerical example: (a) LWSBN; (b) LWBN; (c) LWSLFA; (d) LWLR

References to color refer to the online version of this figure

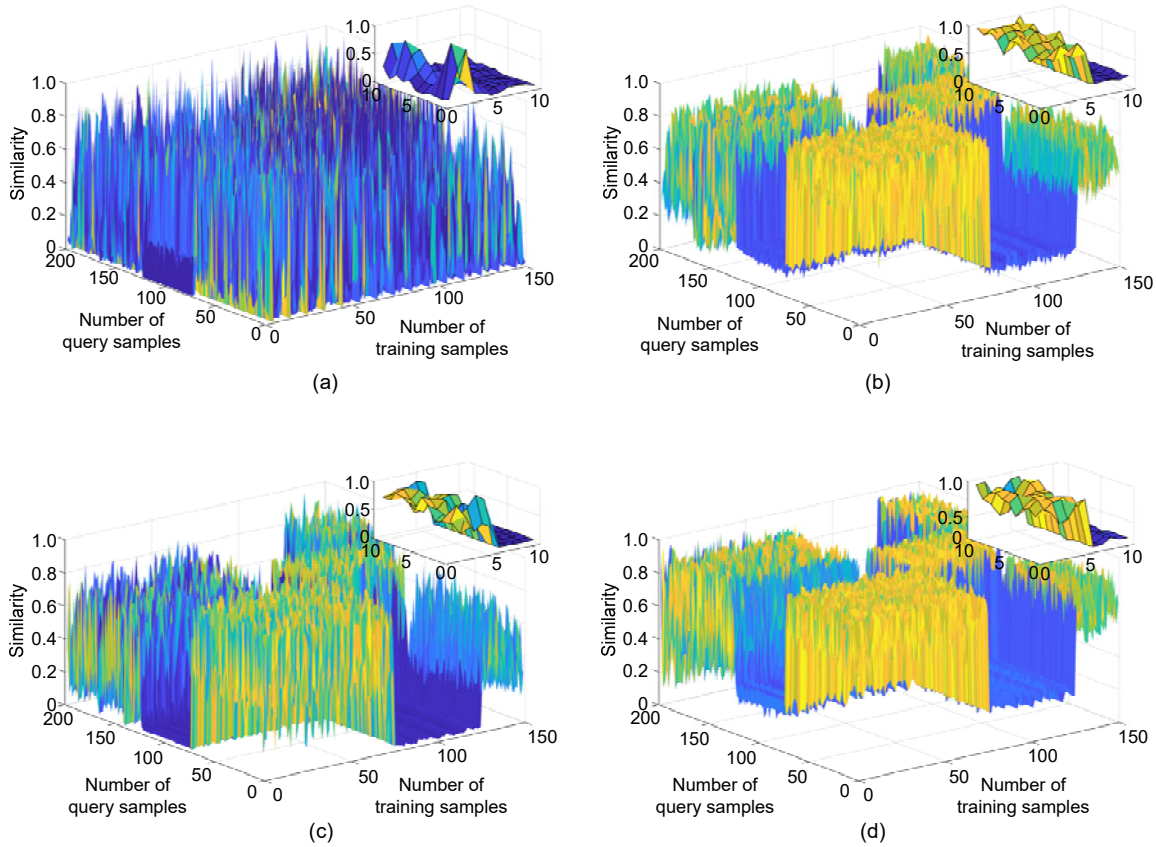

Fig. 9 Similarity diagrams in a numerical example: (a) LWSBN; (b) LWBN; (c) LWSLFA; (d) LWLR

References to color refer to the online version of this figure

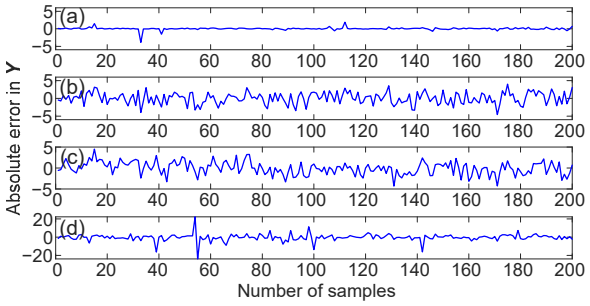

Fig. 10 Prediction of absolute errors in a numerical example: (a) LWSBN; (b) LWBN; (c) LWSLFA; (d) LWLR

accuracy can be reflected when the R^2 value is very close to 1. The RMSE and R^2 values of all methods are shown in Table 2. Compared to LWBN and LWSLFA, LWSBN achieves a better prediction result.

5.2 Industrial application: debutanizer column

In industrial refineries, it is important to optimize and test individual units and the whole process.

Table 2 Prediction of precision of the four methods in the numerical example

Index	Precision			
	LWSBN	LWBN	LWSLFA	LWLR
RMSE	0.3862	0.5225	0.9361	**0.1057**
R^2	**0.9837**	0.9712	0.9021	0.4640

RMSE: root mean squared error. The best results are in bold

This can greatly improve the safety and efficiency of products. The debutanizer column is an important part of the oil refining process. As a common soft sensor model testing platform, strong non-stationaries are included in the process data of a debutanizer column. Therefore, it is necessary to adopt a nonlinear model to obtain high prediction accuracy. The butane content in the debutanizer bottoms, which represents the soft sensor output, is measured on the overheads of the debutanizer column. The debutanizer column dataset can be obtained from a real industrial sequential distillation process and contains 2394 samples and 7 input variables (Fortuna et al., 2007), which are described in detail in Table 3.

One thousand training samples are selected and another 400 samples are collected as the query samples under the same operating conditions. For each query sample, 50 training samples are selected by using 10 moving windows of size 5 and put into the SBN for parameter learning using the EM algorithm. Similarly, to further verify the performance of the proposed method for the debutanizer column data, LWBN, LWSLFA, and LWLR are employed. The detailed prediction results of the four methods are shown in Fig. 11. To clearly show the variation trend of butane content, a semi-logarithmic diagram is plotted. The clear difference and opposite change trend are more easily seen in the positions of arrows 1 and 2 in the prediction results of LWBN and LWSLFA. LWSBN is more accurate in tracking the non-stationary system.

The similarity graphs are shown in Fig. 12. Through four mesh diagrams of different similarities, we can clearly see that different similarity calculation methods or different latent variable calculation methods produce different similarities. As can be seen from the prediction results, LWSBN has better adaptability to the fluctuation of data.

The predicted errors of these four methods are depicted in Fig. 13. We can see that the online prediction of LWSBN has a smaller absolute prediction error. In addition, R^2 and RMSE results of all methods

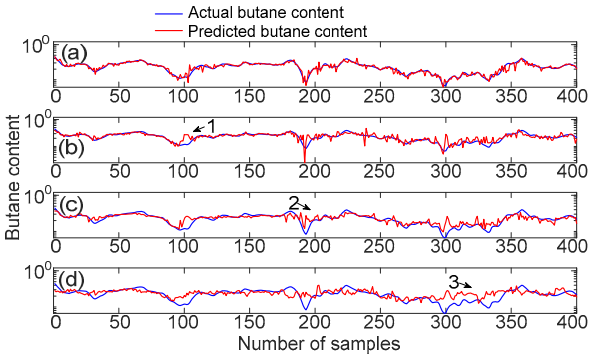

Fig. 11 Prediction results in the debutanizer column: (a) LWSBN; (b) LWBN; (c) LWSLFA; (d) LWLR
References to color refer to the online version of this figure

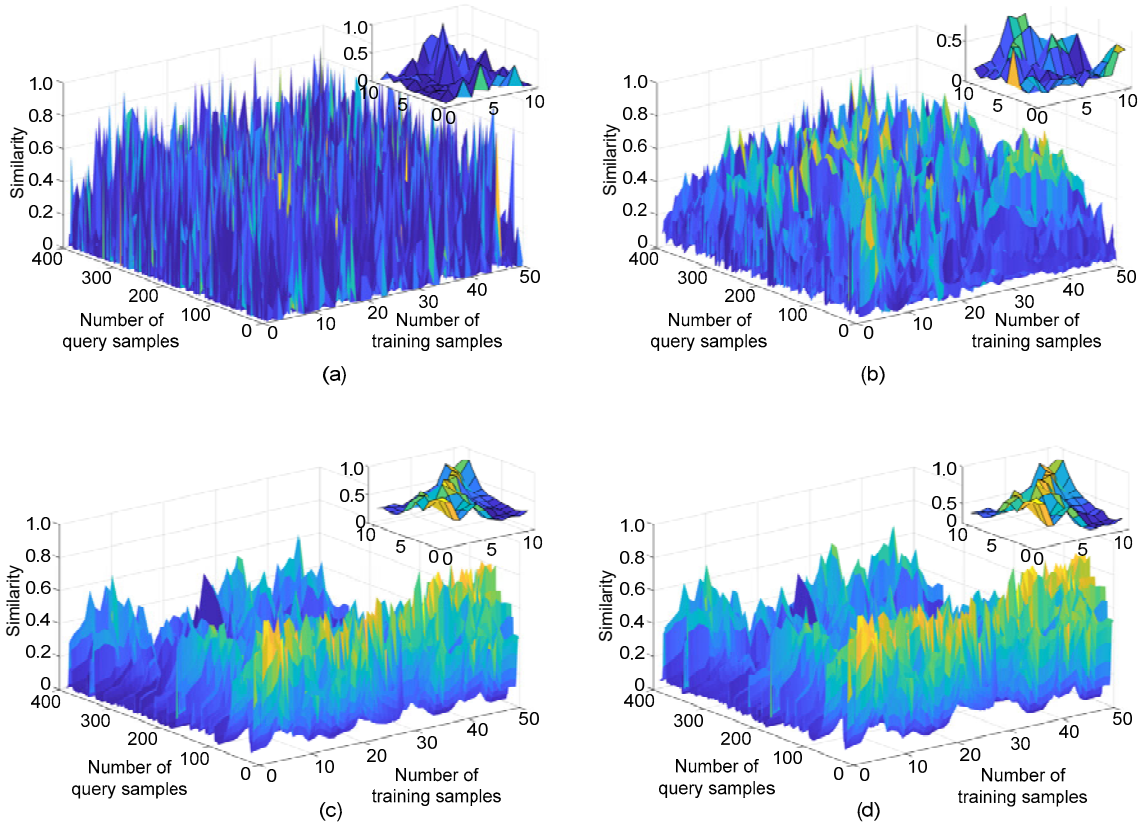

Fig. 12 Similarity diagrams in the debutanizer column: (a) LWSBN; (b) LWBN; (c) LWSLFA; (d) LWLR
References to color refer to the online version of this figure

are listed in Table 4, showing that LWSBN acquires higher prediction accuracy and better fitting effect. All results indicate the superiority of the presented LWSBN method in soft sensing of quality variable prediction.

6 Conclusions

In this study, a quality-related locally weighted soft sensing method based on a supervised BN with latent variables has been presented for non-stationary processes to precisely estimate key quality variables. Different from previous BNs, latent variables were extracted based on the SBN and a corresponding double-layer similarity calculation strategy has been proposed where the integration of local sample similarity and window confidence helps capture local and global similarity information. To test the validity of this method, a numerical example and a real case were used where the prediction accuracy and reliability of the proposed method have been emphasized against previous LWBN, LWLR, and LWSLFA. The similarity measurement method involved in this paper can further be extended to other BNs.

Table 3 List of variables used in the design of soft sensors for the debutanizer column

Variable	Description
u_1	Top temperature
u_2	Top pressure
u_3	Reflux flow
u_4	Flow to the next process
u_5	6^{th} tray temperature
u_6	Bottom temperature
u_7	Bottom pressure

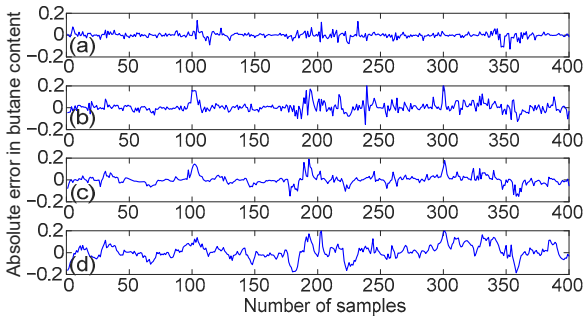

Fig. 13 Prediction of absolute errors in the debutanizer column: (a) LWSBN; (b) LWBN; (c) LWSLFA; (d) LWLR

Table 4 Prediction of precision of the four methods in the debutanizer column

Index	Precision			
	LWSBN	LWBN	LWSLFA	LWLR
RMSE	**0.0321**	0.0525	0.0438	0.0936
R^2	**0.9506**	0.8679	0.9077	0.5790

RMSE: root mean squared error. The best results are in bold

Contributors

Yuxue XU and Yuchen HE designed the research. Yuchen HE and Yun WANG processed the data. Yuchen HE, Tianhong YAN, Haiping DU, and Weihua LI drafted the manuscript. Jun WANG helped polish the language. De GU helped organize the manuscript. Yuxue XU and Yuchen HE revised and finalized the paper.

Compliance with ethics guidelines

Yuxue XU, Yun WANG, Tianhong YAN, Yuchen HE, Jun WANG, De GU, Haiping DU, and Weihua LI declare that they have no conflict of interest.

References

Al-Jlibawi A, Othman MLB, Al-Huseiny MS, et al., 2019. Efficient soft sensor modelling for advanced manufacturing systems by applying hybrid intelligent soft computing techniques. *Int J Simul Syst Sci Technol*, 19(3): 15.1-15.7. https://doi.org/10.5013/IJSSST.a.19.03.15

Atkeson CG, Moore AW, Schaal S, 1997. Locally weighted learning. In: Aha DW (Ed.), Lazy Learning. Springer, Dordrecht, p.11-73. https://doi.org/10.1007/978-94-017-2053-3_2

Ben-Gal I, 2008. Bayesian networks. In: Ruggeri F, Kenett RS, Faltin FW (Eds.), Encyclopedia of Statistics in Quality and Reliability. John Wiley & Sons, Chichester, p.1. https://doi.org/10.1002/9780470061572.eqr089

Bidar B, Sadeghi J, Shahraki F, et al., 2017. Data-driven soft sensor approach for online quality prediction using state dependent parameter models. *Chemom Intell Lab Syst*, 162:130-141. https://doi.org/10.1016/j.chemolab.2017.01.004

Bishop CM, 1998. Latent variable models. In: Jordan MI (Ed.), Learning in Graphical Models. MIT Press, Cambridge, p.371-403.

Cain C, 2016. Modelling latent variables for Bayesian networks. Undergraduate Research Symp. Available from https://digitalcommons.morris.umn.edu/urs_2016/2/

Chang SY, Baughman EH, McIntosh BC, 2001. Implementation of locally weighted regression to maintain calibrations on FT-NIR analyzers for industrial processes. *Appl Spectrosc*, 55(9):1199-1206. https://doi.org/10.1366/0003702011953207

Chen GF, Yu HL, 2007. Bayesian network and its application in maize diseases diagnosis. Proc Int Conf on Computer and Computing Technologies in Agriculture, p.917-924. https://doi.org/10.1007/978-0-387-77253-0_22

Fortuna L, Graziani S, Rizzo A, et al., 2007. Soft Sensors for Monitoring and Control of Industrial Processes. Springer, London, UK. https://doi.org/10.1007/978-1-84628-480-9

Frank E, Trigg L, Holmes G, et al., 2000. Technical note: naive Bayes for regression. *Mach Learn*, 41(1):5-25. https://doi.org/10.1023/A:1007670802811

Frank E, Hall M, Pfahringer B, 2002. Locally weighted naive Bayes. Proc 19th Conf on Uncertainty in Artificial Intelligence, p.249-256.

Ge ZQ, 2016. Supervised latent factor analysis for process data regression modeling and soft sensor application. *IEEE Trans Contr Syst Technol*, 24(3):1004-1011. https://doi.org/10.1109/TCST.2015.2473817

Ge ZQ, 2018. Process data analytics via probabilistic latent variable models: a tutorial review. *Ind Eng Chem Res*, 57(38):12646-12661. https://doi.org/10.1021/acs.iecr.8b02913

Ge ZQ, Song ZH, Ding SX, et al., 2017. Data mining and analytics in the process industry: the role of machine learning. *IEEE Access*, 5:20590-20616. https://doi.org/10.1109/ACCESS.2017.2756872

Geiger D, Heckerman D, 1994. Learning Gaussian networks. In: de Mantaras RL, Poole D (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the Tenth Conference (1994). Morgan Kaufmann Publishers, California, USA, p.235-243. https://doi.org/10.1016/B978-1-55860-332-5.50035-3

Jiang LX, Cai ZH, Zhang H, et al., 2013. Naive Bayes text classifiers: a locally weighted learning approach. *J Exp Theor Artif Intell*, 25(2):273-286. https://doi.org/10.1080/0952813X.2012.721010

Kadlec P, Gabrys B, Strandt S, 2009. Data-driven soft sensors in the process industry. *Comput Chem Eng*, 33(4):795-814. https://doi.org/10.1016/j.compchemeng.2008.12.012

Kadlec P, Grbić R, Gabrys B, 2011. Review of adaptation mechanisms for data-driven soft sensors. *Comput Chem Eng*, 35(1):1-24. https://doi.org/10.1016/j.compchemeng.2010.07.034

Kano M, Fujiwara K, 2012. Virtual sensing technology in process industries: trends and challenges revealed by recent industrial applications. *J Chem Eng Jpn*, 46(1):1-17. https://doi.org/10.1252/jcej.12we167

Kim JS, Jun CH, 2013. Ranking evaluation of institutions based on a Bayesian network having a latent variable. *Knowl-Based Syst*, 50:87-99. https://doi.org/10.1016/j.knosys.2013.05.010

Kim S, Okajima R, Kano M, et al., 2013. Development of soft-sensor using locally weighted PLS with adaptive similarity measure. *Chemom Intell Lab Syst*, 124:43-49. https://doi.org/10.1016/j.chemolab.2013.03.008

Li CQ, Jiang LX, Li HW, 2014. Naive Bayes for value difference metric. *Front Comput Sci*, 8(2):255-264. https://doi.org/10.1007/s11704-014-3038-5

Liu F, Xu DX, Yuan C, et al., 2006. Image segmentation based on Bayesian network-Markov random field model and its application to in vivo plaque composition. Proc 3rd IEEE Int Symp on Biomedical Imaging: Nano to Macro, p.141-144. https://doi.org/10.1109/ISBI.2006.1624872

Liu Q, Zhuo J, Lang ZQ, et al., 2018. Perspectives on data-driven operation monitoring and self-optimization of industrial processes. *Acta Autom Sin*, 44(11):1944-1956 (in Chinese). https://doi.org/10.16383/j.aas.2018.c180207

Liu ZW, Ge ZQ, Chen GJ, et al., 2018. Adaptive soft sensors for quality prediction under the framework of Bayesian network. *Contr Eng Pract*, 72:19-28. https://doi.org/10.1016/j.conengprac.2017.10.018

Masmoudi K, Abid L, Masmoudi A, 2019. Credit risk modeling using Bayesian network with a latent variable. *Expert Syst Appl*, 127:157-166. https://doi.org/10.1016/j.eswa.2019.03.014

Mohammadi A, Zarghami R, Lefebvre D, et al., 2019. Soft sensor design and fault detection using Bayesian network and probabilistic principal component analysis. *J Adv Manuf Process*, 1(4):e10027. https://doi.org/10.1002/amp2.10027

Montáns FJ, Chinesta F, Gómez-Bombarelli R, et al., 2019. Data-driven modeling and learning in science and engineering. *Compt Rend Mécan*, 347(11):845-855. https://doi.org/10.1016/j.crme.2019.11.009

Murphy KP, 2001. The Bayes net toolbox for Matlab. *Comput Sci Stat*, 33(2):1024-1034.

Nie SQ, Zheng M, Ji Q, 2018. The deep regression Bayesian network and its applications: probabilistic deep learning for computer vision. *IEEE Signal Process Mag*, 35(1):101-111. https://doi.org/10.1109/MSP.2017.2763440

Shao WM, Tian XM, Wang P, 2015. Supervised local and non-local structure preserving projections with application to just-in-time learning for adaptive soft sensor. *Chin J Chem Eng*, 23(12):1925-1934. https://doi.org/10.1016/j.cjche.2015.11.012

Steurtewagen B, van den Poel D, 2020. Machine learning refinery sensor data to predict catalyst saturation levels. *Comput Chem Eng*, 134:106722. https://doi.org/10.1016/j.compchemeng.2020.106722

Tang H, Liu S, 2007. Basic theory of fuzzy Bayesian networks and its application in machinery fault diagnosis. Proc 4th Int Conf on Fuzzy Systems and Knowledge Discovery, p.132-137. https://doi.org/10.1109/FSKD.2007.202

Vallejo M, de la Espriella C, Gómez-Santamaría J, et al., 2019. Soft metrology based on machine learning: a review. *Meas Sci Technol*, 31(3):032001. https://doi.org/10.1088/1361-6501/ab4b39

Wang JB, Shao WM, Song ZH, 2019. Semi-supervised variational Bayesian student's *t* mixture regression and robust inferential sensor application. *Contr Eng Pract*, 92:

104155. https://doi.org/10.1016/j.conengprac.2019.104155

Wu J, Wu B, Pan SR, et al., 2015. Locally weighted learning: how and when does it work in Bayesian networks? *Int J Comput Intell Syst*, 8(S1):63-74. https://doi.org/10.1080/18756891.2015.1129579

Yao L, Ge ZQ, 2017a. Locally weighted prediction methods for latent factor analysis with supervised and semisupervised process data. *IEEE Trans Autom Sci Eng*, 14(1): 126-138. https://doi.org/10.1109/TASE.2016.2608914

Yao L, Ge ZQ, 2017b. Moving window adaptive soft sensor for state shifting process based on weighted supervised latent factor analysis. *Contr Eng Pract*, 61:72-80. https://doi.org/10.1016/j.conengprac.2017.02.002

Yuan XF, Ge ZQ, Song ZH, 2014. Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes. *Ind Eng Chem Res*, 53(35):13736-13749. https://doi.org/10.1021/ie4041252

Yuan XF, Huang B, Ge ZQ, et al., 2016. Double locally weighted principal component regression for soft sensor with sample selection under supervised latent structure. *Chemom Intell Lab Syst*, 153:116-125. https://doi.org/10.1016/j.chemolab.2016.02.014

Yuan XF, Huang B, Wang YL, et al., 2018. Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. *IEEE Trans Ind Inform*, 14(7):3235-3243. https://doi.org/10.1109/TII.2018.2809730