IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. 31 (2020) 055101 (11pp) https://doi.org/10.1088/1361-6501/ab5f1b

Locally weighted slow feature regression for nonlinear dynamic soft sensor modeling and its application to an industrial hydrocracking process

Xiaofeng Yuan, Jiao Zhou and Yalin Wang[1]

School of Automation, Central South University, Changsha 410083, People's Republic of China

E-mail: ylwang@csu.edu.cn

Received 21 July 2019, revised 22 November 2019
Accepted for publication 5 December 2019
Published 31 January 2020

Abstract

Latent variable (LV) models have been extensively constructed to obtain informative low-dimensional features for process soft sensors. However, static LV models are more often adopted, which cannot describe the process dynamics. Recently, slow feature analysis and its regression model (SFR) have been introduced for dynamic LV modeling in industrial processes. However, linear SFR is limited in its modeling capacity because most industrial processes have time-varying and nonlinear characteristics. To alleviate this problem, a novel locally weighted slow feature regression (LWSFR) is proposed in this paper for nonlinear dynamic modeling. Unlike other static locally weighted learning, two weighting techniques are designed in LWSFR. First, sample weighting is used to deal with static nonlinear relationships based on Euclidean distance. Temporal weighting for the first-order sample time difference is then designed to locally linearize the nonlinear dynamics. The effectiveness of the proposed method is validated on an industrial hydrocracking process.

Keywords: soft sensor, latent variable model, locally weighted learning, slow feature analysis, quality prediction

(Some figures may appear in colour only in the online journal)

1. Introduction

In modern industrial plants, soft sensor techniques have been widely applied since they are important for online process monitoring, the predicting of quality variability, and control strategy design [1–10]. On some occasions, soft sensors are used as alternatives or backups for hardware sensors, due to their good environmental adaptability, low maintenance cost, and real-time estimation capabilities. In particular, data-driven soft sensor models have been extensively developed in recent decades since the cost of data collection and storage has fallen and the volume of data has grown in modern process plants [11–14]. To construct data-driven soft sensors, certain easily measured process auxiliary variables are used to estimate the key variables that are difficult to measure, by constructing mathematical predictive models. In recent decades, various soft sensor methods have been applied in a number of industries. Among them, principal component regression (PCR) [15, 16] and partial least squares regression (PLSR) [17–19] are the most popular linear modeling methods, while support vector machines (SVMs) [20, 21] and artificial neural networks (ANNs) [22, 23] are the most commonly used nonlinear types. The recently developed deep learning has also been widely used for soft sensor applications [24, 25].

Typically, process data often presents high collinearities and strong correlations, because industrial plants are usually installed with a number of redundant sensors on site. If they are directly used to construct the regression models, it can cause algorithm instability, poor robustness, and inaccurate performance of the soft sensor models. To handle the correlations

[1] Author to whom any correspondence should be addressed.

 © 2020 IOP Publishing Ltd Printed in the UK

of process data, latent variable (LV) models are widely used to extract useful low-dimensional features from the original high-dimensional variables [26, 27]. In LV models, the input and output are linked by their shared latent variables. The LVs are a kind of abstraction and generalization of the original process variables, which directly affect the performance of the subsequent prediction. LV models can greatly reduce model complexity and improve the accuracy of soft sensors. Most LV models are based on the assumptions that the operation state of industrial processes is stable and that the collected data are independent and identically distributed (i.i.d.).

As a matter of fact, the static i.i.d. assumption is often not met in industrial data. On the contrary, in industrial processes there are often highly dynamic characteristics which change slowly. The temporal correlations between process data can affect the structure of soft sensor models. Naturally, it is necessary to develop LV modeling methods that can deal with the dynamic characteristics. Recently, slow feature analysis (SFA), an important dynamic LV model proposed by Wiskott and Sejnowski [28], has been used to learn about time-related features referred to as slow features (SFs) [29–32]. It is necessary that the SFs have minimum temporal variations, which means that the change of SFs should vary as slowly as possible. The invariant attributes of varying signals are very useful because most variances in process data, such as disturbances to the environment, derive from inherent 'common causes' which vary quickly. SFA can extract LVs that are not related to each other and with different variation frequencies, when the inputs are driven by some inherent auto-correlation tendencies. Thus, SFA can be used to learn about dynamic data features in order to build a regression model for the predicting of quality.

In addition to the dynamic characteristic, nonlinear and time-varying process problems are also very common in actual industrial processes. In nonlinear time-varying processes, the process models often change with varying running conditions. However, traditional SFA is a linear time-invariant model, which is limited in its ability to handle nonlinear and time-varying data relationships. To improve the prediction performance in nonlinear time-varying processes, local modeling techniques have been developed, such as just-in-time learning (JITL) [33–35] and locally weighted learning (LWL) [36–38], which ensure that the model can track the most relevant process state by constantly reconstructing local models for each query sample in an online manner. In JITL and LWL, it is assumed that all the historical data samples are stored in the database. Moreover, the predicted local model is built online upon querying dynamically. For online prediction, if a sample is obtained with its easy-to-measure input part and it is necessary to estimate the difficult-to-measure output part, this sample is called a query type. A local modeling technique can use linear models to approximate the complex nonlinear relationships by local linearization. However, traditional local modeling approaches are mostly based on static models, which do not consider the dynamics of process data. This is mainly because a local modeling strategy selects only those samples that have the smallest spatial distances from the query sample for online local modeling, and this may break the dynamic relationships of the data sequence.

To alleviate these problems, a new locally weighted model is proposed in this paper for nonlinear dynamic process modeling. It is based on locally weighted slow feature analysis (LWSFA). LWSFA adopts the idea of LWL that mainly relies on similarity-based weighting for local modeling. For traditional LWL, the main intention of the sample weighting technique is to handle process nonlinearity, which breaks the process data dynamics since only the most relevant samples are used for local modeling. To deal with process nonlinearity and dynamics simultaneously, two kinds of similarity metrics are adopted in LWSFA: static sample similarity and dynamic first-order time difference similarity. Correspondingly, two different weighting techniques are used to deal with the static and dynamic nonlinearities, respectively. First, the traditional sample similarity is calculated for each historical sample, to deal with the static nonlinear relationship between LVs and the original input variables. The first-order time difference similarity is then used to measure and calculate the dynamic slowness relevance, which can preserve the nonlinear dynamic information for modeling. Based on the two similarities, each historical sample and its first-order time difference are assigned a static sample weight and a dynamic weight, according to their respective similarities with that of the query sample. After that, nonlinear dynamic SFs from the original inputs are implemented and obtained by online LWSFA. Finally, the extracted SFs are used to build locally weighted slow feature regression (LWSFR) for output prediction. In this step, the nonlinear features and the local model structure can greatly enhance the nonlinear dynamic adaptability of the proposed LWSFR. The trained local model is discarded once the prediction is finished for the current query sample. The modeling steps are repeated when the next query sample arrives. Therefore, the proposed LWSFA-based local modeling framework can not only enhance the nonlinear dynamic adaptation, but also maintain the high performance of the soft sensor model.

This paper mainly deals with the topic of virtual measurement for industrial processes, which is also known as soft sensor and inferential sensor. The main contributions of this paper lie in the following aspects. A locally weighted slow feature regression (LWSFR) is proposed for the virtual measurement of the difficult-to-measure quality variables in nonlinear dynamic industrial processes. In LWSFR, two kinds of weights are designed in order to cope with the nonlinearities and dynamics based on SFA. The sample weight is first calculated and used for local linearization for static nonlinearity. The temporal weight is then designed to locally linearize the dynamic nonlinearity of the process. In this way, the proposed LWSFR can handle complex nonlinear dynamic data relationships with a simple adaptive modeling strategy.

The remaining parts of this article are structured as follows. In section 2, SFA is briefly introduced. Section 3 provides details of the proposed LWSFA and LWSFR models. In section 4, case studies are carried out on an industrial hydrocracking process to demonstrate the effectiveness of the

proposed method. A summary of the main contributions of this paper is given in section 5.

2. Slow feature analysis (SFA)

The SFA algorithm was first proposed in 2002 and was originally applied in the computational neuroscience field [28], the intention being to assist in the organization of the visual system in the human brain. Thereafter, SFA was successfully applied in other fields, such as change detection and object recognition. SFA is a linear factor model that aims to learn invariant features from time signals. Its basic idea is that the important features are mostly unchanging in a scene. In this section, the basic principle and implementation steps of traditional SFA are reviewed. First, the symbols and definitions used in the section are explained. The mathematical formulation of SFA is then given in detail.

2.1. Definition of slowness

Assume that there is a continuous stochastic ergodic signal $X(t)$, which can change slowly or quickly. The change is measured quantitatively as

$$\Delta(X(t)) \triangleq \left\langle \dot{X}^2(t) \right\rangle_{\mathrm{t}} \tag{1}$$

where the notation $\Delta(\bullet)$ can be seen as a definition of slowness, and $\dot{X}^2(t)$ is the square of $\dot{X}(t)$, with $\dot{X}(t)$ being the derivative of $X(t)$. The operator $\langle\bullet\rangle_{\mathrm{t}}$ represents the expectation term obtained by time-averaging over $\bullet$. Usually, the expectation of $X(t)$ during the time range $[t_0, t_1]$ can be calculated as

$$E\{X(t)\} \approx \langle X(t)\rangle_{\mathrm{t}} \triangleq \frac{1}{t_1 - t_0}\int_{t_0}^{t_1} X(t)\mathrm{d}t. \tag{2}$$

$\dot{X}(t)$ is the first-order derivative of $X(t)$ with respect to time as

$$\dot{X}(t) = \frac{\mathrm{d}X(t)}{\mathrm{d}t}. \tag{3}$$

In industrial process applications, process data $X(t)$ is discretely sampled because it is measured with a certain sampling interval. Suppose that there is a section of time series data $X(t) = \{X(1), X(2), \cdots, X(T)\}$, where T is the total number of the sampled data series. Hence, the discrete form of the expectation by time-averaging in equation (2) is calculated as

$$\langle X(t)\rangle_{\mathrm{t}} \triangleq \frac{1}{t_1 - t_0}\int_{t_0}^{t_1} X(t)\mathrm{d}t \approx \frac{1}{T}\sum_{t=1}^{T} X(t). \tag{4}$$

Similarly, the discrete form of the derivation in equation (3) can be rewritten as

$$\dot{X}(t) = \frac{\mathrm{d}X(t)}{\mathrm{d}t} \approx X(t) - X(t-1). \tag{5}$$

2.2. Slow feature analysis (SFA)

Suppose that there is a time series $\{x(1), x(2), ..., x(t), \ldots\}$ with each input vector as $x(t) = \{x_1(t), x_2(t), \cdots x_m(t)\}$, where m represents the dimension of the input variables. A real-valued mapping function $g(\bullet)$ is first used to map the m-dimensional input vector to q-dimensional latent variable space, as $g(x(t)) = \{g_1(x(t)), g_2(x(t)), \cdots, g_q(x(t))\}$. In the feature space, the primary LVs (called SFs) are represented as the outputs of the given real-valued mapping functions $s_j(t) \triangleq g_j(x(t)), (1 \leqslant j \leqslant q)$. The purpose of SFA is to make SFs change as slowly as possible, which means to make the SFs as slow as possible. In other words, it seeks out a combination of real-valued functions $g_j(\bullet), 1 \leqslant j \leqslant q$ in order to minimize

$$\min_{g_j(\cdot)} \Delta(s_j) \triangleq \left\langle \dot{s}_j^2(t) \right\rangle_t \tag{6}$$

with the constraints as

$$\langle s_j(t)\rangle_t = 0 \quad \text{(zero mean)} \tag{7}$$

$$\left\langle s_j^2(t)\right\rangle_{\mathrm{t}} = 1, \quad \text{(unit variance)} \tag{8}$$

$$\forall i \neq j, \langle s_i(t)s_j(t)\rangle_t = 0, \quad \text{(decorrelation and order)}. \tag{9}$$

It can be seen that any constant function $g_j(x(t))$ will have an optimal solution if the optimization objective in equation (6) is considered without any constraints. Hence, the role of the constraint in equations (7) and (8) is to avoid meaningless solutions by scaling SFs to zero mean and unit variance. The constraint in equation (9) ensures that the SFs are independent of each other and that the SFs are naturally arranged in ascending order, according to their slowness. That is to say, s_1 is the SF with the minimum slowness while s_q is the SF with the maximum slowness.

To satisfy the constraints of zero mean and unit variance, the input data $x(t)$ should be standardized as

$$x_j := \frac{x_j - \bar{x}_j}{\sigma_j}, 1 \leqslant j \leqslant m \tag{10}$$

where x_j, $\bar{x}_j$, and σ_j are the input, the mean, and the standard deviation of the jth input variable, respectively. For consistency, $x(t)$ will represent the standardized input data in the following context. SFA hypothesizes that each SF is derived from a linear mapping $g(\bullet)$ of input data $x(t)$, which is

$$s(t) = W^T x(t) \tag{11}$$

where $W \in \mathbb{R}^{m\times q}$ is the projection matrix. Thus, the optimization objective in equation (6) and the constraint in equation (8) can be rewritten as the following forms:

$$\Delta(s_j) = \left\langle \dot{s}_j^2(t)\right\rangle_{\mathrm{t}} = \left\langle \left(w_j^T \dot{x}(t)\right)^2 \right\rangle_{\mathrm{t}} = w_j^T \left\langle \dot{x}(t)\dot{x}^T(t)\right\rangle_{\mathrm{t}} w_j \tag{12}$$

$$\left\langle s_j^2(t)\right\rangle_{\mathrm{t}} = \left\langle \left(w_j^T x(t)\right)^2 \right\rangle_{\mathrm{t}} = w_j^T \left\langle x(t)x^T(t)\right\rangle_{\mathrm{t}} w_j = 1. \tag{13}$$

With equations (12) and (13), equation (14) can be obtained as

$$\frac{\Delta(s_j)}{1} = \frac{\left\langle \dot{s}_j^2(t)\right\rangle_{\mathrm{t}}}{\left\langle s_j^2(t)\right\rangle_{\mathrm{t}}} = \frac{w_j^T \left\langle \dot{x}(t)\dot{x}^T(t)\right\rangle_{\mathrm{t}} w_j}{w_j^T \left\langle x(t)x^T(t)\right\rangle_{\mathrm{t}} w_j} = \frac{w_j^T A w_j}{w_j^T B w_j} \tag{14}$$

where $A = \left\langle \dot{x}(t)\dot{x}^T(t)\right\rangle_{\mathrm{t}}$ with $A_{ij} = \left\langle \dot{x}_i(t)\dot{x}_j^T(t)\right\rangle_{\mathrm{t}}$ and $B = \langle x(t)x^T(t)\rangle_{\mathrm{t}}$ with $B_{ij} = \left\langle x_i(t)x_j^T(t)\right\rangle_{\mathrm{t}}$ are the covariance matrices of

$\dot{x}(t)$ and$x(t)$, respectively. As a result, the projection matrix W can be optimized by the following generalized eigenvalue problem:

$$AW = BW\Omega \tag{15}$$

where Ω is a diagonal matrix composed of generalized eigenvalues $\lambda_1, \lambda_2, ..., \lambda_m$ with $\lambda_1 < \lambda_2 < ... < \lambda_m$, and $\lambda_i, i = 1, ..., m$ is the generalized eigenvalues of pair (A, B). Correspondingly, the optimal values of the objective function in equation (6) can be sequentially obtained as

$$\Delta(s_j) = \left\langle \dot{s}_j^2(t) \right\rangle_t = \lambda_j. \tag{16}$$

Finally, the SFs of the input data $x(t)$ can be extracted by equation (11).

3. Locally weighted slow feature regression (LWSFR)

In this section, the derivation of LWSFA and the calculations for two kinds of weights are introduced in detail. Furthermore, a summary of the modeling procedures is provided for the LWSFR-based prediction framework.

3.1. Locally weighted slow feature analysis

Although SFA can extract dynamic features from process data series, it is limited when modeling for nonlinear time-varying processes and may not meet the accuracy requirements. Thus, a LWSFA method is proposed in this section that is enlightened by the idea of LWL. As mentioned above, the input matrix of the time series is $X = [x_1, x_2, ..., x_m]^T, X \in \mathbb{R}^{m\times n}$, where m and n are the dimensions of variables and the number of samples, respectively. For the formulas in SFA, there are two kinds of basic terms. One is the sample term $s(t)$ that is related to matrix B, which represents the static information. The other is the temporal derivative $\dot{s}(t)$ that is related to matrix A, which reflects the dynamics of process data. In order to deal with the nonlinear static and dynamic relationships, two weighting techniques are designed for local linearization. One is the sample weight, which is similar to the traditional weight type used in LWL for static nonlinear relationships. The second type is the weight for the first-order time difference term that is used to describe the nonlinear dynamics.

The first type of sample weight is determined by the spatial similarity of each historical sample to the query sample, which is often used to handle nonlinear relationships in LWL. Assume μ_i is the weight for the ith historical sample. The weight matrix of the input samples is then denoted as $\mu_{\mathrm{B}} = \mathrm{diag}\left([\mu_1, \mu_2, \cdots \mu_n]\right)$. The illustration and calculation of the weight matrix μ_{B} will be given in section 3.2. To handle the static nonlinear relationship, sample weights are used to construct a weighted constraint for LWSFA, which can be written as

$$\left\langle s_j^2 \right\rangle_{t,\mu_{\mathrm{B}}} = \left\langle \left(w_j^T (X\mu_{\mathrm{B}})\right)^2 \right\rangle_{\mathrm{t}} = w_j^T \left\langle X\mu_{\mathrm{B}}\mu_{\mathrm{B}}^T X^T \right\rangle_{\mathrm{t}} w_j = 1. \tag{17}$$

If a sample is assigned with a large weight, which represents it is more related to the query one, it can occupy more importance than the other samples in the constraint of LWSFA.

To locally linearize the dynamic nonlinearities, a weighting technique should be adopted to measure the similarity of nonlinear dynamics. Since the data dynamic is represented by the first-order time difference of the SFs, the dynamic weighting technique is needed to measure the first-order time difference terms between the query sample and the historical samples. Assume φ_i is the dynamic weight of the first-order time difference term between the ith historical sample and its previous sampling data. The dynamic weight matrix can then be denoted as $\varphi_{\mathrm{A}} = \mathrm{diag}\left([\varphi_2, \varphi_3, ...\varphi_n]\right)$ for the historical dataset. The calculation method for dynamic weight will be given in detail in section 3.2.

The dynamic weight is then used to construct a new weighted objective function in LWSFA. With the dynamic weight matrix, the optimization objective in equation (6) of SFA is rewritten for LWSFA as follows:

$$\Delta(s_j)_{\varphi_{\mathrm{A}}} = \left\langle \dot{s}_j^2 \right\rangle_{t,\varphi_{\mathrm{A}}} = \left\langle \left(w_j^T (\dot{X}\varphi_{\mathrm{A}})\right)^2 \right\rangle_{\mathrm{t}} = w_j^T \left\langle \dot{X}\varphi_{\mathrm{A}}\varphi_{\mathrm{A}}^T \dot{X}^T \right\rangle_{\mathrm{t}} w_j. \tag{18}$$

To derive the solutions for the optimization problem with the new objective in equation (18) and the new constraint in equation (17), the following formula can be obtained as

$$\frac{\left\langle \dot{s}_j^2 \right\rangle_{t,\varphi_{\mathrm{A}}}}{\left\langle s_j^2 \right\rangle_{t,\mu_{\mathrm{B}}}} = \frac{w_j^T \left\langle \dot{X}\varphi_{\mathrm{A}}\varphi_{\mathrm{A}}^T \dot{X}^T \right\rangle_{\mathrm{t}} w_j}{w_j^T \left\langle X\mu_{\mathrm{B}}\mu_{\mathrm{B}}^T X^T \right\rangle_{\mathrm{t}} w_j} = \frac{w_j^T A^{\varphi} w_j}{w_j^T B^{\mu} w_j} \tag{19}$$

where $A^{\varphi} = \left\langle \dot{X}\varphi_{\mathrm{A}}\varphi_{\mathrm{A}}^T \dot{X}^T \right\rangle_{\mathrm{t}}$ with $A_{ij}^{\varphi} = \left\langle \dot{x}_i\varphi_{\mathrm{A}}\varphi_{\mathrm{A}}^T \dot{x}_j^T \right\rangle_{\mathrm{t}}$ and $B^{\mu} = \left\langle X\mu_{\mathrm{B}}\mu_{\mathrm{B}}^T X^T \right\rangle_{\mathrm{t}}$ with $B_{ij}^{\mu} = \left\langle x_i\mu_{\mathrm{B}}\mu_{\mathrm{B}}^T x_j^T \right\rangle_{\mathrm{t}}$ are the weighted covariance matrices of $\dot{x}(t)$ and $x(t)$, respectively.

To determine the projection matrix W, a generalized eigenvalue problem is constructed:

$$A^{\varphi} W = B^{\mu} W\Omega \tag{20}$$

where $\Omega = \mathrm{diag}\left(\{\lambda_1, \lambda_2, ...\lambda_m\}\right).\{\lambda_j\}$is the set of generalized eigenvalues, where λ_1 is the smallest and λ_{m} is the largest. The projection matrix W can be obtained because it happens to be the matrix consisting of the generalized eigenvectors, and the nonlinear SFs of the original data can also be determined by equation (20).

3.2. Weight calculation

In this part, the two weight types should be determined and calculated. The first task is to determine the sample weight. There are many criteria for sample weight assignment, in which the Euclidean distance is one of the most widely used. Two steps are carried out to calculate the sample weights. In the first step, the Euclidean distance criterion is used to calculate the distance d_n between each historical sample x_n and the query sample x_q. A function based on d_n is then constructed for weight calculation. In the constructed function, a sample with a large d_n is often given a smaller weight, and vice versa. In LWSFA, the Euclidean distance criterion is employed to

calculate the weight matrix μ_{B}. Therefore, the distance of the ith sample is calculated as

$$d_i^B = \sqrt{(x_i - x_q)^T (x_i - x_q)}, \quad i = 1, 2, ..., n. \tag{21}$$

A weight should then be given to each historical sample according to the obtained distance. If a historical sample is a small distance from the query sample, it means that this sample is more related to the query sample and a larger weight should be given. Conversely, a historical sample that is a large distance from the query sample is considered to be less similar to the query sample and will be assigned a smaller weight. Thus, the weight can be calculated as

$$\mu_i = \exp\left(-\frac{\left(d_i^B\right)^2}{\delta_B^2}\right), \quad i = 1, 2, ..., n \tag{22}$$

where δ_{B}^2 is an adjustment parameter, which controls the decay rate of the weight with regard to d_i^B. If δ_{B}^2 is large, the weight deceases slowly, or rapidly if δ_{B}^2 is small. After the sample weight is calculated for each historical sample, the weighted matrix $\mu_{\mathrm{B}} = \mathrm{diag}\left([\mu_1, \mu_2, \cdots, \mu_n]\right)$ can be obtained. As can be seen, samples that are more related to the query sample will be assigned large weights, and will be of greater importance within the constraint of LWSFA. With this weighted constraint, the static nonlinear relationship can be well handled.

The second type of weight matrix $\varphi_{\mathrm{A}} = \mathrm{diag}\left([\varphi_2, \varphi_3, \cdots, \varphi_n]\right)$ is then calculated to deal with the nonlinear dynamic data relationships, and is used to determine the importance of the first-order time difference terms for the historical sample sequence. To this end, the first-order time difference for the historical samples and the query sample are calculated as

$$\begin{aligned} \Delta x_i &= x_i - x_{i-1}, \quad i = 2, \cdots, n \\ \Delta x_q &= x_q - x_{q-1} \end{aligned} \tag{23}$$

where the operator Δ represents the calculation of the first-order time difference term. The distances of the first-order time difference terms for the historical samples are then obtained as

$$d_i^A = \sqrt{(\Delta x_i - \Delta x_q)^T (\Delta x_i - \Delta x_q)}, \quad i = 2, \cdots, n. \tag{24}$$

Correspondingly, the weight of the first-order time difference is calculated based on distance d_i^A as

$$\varphi_i = \exp\left(-\frac{\left(d_i^A\right)^2}{\delta_A^2}\right), \quad i = 2, \cdots, n \tag{25}$$

where δ_{A}^2 is an adjustment parameter that is similar with δ_{B}^2 but independent from it. With the weight matrices φ_{A} and μ_{B}, the nonlinear dynamic features can be extracted progressively. After the extraction of nonlinear dynamic features, a local LWSFR soft sensor model will be built for each query sample. The details of the LWSFR modeling method will be introduced in next part.

3.3. Locally weighted slow feature regression (LWSFR)

Local modeling technology adopts the idea of local linearization which continually rebuilds a simple model to replace a complex global nonlinear model, to reduce model complexity. LWSFA retains the necessary information of the input data in the SF feature space. The adaptive SFs extracted by LWSFA algorithm can describe the complex nonlinear dynamic characteristics of industrial processes. According to the basic principles of LWSFA, the slowest SF describes the essential change of process while the fastest SF mainly describes changes of noise. Hence, the SFs with small slowness are used to build soft sensor models to predict the quality variables. Let $X_{\mathrm{train}} \in \mathbb{R}^{m \times n}$ and $Y_{\mathrm{train}}^T \in \mathbb{R}^n$ be the standardized input and output training data sets, where m and n are the dimensions of the input variables and the quantity of training data, respectively. Let the input of the query sample be $x_q \in \mathbb{R}^m$. They can used to obtain the projection matrix W with the described LWSFA.

To improve the performance of the model, the first M columns of mapping matrix W are used to map the input data, so that the feature space contains only the required SFs. The extracted SFs for the training data and the query data are

$$S_{\mathrm{train}} = W_{1:M}^T X_{\mathrm{train}} \tag{26}$$

$$s_q = W_{1:M}^T x_q. \tag{27}$$

The quality variable of the query sample can then be predicted as

$$\hat{y}_q = \sum_{j=1}^{M} \theta_j^T s_{q,j} = \sum_{j=1}^{M} \theta_j^T w_j^T x_q = \theta^T W_{1:M}^T x_q = \theta^T s_q \tag{28}$$

where $\hat{y}_q$ is the predicted output of the query data. $\theta \in \mathbb{R}^M$ is the coefficient vector of the regression model, which is computed by the least squares method as $\theta = \left(S_{\mathrm{train}}^{\mathrm{T}} S_{\mathrm{train}}\right)^{-1} S_{\mathrm{train}}^{\mathrm{T}} Y_{\mathrm{train}}$.

The regression model is then given up and a new LWSFR is built when the next query data arrives. Figure 1 shows a flowchart of the LWSFR modeling framework. The detailed steps in building a LWSFR model are summarized as follows.

Step 1: Collect the historical training input and output data from the industrial process.

Step 2: Whenever a query sample arrives, calculate the distances between each historical training sample and the query sample in the original space and in the first-order time difference space. Based on the calculated distance, assign a static weight to each training sample and a dynamic weight to each first-order time difference term.

Step 3: Construct the weighted matrixes B and A of the LWSFA algorithm, using the weighted training samples and the weighted differential training samples, respectively. A generalized eigenvalue problem is

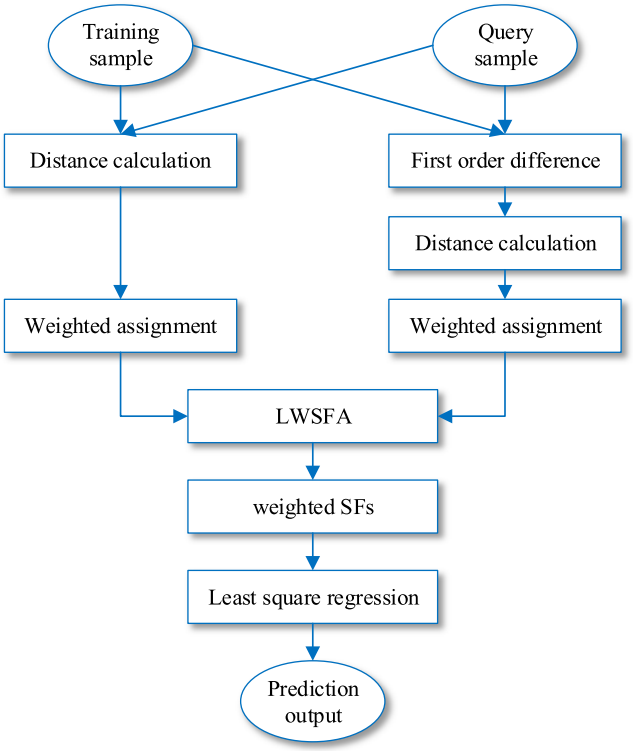

Figure 1. Flow chart of LWSFR modeling procedure.

established with weighted matrixes B and A. The projection directions of the SF space are then obtained by solving this generalized eigenvalue problem.

Step 4: Extract the first M SFs of the training samples and the query sample by mapping them to weighted SF space. A least squares regression model is then used to predict the quality output of the query sample between the extracted nonlinear SFs and the quality variable.

Step 5: Discard the built model after finishing the prediction of the query sample. Repeat step 2 to step 4 to build a new LWSFR model when the next query sample arrives.

4. Case studies

In this section, an industrial hydrocracking process is used to demonstrate the validity of the LWSFR model. First, a brief description is given of the hydrocracking process. The LWSFR model is then built to predict its product quality. To illustrate the effectiveness of LWSFR, another two SFR-based soft sensor models are established for the purposes of performance comparison: a global SFR and a local linear SFR (LSFR).

There are three indexes used to evaluate the performance of the proposed model, which are the root mean square error (RMSE), the coefficient of determination (R^2), and tracking precision (TP). The three indicators are defined as follows:

$$\begin{cases} \mathrm{RMSE} = \sqrt{\frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{test}} (\hat{y}_i - y_i)^2} \\ R^2 = 1 - \frac{\sum_{i=1}^{N_{\mathrm{test}}} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{N_{\mathrm{test}}} (y_i - \bar{y})^2} \\ TP = 1 - \frac{\sigma^2_{\mathrm{error}}}{\sigma^2_{\mathrm{property}}} \end{cases} \tag{29}$$

where N_{test} is the number of testing samples; $\hat{y}_i$ is the predicted output value of the ith sample; y_i is the labeled output value of the ith sample; $\bar{y}$ is the mean value of the labeled output over the testing samples; $\sigma^2_{\mathrm{error}}$ is the variance of the deviation between the prediction and the label value; and $\sigma^2_{\mathrm{property}}$ is the variance of the labeled value. The RMSE index reveals the degree of deviation between the predicted and the labeled value. R^2 shows how much of the output variance can be interpreted by the model. TP reflects the tracking effect of the model predictions on the real outputs.

4.1. The hydrocracking process

Hydrocracking is an important part of a petroleum refining plant. It has a wide range of applications in the petrochemical industry, since it transforms heavy residues and heavy oils into light oils in an efficient and environmentally friendly way. In the hydrocracking process, there are two major reactions with hydrogenation and cracking. The basic flowchart of the hydrocracking process is exhibited in figure 2. There are four principal parts in the process, which are the material feeding part, reaction part, high and low pressure separation part, and fractionation part. First, various feedstock oils are blended into the feed part. The recycled and make-up hydrogen are mixed and heated in the heating furnace. The mixed oil materials and the heated hydrogen are then fed into the reaction part for hydrogenation and cracking. Eventually, the mixture products from the reaction part are separated into a variety of desired petrochemical products through the high and low pressure separation part and fractionation part.

There are a variety of such products in the hydrocracking process, including diesel, aviation kerosene, light naphtha and heavy naphtha, each of which has many quality properties. Many of them must be controlled and optimized in real time. Moreover, they should be monitored to provide real-time adjustment information for process optimization. However, it is difficult to measure the quality attributes of the products by online analyzers. Instead, they are often analyzed by carrying out off-line laboratory tests. This often results in time delays and low sampling frequency, which makes it difficult to implement online adjustment and optimization schemes. Therefore, it is desirable to develop soft sensors to predict those difficult-to-measure variables via these easily measured variables. In this study, the 10% recovery temperature of the diesel and the final boiling point of the aviation kerosene are chosen as the quality variables for soft sensor modeling.

There are a large number of easy-to-measure process variables in the hydrocracking process. Among them, 148 variables

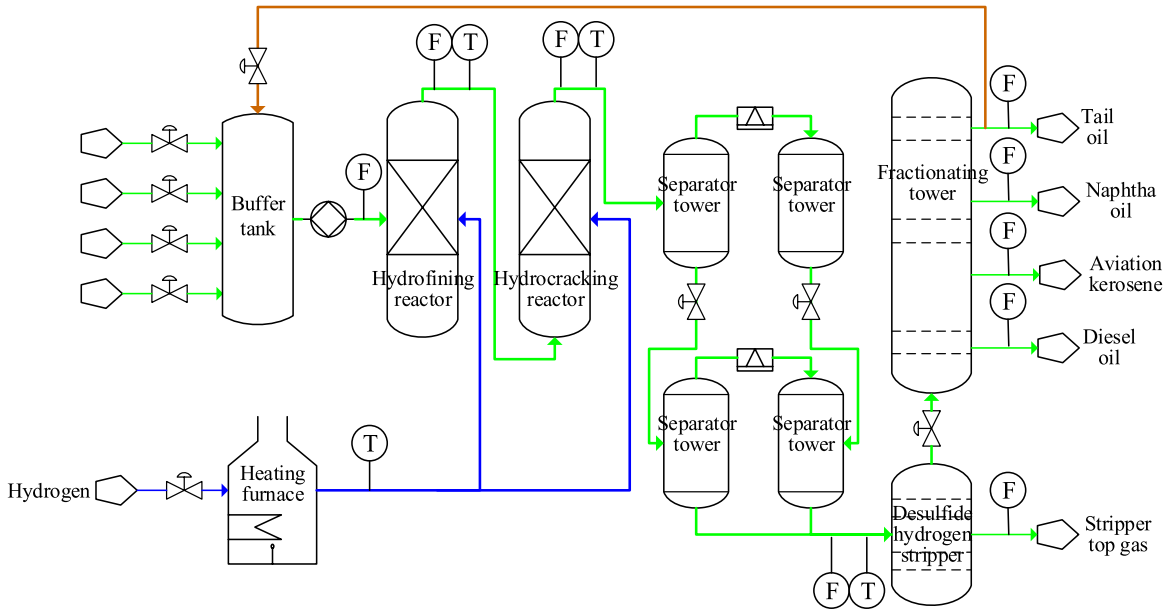

Figure 2. Flow chart of the hydrocracking process.

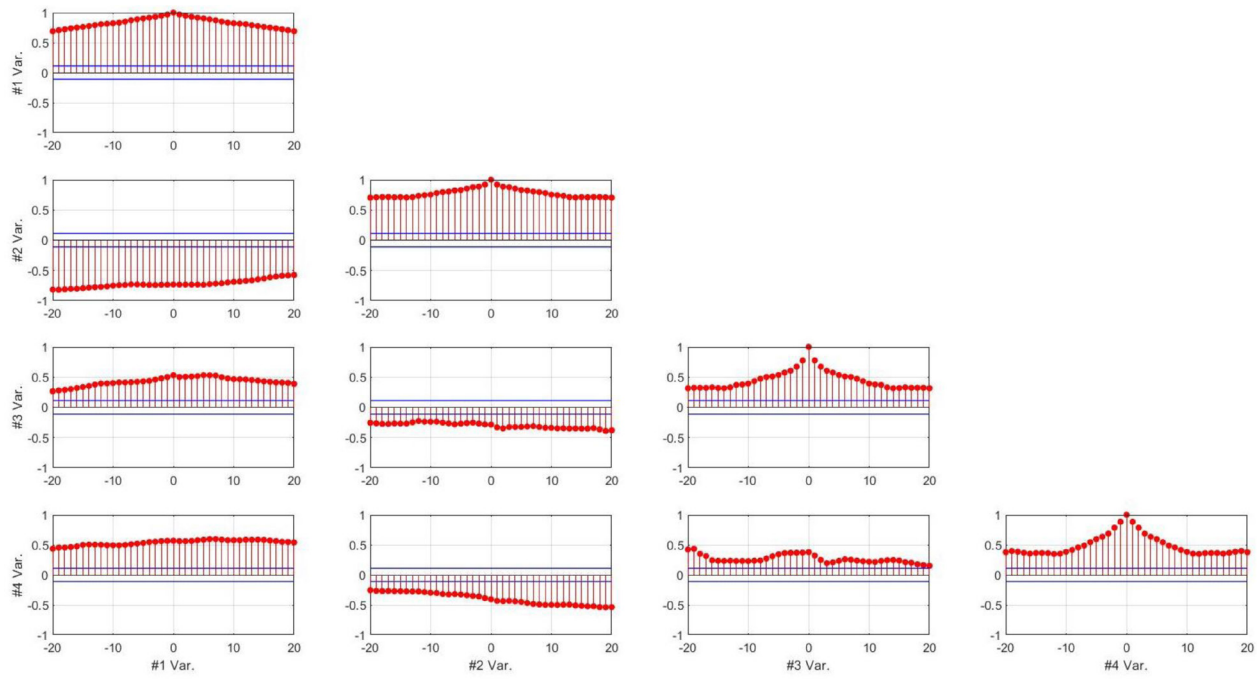

Figure 3. The auto- and cross-correlation analysis for four input variables.

related to the quality of products are preliminarily selected by prior knowledge. Due to the page limit, the 148 variables are not listed here. The data were collected from a real industrial hydrocracking process at a refinery enterprise in China. In total there are 324 samples, taken from April 11, 2017 to February 28, 2018. The sampling frequency of the quality variables is one sample per day. The hydrocracking process is characterized by continuous and uninterrupted production. It often takes a long time—several hours—from the feeding of raw materials to the obtaining of the product. Moreover, the dynamic adjustment and control are frequently implemented step by step in this process. Hence, there are highly dynamic relationships between the sampled data. Also, there are some problems that necessitate the dynamic modeling, such as fluctuations in raw materials. Although the sampling frequency is one sample per day, there is still a dynamic relationship in these data samples. To verify this, four input variables are taken to carry out, for example, the auto-correlation and cross-correlation tests. The results are shown in figure 3. As can be seen, there are strong auto-correlations for each of the four input variables. Moreover, many of the absolute values of the cross-correlation coefficients are larger than 0.5

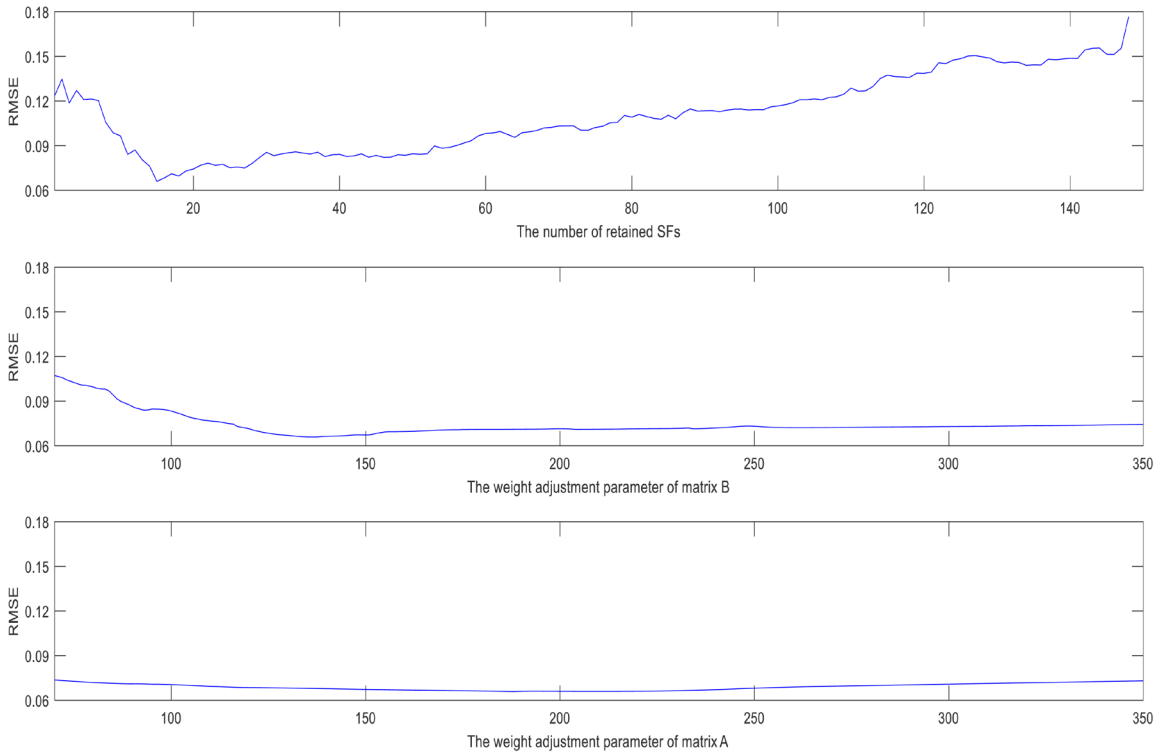

Figure 4. Sensibility analysis of the number of retained SFs, δ_B^2, and δ_A^2 in LWSFR model for the 10% recovery temperature of diesel.

Table 1. Performance comparison for the 10% recovery temperature of diesel with the three methods on the testing dataset.

Method	SFR	LSFR	LWSFR
RMSE	0.1615	0.0832	0.0660
R^2	−1.1909	0.4185	0.6339
TP	−0.6901	0.4244	0.6352

for different pairs of input variables. Hence, it is necessary to build dynamic models for soft sensor application in this hydrocracking process.

4.2. The 10% recovery temperature of diesel

Diesel is one of the main products of the hydrocracking process. The 10% recovery temperature, one of the quality indicators for diesel, reflects the light diesel fraction, which has a significant effect on the starting performance of diesel engines. In order to monitor this index in real time, it is essential to establish a soft sensor model for this quality variable.

To build the SFR, LSFR and LWSFR soft sensor models, the first 216 samples from April 11, 2017 to November 12, 2017, are used as training data. The rest of the samples are used to test the model performance. SFR trains the model in the offline phase and implements the prediction in the online phase. LSFR and LWSFR are methods for online modeling and prediction. When a query sample appears, the LSFR and LWSFR models are built immediately for quality prediction and are discarded after obtaining the predicted output for the query sample. Furthermore, the training dataset is updated frequently when new data samples are available. There are several parameters that must be set properly for the soft sensor

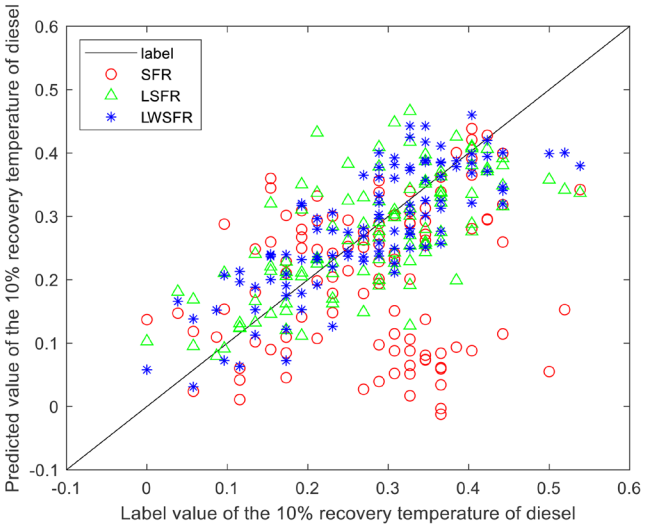

Figure 5. Prediction results of the SFR, LSFR, and LWSFR for the 10% recovery temperature of diesel.

Table 2. RMSE, R^2, and TP performance for the final boiling point of aviation kerosene with the three methods on the testing dataset.

Method	SFR	LSFR	LWSFR
RMSE	0.2497	0.1137	0.1009
R^2	−1.6238	0.4555	0.5714
TP	−1.3826	0.4695	0.5795

models. These parameters are determined by the trial and error technique [39–41]. For example, the number of SFs in the three models is set at 15. The weight adjustment parameters δ_A^2 and δ_B^2 of the LWSFR are determined to be 210 and 134,

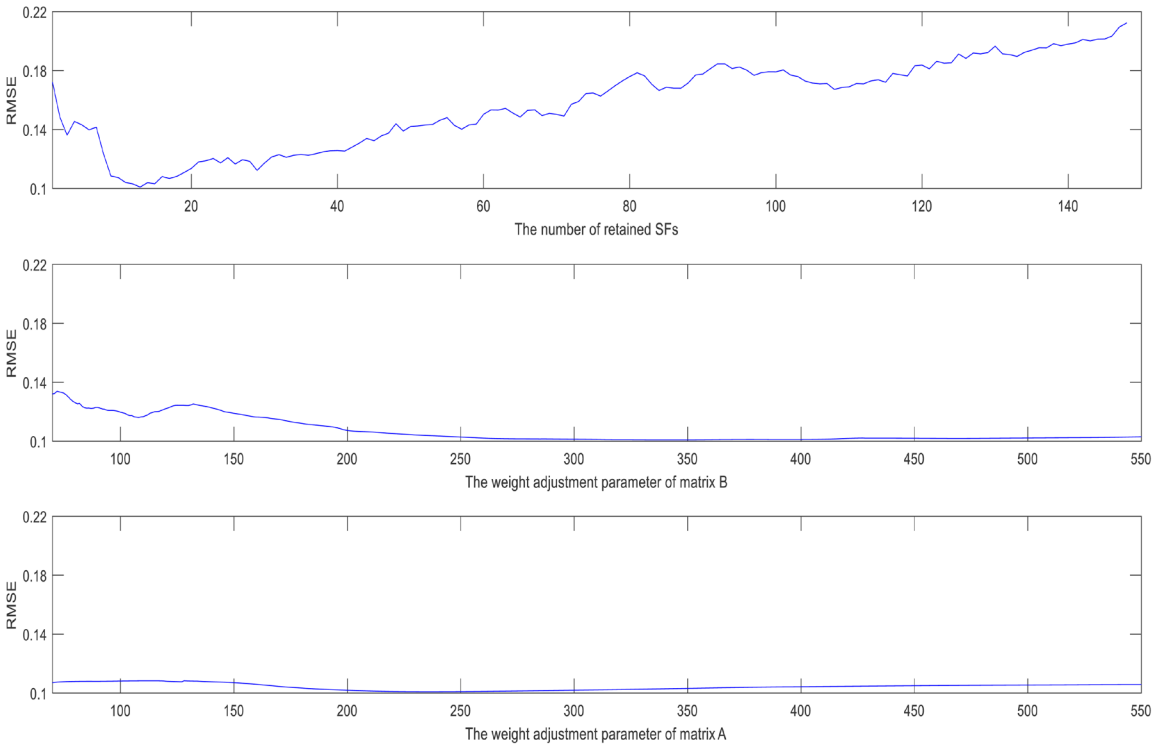

Figure 6. Sensibility analysis of the number of retained SFs, δ_B^2, and δ_A^2 for LWSFR model of the final boiling point of aviation kerosene.

respectively. The influence of these parameters on the RMSE accuracy is shown in figure 4 for LWSFR. As can be seen from figure 4, the number of retained SFs and δ_B^2 have more impact on the accuracy of the LWSFR than the other parameter, since the RMSE will change greatly with the number of retained SFs and δ_B^2. It can be seen that δ_A^2 has the least influence on the prediction performance, since the fluctuation of the RMSE of the LWSFR model is still small when the change of δ_B^2 is large. Therefore, the number of retained SFs and δ_B^2 should be paid more attention when tuning the parameters.

Table 1 displays the predicted results on the testing dataset for the three models, after the parameters were determined. In the second column in table 1, the R^2 index can be given negative values. This can be seen from its definition as shown in equation (29). As can be seen, it is possible to obtain a negative R^2 value since the numerator of $\sum_{i=1}^{N_{\text{test}}}(\hat{y}_i - y_i)^2$can be much larger than the denominator of $\sum_{i=1}^{N_{\text{test}}}(y_i - \bar{y})^2$. If the prediction accuracy of one model is very poor. Then, the predicted error is very large at most samples and thus the value of R^2 can be negative. It is similar for the TP index.

Table 1 clearly shows that SFR has the worst predictive performance, with the largest RMSE but the lowest R^2 and TP. SFR is a fixed global modeling approach. The latest process state cannot be tracked. On the other hand, SFR extracts only linear features so it cannot adapt well to this complex nonlinear process. By introducing a local modeling technique, LSFR offers a significant improvement over the performance of the SFR. With a newly rebuilt model for each testing sample, LSFR can track the latest state of the process in real-time. Furthermore, LWSFR has achieved the best prediction accuracy by adopting the locally weighted trick. Thus, the nonlinear features of the process are properly extracted and the updating capability is also preserved. Therefore, LWSFR is more effective than SFR and LSFR. In particular, the scatter plots are revealed in figure 5 for the predicted and labeled values of the testing samples. If the data points are more concentrated along the diagonal line in the scatter plot, it indicates that the prediction is more accurate for this model. It can be clearly seen that LWSFR tracks the label data the best, because the data points are more compactly distributed along the diagonal line than for the other two methods.

4.3. The final boiling point of aviation kerosene

Another important product in the hydrocracking process is aviation kerosene. The final boiling point, one of its quality indicators, reflects the heavy kerosene fraction, which affects both the combustion of aviation kerosene and engine wear. Therefore, soft sensor models are also built to monitor the final boiling point of aviation kerosene in the hydrocracking process.

There are again 324 data samples in total. The method of dividing the training and testing samples is consistent with the previous case. Similarly, the model parameters are decided by the trial and error technique. For the three models, the number of SFs is set at 13. The parameters δ_A^2 and δ_B^2 of LWSFR are determined as 346 and 235, respectively. Figure 6 shows the influence of the three parameters on the prediction RMSE accuracy of the model. It illustrates that the model accuracy is more sensitive to the number of retained SFs and δ_B^2.

Table 2 describes the prediction RMSE, R^2, and TP of the three models on the testing dataset. From table 2, LWSFR shows the highest accuracy of the three methods. For global linear SFR, once the process state has changed, the accuracy

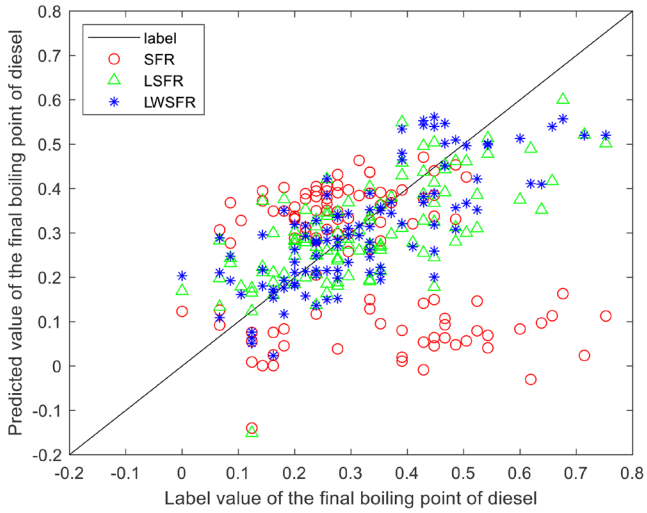

Figure 7. Prediction results of SFR, LSFR, and LWSFR for the final boiling point of aviation kerosene.

is rapidly reduced. Thereby, the SFR model has the lowest prediction performance. By incorporating the idea of local modeling, LSFR realizes an online update for each testing query sample. Therefore, the accuracy of LSFR is better than that of SFR. Considering the weighted technique for nonlinear dynamic feature learning, LWSFR can further improve the accuracy. The prediction results in table 2 show that both LSFR and LWSFR exhibit a significant improvement in performance over SFR. It is insufficient for a local modeling technique when the nonlinear problem is complicated. Hence, LWSFR can be better applied in time-varying nonlinear industrial processes. Figure 7 provides the prediction results of the three models on the testing set in the form of scatter plots. The results demonstrate the flexibility and effectiveness of the proposed LWSFR method.

5. Concluding remarks

In this paper, a soft sensor is developed for dynamic nonlinear processes from the perspective of latent variable modeling, which is based on LWSFR. The proposed method has several merits, as follows. First, SFs contain the temporal information of the data, which can deal with dynamic process characteristics. Second, LWL is used to weight the training samples and their first-order time differences, in order to handle the nonlinearity dynamics. Third, the form of local modeling ensures that the model can keep track of the latest process state, for adaptive modeling. Two cases have been implemented on an industrial hydrocracking process to test the performance of the proposed method. The comparative results of SFR, LSFR, and LWSFR show that LWSFR achieves the highest prediction accuracy.

Acknowledgment

This paper is supported in part by the National Natural Science Foundation of China (NSFC) (Grant Nos. 61590921, 61860206014, 61703440, 61621062), in part by the Innovation-driven plan in Central South University (Grant No. 2018CX011), in part by the Natural Science Foundation of Hunan Province (Grant No. 2018JJ3687), and in part by the Hunan Postgraduate Research Innovation Project (Grant No. 2019zzts151).

ORCID iDs

Xiaofeng Yuan https://orcid.org/0000-0002-9072-7179

References

[1] Yuan X, Huang B, Wang Y, Yang C and Gui W 2018 Deep learning based feature representation and its application for soft sensor modeling with variable-wise weighted SAE *IEEE Trans. Ind. Inf.* **14** 3235–43

[2] Fu Y, Yang W, Xu O, Zhou L and Wang J 2017 Soft sensor modelling by time difference, recursive partial least squares and adaptive model updating *Meas. Sci. Technol.* **28** 045101

[3] Yuan X, Gu Y, Wang Y, Yang C and Gui W 2019 A deep supervised learning framework for data-driven soft sensor modeling of industrial processes *IEEE Trans. Neural Netw. Learn. Syst.* (https://doi.org/10.1109/TNNLS.2019.2957366)

[4] Marchi M, Baptista F G, de Aguiar P R and Bianchi E C 2015 Grinding process monitoring based on electromechanical impedance measurements *Meas. Sci. Technol.* **26** 045601

[5] Ge Z, Song Z and Gao F 2013 Review of recent research on data-based process monitoring *Ind. Eng. Chem. Res.* **52** 3543–62

[6] Yuan X, Li L and Wang Y 2019 Nonlinear dynamic soft sensor modeling with supervised long short-term memory network *IEEE Trans. Ind. Inf.* (https://doi.org/10.1109/TII.2019.2902129)

[7] Forruna L, Graziani S, Rizzo A and Xibilia M G 2007 *Soft Sensors for Monitoring and Control of Industrial Processes* (Berlin: Springer) (https://doi.org/10.1007/978-1-84628-480-9)

[8] Liu Y, Zeng J, Xie L, Luo S and Su H 2019 Structured joint sparse principal component analysis for fault detection and isolation *IEEE Trans. Ind. Inf.* **15** 2721–31

[9] Wang Y, Pan Z, Yuan X, Yang C and Gui W 2019 A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network *ISA Trans.* (https://doi.org/10.1016/j.isatra.2019.1007.1001)

[10] Dai J, Chen N, Yuan X, Gui W and Luo L 2019 Temperature prediction for roller kiln based on hybrid first-principle model and data-driven MW-DLWKPCR model *ISA Trans.* (https://doi.org/10.1016/j.isatra.2019.1008.1023)

[11] Shao W, Yao L, Ge Z and Song Z 2019 Parallel computing and SGD-based DPMM for soft sensor development with large-scale semi-supervised data *IEEE Trans. Ind. Electron.* **66** 6362–73

[12] Yuan X, Zhou J, Huang B, Wang Y, Yang C and Gui W 2019 Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy *IEEE Trans. Ind. Inf.* (https://doi.org/10.1109/TII.2019.2938890)

[13] Kadlec P, Gabrys B and Strandt S 2009 Data-driven soft sensors in the process industry *Comput. Chem. Eng.* **33** 795–814

[14] Shao W, Ge Z and Song Z 2019 Quality variable prediction for chemical processes based on semi-supervised Dirichlet process mixture of Gaussians *Chem. Eng. Sci.* **193** 394–410

[15] Yuan X, Ge Z and Song Z 2014 Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes *Ind. Eng. Chem. Res.* **53** 13736–49
[16] Yuan X, Wang Y, Yang C, Ge Z, Song Z and Gui W 2018 Weighted linear dynamic system for feature representation and soft sensor application in nonlinear dynamic industrial processes *IEEE Trans. Ind. Electron.* **65** 1508–17
[17] Kaneko H and Funatsu K 2016 Ensemble locally weighted partial least squares as a just-in-time modeling method *AIChE J.* **62** 717–25
[18] Yuan X, Zhou J, Wang Y and Yang C 2018 Multi-similarity measurement driven ensemble just-in-time learning for soft sensing of industrial processes *J. Chemometr.* **32** e3040
[19] McArthur L and Greensill C 2007 Impact of resolution on NIR PLS calibration of kaolinite content with Weipa bauxite *Meas. Sci. Technol.* **18** 1343–7
[20] Jain P, Rahman I and Kulkarni B D 2007 Development of a soft sensor for a batch distillation column using support vector regression techniques *Chem. Eng. Res. Des.* **85** 283–7
[21] Wang Y, Wu D and Yuan X 2019 A two-layer ensemble learning framework for data-driven soft sensor of the diesel attributes in an industrial hydrocracking process *J. Chemometr.* **33** e3185
[22] Panda S S, Chakraborty D and Pal S K 2008 Flank wear prediction in drilling using back propagation neural network and radial basis function network *Appl. Soft Comput.* **8** 858–71
[23] Yuan X, Ou C, Wang Y, Yang C and Gui W 2019 Deep quality-related feature extraction for soft sensing modeling: a deep learning approach with hybrid VW-SAE *Neurocomputing* (https://doi.org/10.1016/j.neucom.2018.1011.1107)
[24] Yuan X, Ou C, Wang Y, Yang C and Gui W 2019 A layer-wise data augmentation strategy for deep learning networks and its soft sensor application in an industrial hydrocracking process *IEEE Trans. Neural Netw. Learn. Syst.* (https://doi.org/10.1109/TNNLS.2019.2951708)
[25] Yuan X, Li L, Wang Y, Yang C and Gui W 2019 Deep learning for quality prediction of nonlinear dynamic process with variable attention-based long short-term memory network *Can. J. Chem. Eng.* (https://doi.org/10.1002/cjce.23665)
[26] Zhou L, Zheng J, Ge Z, Song Z and Shan S 2018 Multimode process monitoring based on switching autoregressive dynamic latent variable model *IEEE Trans. Ind. Electron.* **65** 8184–94
[27] Zhou L, Chen J, Song Z, Ge Z and Miao A 2014 Probabilistic latent variable regression model for process-quality monitoring *Chem. Eng. Sci.* **116** 296–305
[28] Wiskott L and Sejnowski T J 2002 Slow feature analysis: unsupervised learning of invariances *Neural Comput.* **14** 715–70
[29] Wu C, Du B and Zhang L 2014 Slow feature analysis for change detection in multispectral imagery *IEEE Trans. Geosci. Remote Sens.* **52** 2258–74
[30] Zhang Z and Dacheng T 2012 Slow feature analysis for human action recognition *IEEE Trans. Pattern Anal. Mach. Intell.* **34** 436–50
[31] Shang C, Yang F, Gao X, Huang X, Suykens J A K and Huang D 2015 Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis *AIChE J.* **61** 3666–82
[32] Shang C, Huang B, Yang F and Huang D 2016 Slow feature analysis for monitoring and diagnosis of control performance *J. Process Control* **39** 21–34
[33] Ge Z and Son Z 2010 A comparative study of just-in-time learning-based methods for online soft sensor modeling *Chemometr. Intell. Lab. Syst.* **104** 306–17
[34] Yuan X, Ge Z, Huang B, Song Z and Wang Y 2017 Semi-supervised JITL framework for nonlinear industrial soft sensing based on locally semi-supervised weighted PCR *IEEE Trans. Ind. Inf.* **13** 532–41
[35] Xie L, Zeng J and Gao C 2014 Novel just-in-time learning-based soft sensor utilizing non-Gaussian information *IEEE Trans. Control Syst.* **22** 360–8
[36] Kadlec P and Gabrys B 2011 Local learning-based adaptive soft sensor for catalyst activation prediction *AIChE J.* **57** 1288–301
[37] Chen N, Dai J, Yuan X, Gui W, Ren W and Koivo H 2018 Temperature prediction model for roller kiln by ALD-based double locally weighted kernel principal component regression *IEEE Trans. Instrum. Meas.* **67** 2001–10
[38] Yuan X, Ge Z, Huang B and Song Z 2017 A probabilistic just-in-time learning framework for soft sensor development with missing data *IEEE Trans. Control Syst.* **25** 1124–32
[39] Fortuna L, Graziani S and Xibilia M G 2005 Soft sensors for product quality monitoring in debutanizer distillation columns *Control Eng. Pract.* **13** 499–508
[40] Fujiwara K, Kano M, Hasebe S and Takinami A 2006 Soft-sensor development using correlation-based just-in-time modeling *AIChE J.* **55** 1754–65
[41] Facco P, Doplicher F, Bezzo F and Barolo M 2009 Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process *J. Process. Control* **19** 520–9