Soft-Sensor Development for Processes With Multiple Operating Modes Based on Semisupervised Gaussian Mixture Regression

Weiming Shao, Zhiqiang Ge, *Senior Member, IEEE*, and Zhihuan Song

***Abstract*— Gaussian mixture regression (GMR) is an effective tool in developing soft sensors for online estimating difficult-to-measure variables in industrial processes with multiple operating modes. However, the GMR usually requires a sufficient amount of labeled samples to guarantee accurate probability density function (PDF) estimations because of its supervised learning process. Unfortunately, in soft-sensor applications, labeled samples could be very infrequent due to technical or economic limitations, which may lead the GMR-based soft sensors to unreliable parameter estimation and model selection, resulting in poor prediction performance. To tackle this problem, a semisupervised GMR (S^2GMR) was proposed, where both labeled and unlabeled samples were effective. In the S^2GMR, the PDFs of Gaussian components in input space and the functional dependence between input and output variables were learned simultaneously based on the expectation–maximization algorithm. Moreover, the Bayesian information criterion was employed to automatically determine the number of Gaussians for the S^2GMR. The S^2GMR was first investigated by a numerical example, and then applied to a real-life ammonia synthesis process for estimating the oxygen concentration at the top of the primary reformer. The two case studies verified the effectiveness of the proposed method.**

***Index Terms*— Bayesian information criterion (BIC), expectation–maximization (EM), Gaussian mixture regression (GMR), multimode process, semisupervised learning, soft sensor.**

I. Introduction

IN INDUSTRIAL processes, there are many important quality-related but difficult-to-measure variables such as the melt index of polypropylene, octane number of gasoline, oxygen concentration in the furnace, biomass concentration in the fermentation process, catalyst activity in a chemical reaction, and so on. These variables are referred to as the "primary variables" and often measured through laboratory analyses or online analyzers, which suffer from several problems such as large measurement delay, high price, accuracy deterioration, and maintenance issues [1]. An infrequent sampling of these primary variables may result in bad closed-loop control performance, huge production waste or safety risks [2]. What is worse, closed-loop control based on direct sensing might be disabled in some extreme scenarios [3]. Soft sensors or inferential sensors, which are in essence computer programs, can provide estimations of primary variables using those easy-to-measure secondary variables. Owing to the desirable properties such as being delay free and low cost, soft sensors are popular alternatives of the laboratory analysis or online analyzer. With the aid of distributed control systems, a huge amount of process data that reflect the true process conditions become available, and a variety of data-driven soft sensors have been developed and applied in practical industrial processes [4]. Popular data-driven soft-sensor modeling methods include the principal component analysis/principal component regression (PCR), partial least squares (PLS), support vector regression, extreme learning machine (ELM), Gaussian process regression (GPR), neuro-fuzzy systems, and so on. In addition to their predictive applications, soft sensors can also be used for other industrial purposes such as process monitoring [5]. One can refer to [6] and [7] for comprehensive reviews about soft-sensor modeling algorithms as well as their applications.

Processes with multiple operating modes widely exist in industrial process systems, which may result from multiple product grade demands, feedstock changes, and load variations or seasonal operations [8]. Developing soft sensors for these processes needs to take into account the multimode characteristics that make the processes exhibit non-Gaussian and nonlinear behaviors. A single model may not perform satisfactorily, and multiple models, each of which accounts for one operating mode, are desirable. In addition, due to the noisy measurement environment and transmission disturbances, industrial processes are inherently stochastic [9], [10]. Therefore, the probabilistic models are more suitable for dealing with the uncertainties in contrast to their deterministic counterparts [11], [12]. Finite mixture models (FMMs) under probabilistic learning framework are commonly used strategies to meet the above-mentioned two requirements. The most well-known type of FMM is the Gaussian mixture models (GMMs). Indeed, the GMM is powerful in modeling the multimode or other types of non-Gaussian characteristics. Besides, the GMM can also account for process uncertainties.

Manuscript received November 7, 2017; revised June 6, 2018; accepted July 12, 2018. Manuscript received in final form July 14, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0304203 and in part by the Natural Science Foundation of China under Grant 61703367. Recommended by Associate Editor H. Wang. *(Corresponding author: Weiming Shao.)*

The authors are with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310007, China (e-mail: shaowm@zju.edu.cn; gezhiqiang@zju.edu.cn; songzhihuan@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2018.2856845

Therefore, the GMM has recently established itself as a popular tool for developing soft sensors for processes with multiple operating conditions [10], [13]–[20]. Other types of FMM have also been reported for soft-sensor development, such as the mixture of GPR models [1] and mixture of PCR models [21], but we are focused on the GMM-based ones in this paper.

Those GMM-based soft sensors can generally be categorized into two groups. In the first group [13]–[17], secondary and primary variables are separately manipulated, and the GMM is used as a clustering method for mode identification. Then, the localized models are trained for each mode using regression algorithms such as the kernel PLS [13] and GPR [14]. In the other group [10], [18]–[20], the Gaussian mixture regression (GMR) is developed, which treats secondary and primary variables together and learns their joint probability density functions (PDFs) in each mode. The functional relationship for soft-sensor development can be derived directly from the joint PDFs. For instance, in [18], soft sensors were trained using the GMR for multimode/multiphase processes, showing the priority of the GMR over the GMM, since the GMR learns model parameters together instead of separately, and thus is not constrained by the number of samples in each mode. Furthermore, a variational GMR was developed in [20] to model non-Gaussian processes, indicating that the Bayesian treatment can not only determine the number of Gaussian components (GCs) automatically but also improves the estimation performance compared with the GMR.

Although the GMR-based soft sensors outperform those based on the two-stage GMM, the GMR requires sufficient labeled samples for reliable PDF estimations. However, in soft-sensor applications, labeling samples could be expensive and infrequent due to certain technical or economic limitations such as the time-consuming laboratory analysis or high investment of mass spectrometer. As a result, labeled samples are usually rare and the success of GMR may not be guaranteed because insufficiency of labeled samples often leads to overfitting and unreliable estimations of PDFs and model selection, particularly when the dimensionality of process variables is high. On the contrary, there are large amounts of unlabeled samples, which also contain useful information yet have not been utilized by the GMR. Exploiting both labeled and unlabeled samples, namely the semisupervised learning, is a promising manner to remedy the limitation of merely relying on labeled samples [22]. In fact, the semisupervised learning strategy has proven to be effective in the GMM for classification purposes [23], [24]. Nevertheless, to the best of our knowledge, its counterpart for the regression purpose, especially for the soft-sensor development, has not been found reported.

In order to deal with the above-discussed deficiency of GMR-based soft sensors, this paper proposes a semisupervised GMR (S^2GMR), where both labeled and unlabeled samples where effective, and the PDF parameters and regression coefficients for each component are learned simultaneously by the expectation–maximization (EM) algorithm. In addition, the Bayesian information criterion (BIC) is employed to perform model selection to determine the number of GCs.

The remaining parts of this paper are organized as follows. In Section II, the standard GMR is briefly introduced and the proposed S^2GMR are elaborated, followed by detailed soft-sensor development procedures using the S^2GMR in Section III. Two case studies are carried out and comprehensive discussions are presented in Section IV. Finally, the conclusion is put forward in Section V.

II. Semisupervised Gaussian Mixture Regression

In this section, the standard GMR [25] is first briefly introduced; then, the detailed model formulation and parameter learning for the S^2GMR are presented.

A. Gaussian Mixture Regression

Let $\boldsymbol{x}_i \in \mathbb{R}^d$ and $\boldsymbol{y}_i \in \mathbb{R}^m$ be the d-dimensional input and m-dimensional output vectors of the ith sample, respectively; let $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}^T$ and $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}^T$ be the input and output data matrices with n samples, respectively. In the GMR, $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ merge into a new sample $\boldsymbol{t}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathbb{R}^{d+m}$, which is assumed to obey a mixture of Gaussian distributions with a total of K components. That is,

$$p(\boldsymbol{t}_i) = \sum_{k=1}^{K} \alpha_k p_k(\boldsymbol{t}_i) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\boldsymbol{t}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1}$$

where α_k and $p_k(\boldsymbol{t}_i)$ represent the mixing coefficient and PDF of $\boldsymbol{t}_i$ for the kth component, respectively, $\mathcal{N}(\cdot|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ represents the PDF of Gaussian distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. After parameter identification, which is usually realized by the EM algorithm, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are partitioned into

$$\boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_k^{\boldsymbol{x}} \\ \boldsymbol{\mu}_k^{\boldsymbol{y}} \end{bmatrix}, \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k^{\boldsymbol{x}} & \boldsymbol{\Sigma}_k^{\boldsymbol{xy}} \\ \boldsymbol{\Sigma}_k^{\boldsymbol{yx}} & \boldsymbol{\Sigma}_k^{\boldsymbol{y}} \end{bmatrix}. \tag{2}$$

Then, for the kth component, by using linear Gaussian operations the PDF of $\boldsymbol{y}$ conditioning on $\boldsymbol{x}$ is calculated as

$$p_k(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{y} \middle| \boldsymbol{\mu}_k^{\boldsymbol{y}|\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{y}|\boldsymbol{x}}\right) \tag{3}$$

where $\boldsymbol{\mu}_k^{\boldsymbol{y}|\boldsymbol{x}} = \boldsymbol{\mu}_k^{\boldsymbol{y}} + \boldsymbol{\Sigma}_k^{\boldsymbol{yx}}(\boldsymbol{\Sigma}_k^{\boldsymbol{x}})^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k^{\boldsymbol{x}})$ and $\boldsymbol{\Sigma}_k^{\boldsymbol{y}|\boldsymbol{x}} = \boldsymbol{\Sigma}_k^{\boldsymbol{y}} - \boldsymbol{\Sigma}_k^{\boldsymbol{yx}}(\boldsymbol{\Sigma}_k^{\boldsymbol{x}})^{-1}\boldsymbol{\Sigma}_k^{\boldsymbol{xy}}$. Therefore, the overall predictive PDF of $\boldsymbol{y}$ given $\boldsymbol{x}$ is obtained as

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{k=1}^{K} \gamma_k^{\boldsymbol{x}} p_k(\boldsymbol{y}|\boldsymbol{x}) = \sum_{k=1}^{K} \gamma_k^{\boldsymbol{x}} \mathcal{N}\left(\boldsymbol{y} \middle| \boldsymbol{\mu}_k^{\boldsymbol{y}|\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{y}|\boldsymbol{x}}\right) r \tag{4}$$

where $\gamma_k^{\boldsymbol{x}} = \alpha_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}) / \sum_{k=1}^{K} \alpha_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}})$.

B. Semisupervised Gaussian Mixture Regression

The S^2GMR is elaborated by taking the single-output case as an example but its extension to the multioutput case is straightforward. In the S^2GMR, the marginal distribution over $\boldsymbol{x}$ is assumed to be Gaussian, and the functional relationship between y and $\boldsymbol{x}$ for the kth component is considered to be linear, which leads to

$$p_k(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \middle| \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\right) \tag{5}$$

$$p_k(y|\boldsymbol{x}) = \mathcal{N}\left(y \middle| \boldsymbol{x}^T \boldsymbol{\omega}_k + \psi_k, \sigma_k^2\right) \tag{6}$$

where $\boldsymbol{\omega}_k$ and ψ_k are the regression coefficients, and σ_k^2 is the variance of the zero-mean Gaussian measurement noise.

Accordingly, the joint PDF of $\boldsymbol{x}$ and y for the kth component is determined as

$$p_k(\boldsymbol{x}, y) = \mathcal{N}\big(\boldsymbol{x}, y \big| \boldsymbol{\mu}_k^{xy}, \boldsymbol{\Sigma}_k^{xy}\big) \tag{7}$$

where

$$\boldsymbol{\mu}_k^{xy} = \begin{bmatrix} \boldsymbol{\mu}_k^{\boldsymbol{x}} \\ \boldsymbol{\omega}_k^T \boldsymbol{\mu}_k^{\boldsymbol{x}} + \psi_k \end{bmatrix}, \quad \boldsymbol{\Sigma}_k^{xy} = \begin{bmatrix} \boldsymbol{\Sigma}_k^{\boldsymbol{x}} & \boldsymbol{\Sigma}_k^{\boldsymbol{x}} \boldsymbol{\omega}_k \\ \boldsymbol{\omega}_k^T \boldsymbol{\Sigma}_k^{\boldsymbol{x}} & \boldsymbol{\omega}_k^T \boldsymbol{\Sigma}_k^{\boldsymbol{x}} \boldsymbol{\omega}_k + \sigma_k^2 \end{bmatrix}.$$

Up to now, the overall joint PDF of $\boldsymbol{x}$ and y and the overall marginal PDF of $\boldsymbol{x}$ can be calculated as

$$p(\boldsymbol{x}, y) = \sum_{k=1}^{K} \alpha_k p_k(\boldsymbol{x}, y) = \sum_{k=1}^{K} \alpha_k \mathcal{N}\big(\boldsymbol{x}, y \big| \boldsymbol{\mu}_k^{xy}, \boldsymbol{\Sigma}_k^{xy}\big) \tag{8}$$

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha_k p_k(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha_k \mathcal{N}\big(\boldsymbol{x} \big| \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\big). \tag{9}$$

In order to learn the mode parameters of the S^{2}GMR, which are denoted as $\boldsymbol{\Theta} = \{\alpha_k, \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}, \boldsymbol{\omega}_k, \psi_k, \sigma_k^2\}_{k=1}^K$, based on the EM algorithm, we develop an iterative learning procedure. Assume the training data set consists of n_l and n_u labeled and unlabeled samples that are denoted as $\{\boldsymbol{X}_l, \boldsymbol{Y}_l\} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n_l}$ and $\boldsymbol{X}_u = \{\boldsymbol{x}_j\}_{j=n_l+1}^{n_l+n_u}$, respectively.

In the expectation step (E-step), the posterior PDFs over latent assignment variables z_i given $(\boldsymbol{x}_i, y_i)$ and z_j given $\boldsymbol{x}_j$ are calculated as

$$p(z_i = k|\boldsymbol{x}_i, y_i) = \frac{\alpha_k \mathcal{N}\big(\boldsymbol{x}_i, y_i \big| \boldsymbol{\mu}_k^{xy}, \boldsymbol{\Sigma}_k^{xy}\big)}{\sum_{k=1}^{K} \alpha_k \mathcal{N}\big(\boldsymbol{x}_i, y_i \big| \boldsymbol{\mu}_k^{xy}, \boldsymbol{\Sigma}_k^{xy}\big)} \tag{10}$$

$$p(z_j = k|\boldsymbol{x}_j) = \frac{\alpha_k \mathcal{N}\big(\boldsymbol{x}_j \big| \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\big)}{\sum_{k=1}^{K} \alpha_k \mathcal{N}\big(\boldsymbol{x}_j \big| \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\big)} \tag{11}$$

where we have used (8) and (9), and the prior of the kth component that $p(z_i = k) = p(z_j = k) = \alpha_k$. To keep subsequent derivations as concise as possible, $p(z_i = k \big| \boldsymbol{x}_i, y_i)$ and $p(z_j = k|\boldsymbol{x}_j)$ are denoted as $\gamma_k^{\,i}$ and $\gamma_k^{\,j}$, respectively.

In the maximization step (M-step), with the assumption that all data samples are independent with each other, the complete data log-likelihood function is defined as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\Theta}) = & \sum_{i=1}^{n_l} \sum_{k=1}^{K} \gamma_k^{\,i} \ln p_k(y_i \big| \boldsymbol{x}_i) \\ & + \sum_{i=1}^{n_l} \sum_{k=1}^{K} \gamma_k^{\,i} \ln p_k(\boldsymbol{x}_i) + \sum_{j=n_l+1}^{n_l+n_u} \sum_{k=1}^{K} \gamma_k^{\,j} \ln p_k(\boldsymbol{x}_j) \\ & + \sum_{i=1}^{n_l} \sum_{k=1}^{K} \gamma_k^{\,i} \ln \alpha_k + \sum_{j=n_l+1}^{n_l+n_u} \sum_{k=1}^{K} \gamma_k^{\,j} \ln \alpha_k. \end{aligned} \tag{12}$$

Combining the constraint $\sum_{k=1}^{K} \alpha_k = 1$ with the Lagrange multiplier β, we have

$$\widetilde{\mathcal{L}}(\boldsymbol{\Theta}) = \mathcal{L}(\boldsymbol{\Theta}) + \beta \left(\sum_{k=1}^{K} \alpha_k - 1 \right). \tag{13}$$

Setting the derivatives of $\widetilde{\mathcal{L}}(\boldsymbol{\Theta})$ with respect to α_k to zeros and eliminating β, we can get the learning formula for α_k as

$$\alpha_k = \big(n_k^l + n_k^u\big) \big/ (n_l + n_u) \tag{14}$$

where $n_k^l = \sum_{i=1}^{n_l} \gamma_k^{\,i}$ and $n_k^u = \sum_{j=n_l+1}^{n_l+n_u} \gamma_k^{\,j}$.

Setting the derivatives of $\mathcal{L}(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\mu}_k^{\boldsymbol{x}}$, $(\boldsymbol{\Sigma}_k^{\boldsymbol{x}})^{-1}$, $\widetilde{\boldsymbol{\omega}}_k$, and σ_k^2 to zeros leads to

$$\boldsymbol{\mu}_k^{\boldsymbol{x}} = \left(\sum_{i=1}^{n_l} \gamma_k^{\,i} \boldsymbol{x}_i + \sum_{j=n_l+1}^{n_l+n_u} \gamma_k^{\,j} \boldsymbol{x}_j \right) \Bigg/ \big(n_k^l + n_k^u\big) \tag{15}$$

$$\boldsymbol{\Sigma}_k^{\boldsymbol{x}} = \frac{\sum_{i=1}^{n_l} \gamma_k^{\,i} \overline{\boldsymbol{x}}_k^i \big(\overline{\boldsymbol{x}}_k^i\big)^T + \sum_{j=n_l+1}^{n_l+n_u} \gamma_k^{\,j} \overline{\boldsymbol{x}}_k^j \big(\overline{\boldsymbol{x}}_k^j\big)^T}{n_k^l + n_k^u} \tag{16}$$

$$\widetilde{\boldsymbol{\omega}}_k = \big(\widetilde{\boldsymbol{X}}_l^T \boldsymbol{\Gamma}_k \widetilde{\boldsymbol{X}}_l\big)^{-1} \widetilde{\boldsymbol{X}}_l^T \boldsymbol{\Gamma}_k \boldsymbol{Y}_l \tag{17}$$

$$\sigma_k^2 = \sum_{i=1}^{n_l} \gamma_k^{\,i} \big(y_i - \widetilde{\boldsymbol{x}}_i^T \widetilde{\boldsymbol{\omega}}_k\big)^2 \big/ n_k^l \tag{18}$$

where $\overline{\boldsymbol{x}}_k^i = \boldsymbol{x}_i - \boldsymbol{\mu}_k^{\boldsymbol{x}}$, $\overline{\boldsymbol{x}}_k^j = \boldsymbol{x}_j - \boldsymbol{\mu}_k^{\boldsymbol{x}}$, $\widetilde{\boldsymbol{X}}_l = (\boldsymbol{X}_l, \boldsymbol{1})$, $\boldsymbol{1} = [1, \ldots, 1]^T \in \mathbb{R}^{n_l}$, $\widetilde{\boldsymbol{\omega}}_k = \begin{bmatrix} \boldsymbol{\omega}_k \\ \psi_k \end{bmatrix}$, $\widetilde{\boldsymbol{x}}_i = \begin{bmatrix} \boldsymbol{x}_i \\ 1 \end{bmatrix}$, and $\boldsymbol{\Gamma}_k = \mathrm{diag}(\gamma_k^{\,1}, \ldots, \gamma_k^{\,n_l})$.

Detailed derivations for (12) and (15)–(18) are placed in the Appendix.

For further antioverfitting and alleviating the singularity problem in (18), the S^{2}GMR can be equipped with the Bayesian regularization if necessary, which imposes a Gaussian prior on the regression coefficients for each component. That is, $p(\widetilde{\boldsymbol{\omega}}_k) = \mathcal{N}(\widetilde{\boldsymbol{\omega}}_k | \boldsymbol{0}, \lambda_k^{-1} \mathbf{I})$, where $\mathbf{I}$ represents the unit matrix and λ_k are the precision parameters. As a result, the *maximum a posteriori* estimates for $\widetilde{\boldsymbol{\omega}}_1, \ldots, \widetilde{\boldsymbol{\omega}}_K$ are obtained by maximizing the following objective function with respect to $\widetilde{\boldsymbol{\omega}}_1, \ldots, \widetilde{\boldsymbol{\omega}}_K$ (where those constant terms have been omitted)

$$\begin{aligned} & \ln p(\widetilde{\boldsymbol{\omega}}_1, \ldots, \widetilde{\boldsymbol{\omega}}_K \big| \boldsymbol{X}_l, \boldsymbol{Y}_l, \boldsymbol{X}_u) \\ & \quad = \ln p(\boldsymbol{X}_l, \boldsymbol{Y}_l, \boldsymbol{X}_u \big| \widetilde{\boldsymbol{\omega}}_1, \ldots, \widetilde{\boldsymbol{\omega}}_K) + \sum_{k=1}^{K} \ln p(\widetilde{\boldsymbol{\omega}}_k) \\ & \quad = \mathcal{L}(\boldsymbol{\Theta}) - \frac{1}{2} \sum_{k=1}^{K} \lambda_k \widetilde{\boldsymbol{\omega}}_k^T \widetilde{\boldsymbol{\omega}}_k. \end{aligned} \tag{19}$$

By combining (19) and (A.10), we can obtain that

$$\begin{aligned} & \frac{\partial}{\partial \widetilde{\boldsymbol{\omega}}_k} \ln p(\widetilde{\boldsymbol{\omega}}_1, \ldots, \widetilde{\boldsymbol{\omega}}_K \big| \boldsymbol{X}_l, \boldsymbol{Y}_l, \boldsymbol{X}_u) \\ & \qquad = \frac{1}{\sigma_k^2} \widetilde{\boldsymbol{X}}_l^T \boldsymbol{\Gamma}_k (\boldsymbol{Y}_l - \widetilde{\boldsymbol{X}}_l \widetilde{\boldsymbol{\omega}}_k) - \lambda_k \widetilde{\boldsymbol{\omega}}_k. \end{aligned} \tag{20}$$

Setting (20) to $\boldsymbol{0}$, we have

$$\widetilde{\boldsymbol{\omega}}_k = \big(\widetilde{\boldsymbol{X}}_l^T \boldsymbol{\Gamma}_k \widetilde{\boldsymbol{X}}_l + \lambda_k \sigma_k^2 \mathbf{I}\big)^{-1} \widetilde{\boldsymbol{X}}_l^T \boldsymbol{\Gamma}_k \boldsymbol{Y}_l. \tag{21}$$

For both the GMR and S^{2}GMR, if the number of GC's K is unknown, commonly used criteria for automatic model selection include the *Akaike information criterion* (AIC), *absolute increment log-likelihood* (AIL), and BIC, and so on [18], [23], [26]. Considering the AIC and AIL are prone to be overfitted, we employ the BIC for model selection, which is defined as [29]

$$\mathrm{BIC}(K) = -2 \ln p(\mathcal{D} \big| \boldsymbol{\Theta}_K) + M \ln N \tag{22}$$

where $\mathcal{D}$ and N refer to the training samples and their quantities, respectively, M is the number of parameters that need to be learned, and $\boldsymbol{\Theta}_K$ stand for the learned parameters with K GCs. The BIC could find a balance between fitting training data and suppressing model complexity, and the optimal K is determined as the one which can minimize (22).

It is noted from the above-mentioned derivations that updating of model parameters are coupled with each other, which requires iterative learning. The convergence can be diagnosed by monitoring the log-likelihood function defined as

$$\ln p(\mathcal{D}|\boldsymbol{\Theta}) = \sum_{i=1}^{n_l} \ln\left(\sum_{k=1}^{K} \alpha_k \mathcal{N}\left(\boldsymbol{x}_i, y_i \middle| \boldsymbol{\mu}_k^{\boldsymbol{x}y}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}y}\right)\right) + \sum_{j=n_l+1}^{n_l+n_u} \ln\left(\sum_{k=1}^{K} \alpha_k \mathcal{N}\left(\boldsymbol{x}_j \middle| \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\right)\right) \tag{23}$$

and the convergence criterion can be set as

$$\left|\frac{\ln p(\mathcal{D}|\boldsymbol{\Theta}^{(t+1)}) - \ln p(\mathcal{D}|\boldsymbol{\Theta}^{(t)})}{\ln p(\mathcal{D}|\boldsymbol{\Theta}^{(t)})}\right| \times 100\% < \epsilon \tag{24}$$

where $\boldsymbol{\Theta}^{(t)}$ represent the parameters learned at the tth iteration, and ϵ represents the user-defined threshold value. In addition, the initialization of such iteration process can be aided by the k-means clustering method using the labeled samples $\{\boldsymbol{t}_i\}_{i=1}^{n_l}$, where $\boldsymbol{t}_i = (\boldsymbol{x}_i, y_i)$, and the centers of K clusters are denoted as $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_K$. Furthermore, by using the 1-of-K coding scheme, a set of binary indicator variables τ_k^i (for $k = 1, \cdots, K$) for each sample $\boldsymbol{t}_i$ can be defined as [29]

$$\tau_k^i = \begin{cases} 1, & \text{if } k = \arg\min_j \|\boldsymbol{t}_i - \boldsymbol{c}_j\|^2 \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

Then, the clustering centers need to be updated as

$$\boldsymbol{c}_k = \sum_{i=1}^{n_l} \tau_k^i \boldsymbol{t}_i \Big/ \sum_{i=1}^{n_l} \tau_k^i. \tag{26}$$

The objective function J of the k-means clustering method is given as

$$J = \sum_{i=1}^{n_l} \sum_{k=1}^{K} \tau_k^i \|\boldsymbol{t}_i - \boldsymbol{c}_k\|^2. \tag{27}$$

Therefore, the convergence criterion of the k-means clustering method can be defined as

$$|J^{(t)} - J^{(t+1)}|/J^{(t)} \times 100\% < \epsilon \tag{28}$$

where $J^{(t)}$ represents the cost function value at the tth iteration.

After detecting the convergence of the k-means clustering method, which is assured [29], the indices of samples allocated to the kth cluster can be obtained as

$$\boldsymbol{I}_k = \left\{i \middle| \tau_k^i = 1, i = 1, \ldots, n_l\right\}. \tag{29}$$

Correspondingly, samples that are assigned to the kth cluster are denoted as $\boldsymbol{X}_l^k = \{\boldsymbol{x}_i\}$ and $\boldsymbol{Y}_l^k = \{y_i\}$ for $i \in \boldsymbol{I}_k$, and the initialized model parameters can be calculated as

$$\alpha_k = |\boldsymbol{I}_k|/n_l \tag{30}$$

$$\boldsymbol{\mu}_k^{\boldsymbol{x}} = \sum_{i \in \boldsymbol{I}_k} \boldsymbol{x}_i \Big/ |\boldsymbol{I}_k| \tag{31}$$

$$\boldsymbol{\Sigma}_k^{\boldsymbol{x}} = \sum_{i \in \boldsymbol{I}_k} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{\boldsymbol{x}}\right)^T \Big/ |\boldsymbol{I}_k| \tag{32}$$

$$\widetilde{\boldsymbol{\omega}}_k = \left(\left(\widetilde{\boldsymbol{X}}_l^k\right)^T \widetilde{\boldsymbol{X}}_l^k\right)^{-1} \left(\widetilde{\boldsymbol{X}}_l^k\right)^T \boldsymbol{Y}_l^k \tag{33}$$

$$\sigma_k^2 = \sum_{i \in \boldsymbol{I}_k} \left(y_i - \widetilde{\boldsymbol{x}}_i^T \widetilde{\boldsymbol{\omega}}_k\right)^2 \Big/ |\boldsymbol{I}_k| \tag{34}$$

where $|\boldsymbol{I}_k|$ represents the number of samples in the kth cluster, and $\widetilde{\boldsymbol{X}}_l^k = (\boldsymbol{X}_l^k, \boldsymbol{1})$ with $\boldsymbol{1} \in \mathbb{R}^{|\boldsymbol{I}_k|}$.

Parameter learning procedures for the S^{2}GMR are summarized in Algorithm 1.

Remark:

1) The iterations in Algorithm 1 for parameter updating are guaranteed to converge, as at the M-step of each iteration, the log-likelihood function defined by (23) is increased; meanwhile, the log-likelihood function is bounded with nonsingular covariance matrices.
2) It is not guaranteed that the EM algorithm finds the globally optimal model parameters. In practical applications, validation data set is preferable for selecting satisfactory initial centers for the k-means clustering method that behaves well on the validation data set.

III. Soft-Sensor Development Based On S^{2}GMR

Based on the S^{2}GMR algorithm, a soft-sensor model can be developed for estimating the true value (y_q) of a primary variable given a sample ($\boldsymbol{x}_q$) of the secondary variable.

First, the posterior distribution over latent variable z_q given $\boldsymbol{x}_q$ is calculated according to

$$p(z_q = k|\boldsymbol{x}_q) = \frac{\alpha_k \mathcal{N}\left(\boldsymbol{x}_q \middle| \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\right)}{\sum_{k=1}^{K} \alpha_k \mathcal{N}\left(\boldsymbol{x}_q \middle| \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\right)}. \tag{35}$$

Again, we denote $p(z_q = k|\boldsymbol{x}_q)$ as γ_k^q for conciseness. Then, the PDF of y_q conditioned on $\boldsymbol{x}_q$ is computed as

$$\begin{aligned} p(y_q|\boldsymbol{x}_q) &= \sum_{k=1}^{K} p(z_q = k|\boldsymbol{x}_q) p(y_q|\boldsymbol{x}_q, z_q = k) \\ &= \sum_{k=1}^{K} \gamma_k^q \mathcal{N}\left(y_q \middle| \boldsymbol{x}_q^T \boldsymbol{\omega}_k + \psi_k, \sigma_k^2\right). \end{aligned} \tag{36}$$

Therefore, the estimation of y_q is determined as

$$\hat{y}_q = \mathbb{E}[y_q|\boldsymbol{x}_q] = \sum_{k=1}^{K} \gamma_k^q \left(\boldsymbol{x}_q^T \boldsymbol{\omega}_k + \psi_k\right) \tag{37}$$

where $\mathbb{E}[\cdot]$ represents the expectation of the corresponding random variable.

In addition to the estimated value of primary variable, the S^{2}GMR-based soft sensor could provide the estimation uncertainties. To be specific, the estimation variance using (37)

Algorithm 1 Parameter Learning Procedures for the S^{2}GMR

for $K = K_{min}, \cdots, K_{max}$ **do**
  Initialization by k**-means**
  Randomly select K samples from labeled dataset as initial centers $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_K$; set $t = 0$;
  while $t < maximum\ iteration\ times$ **do**
    Set $t = t + 1$;
    for $k = 1, \cdots, K$; $i = 1, \cdots, n_l$ **do**
      Calculate τ_k^i using Eq. (25);
      Update $\boldsymbol{c}_k$ using Eq. (26);
    end for
    Calculate the objective function using Eq. (27);
    if the criterion in Eq. (28) is satisfied **then**
      Terminate **while**;
    end if
  end while
  for $k = 1, \cdots, K$ **do**
    Determine $\boldsymbol{I}_k$ using Eq. (29);
    Calculate initial parameters α_k, $\boldsymbol{\mu}_k^{\boldsymbol{x}}$, $\boldsymbol{\Sigma}_k^{\boldsymbol{x}}$, $\widetilde{\boldsymbol{\omega}}_k$ and σ_k^2 using Eqs. (30)$\sim$(34), respectively;
  end for
  Parameter learning
  Set $t = 0$;
  while $t < maximum\ iteration\ times$ **do**
    Set $t = t + 1$;
    for $k = 1, \cdots, K$; $i = 1, \cdots, n_l$; $j = n_l + 1, \cdots, n_l + n_u$ **do**
      Calculate γ_k^i and γ_k^j using Eq. (10) and (11), respectively, with model parameters obtained from the previous iteration;
      Calculate n_k^l and n_k^u;
    end for
    for $k = 1, \cdots, K$ **do**
      Update α_k, $\boldsymbol{\mu}_k^{\boldsymbol{x}}$, $\boldsymbol{\Sigma}_k^{\boldsymbol{x}}$ and σ_k^2 using Eqs. (14)$\sim$(16) and (18), respectively;
      Update $\widetilde{\boldsymbol{\omega}}_k$ using Eq. (17) or (21);
    end for
    Calculate the objective function using Eq. (23);
    if the criterion in Eq. (24) is satisfied **then**
      Terminate **while**;
    end if
  end while
  Calculate BIC(K) using Eq. (22);
end for
Select the optimal number of GC's K^* as the one that minimizes $\{\text{BIC}(K_{min}), \cdots, \text{BIC}(K_{max})\}$, and determine the final model parameters as $\boldsymbol{\Theta}^* = \boldsymbol{\Theta}_{K^*}$.

can be calculated as

$$\sigma_q^2 = \int p(y_q|\boldsymbol{x}_q) y_q^2 \mathrm{d}y_q - (\mathbb{E}[y_q|\boldsymbol{x}_q])^2 = \sum_{k=1}^{K} \gamma_k^q \left(\sigma_k^2 + \left(\boldsymbol{x}_q^T \boldsymbol{\omega}_k + \psi_k\right)^2\right) - \hat{y}_q^2. \quad (38)$$

With (38), the uncertainties of estimated primary variable can be obtained. For example, the confidence interval within twice standard deviations is determined as

$$\hat{y}_q - 2\sigma_q < y_q < \hat{y}_q + 2\sigma_q. \quad (39)$$

Such confidence interval plays a role of performance assessment for the soft-sensor model and could be useful for many purposes, for example, decision making, abnormal sample classification to guarantee reliable online model update, online hardware analyzer calibration, and so on [27], [28].

IV. Case Studies

In this section, the S^{2}GMR is first investigated using a numerical example and then applied to develop a soft sensor for estimating the oxygen concentration in a real-life industrial primary reformer. The performance of state-of-the-art soft sensing methods including the PLS [30], ELM [31], and GMR [18] are presented as benchmarks. For the PLS, the dimensionality of latent space is determined by tenfold cross validation, so is the number of hidden neurons for the ELM, where the activation function is selected as the sigmoid function. The threshold values for convergence diagnosis for both k-means and EM algorithms are set as 10^{-4}. Certain proportions of training samples are uniformly chosen as labeled samples, and the rest ones are treated as unlabeled. The prediction accuracy is measured using average root mean squares error (avgRMSE) of 100 independent simulations obtained on the testing data set. The RMSE is defined as

$$\text{RMSE} = \sqrt{\sum_{n=1}^{N_t} (y_n - \hat{y}_n)^2 \Big/ N_t} \quad (40)$$

where y_n and $\hat{y}_n$ represent the true value and the predicted value of nth test sample, respectively, and N_t is the number of testing samples. The configurations of desktop computer used for carrying out all the experiments are as follows. CPU: Core i7-6700 (3.4 GHz $\times$ 2), RAM: 16 GB, OS: Windows 7, and Software: MATLAB (R2013b). The CPU time (CPT) spent in offline model training (CPT_{trn}, in seconds) and in calculating online predictions (CPT_{tst}, in seconds) are used to evaluate the computational efficiency for various methods. Note that the commands tic and toc in MATLAB software are used for time measurement.

A. Performance Evaluation by Numerical Example

Let a 2-D input vector $\boldsymbol{x} = (x_1, x_2)^T$ and a scalar output y follow the relationship described by (5) and (6) with three GCs, where the configurations of each component are listed in Table I.

The data distributions are visualized in Fig. 1 with samples rearranged by their mode indices, which present clear multimode characteristics. In the simulation, 200–1000 samples were generated for model parameter learning and performance evaluation, respectively.

In this example, the number of GCs is set as three for both the GMR and S^{2}GMR since it is known. Predictions of y obtained by the PLS, ELM, GMR, and S^{2}GMR without the Bayesian regularization are illustrated in Fig. 2, where the labeling rate is 10%. It can be found that although

TABLE I
CONFIGURATIONS OF THE THREE GCS

	$k=1$	$k=2$	$k=3$
α_k	20%	30%	50%
$\boldsymbol{\mu}_k^{\boldsymbol{x}}$	$(0,\ 2)^{\mathrm{T}}$	$(4,\ 6)^{\mathrm{T}}$	$(4,\ 0)^{\mathrm{T}}$
$\boldsymbol{\Sigma}_k^{\boldsymbol{x}}$	$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$	$\begin{pmatrix} 3 & -1 \\ -1 & 1.5 \end{pmatrix}$
$\boldsymbol{\omega}_k$	$(1,\ 1)^{\mathrm{T}}$	$(1,\ -1)^{\mathrm{T}}$	$(-1,\ 1)^{\mathrm{T}}$
ψ_k	0	0	0
σ_k^2	0.5	0.5	0.5

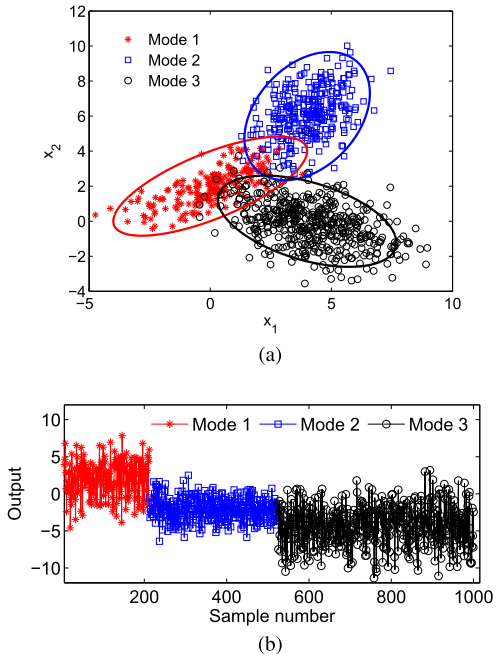


Fig. 1. Visualizations of characteristics of synthetic data. (a) Input space. (b) Output space.

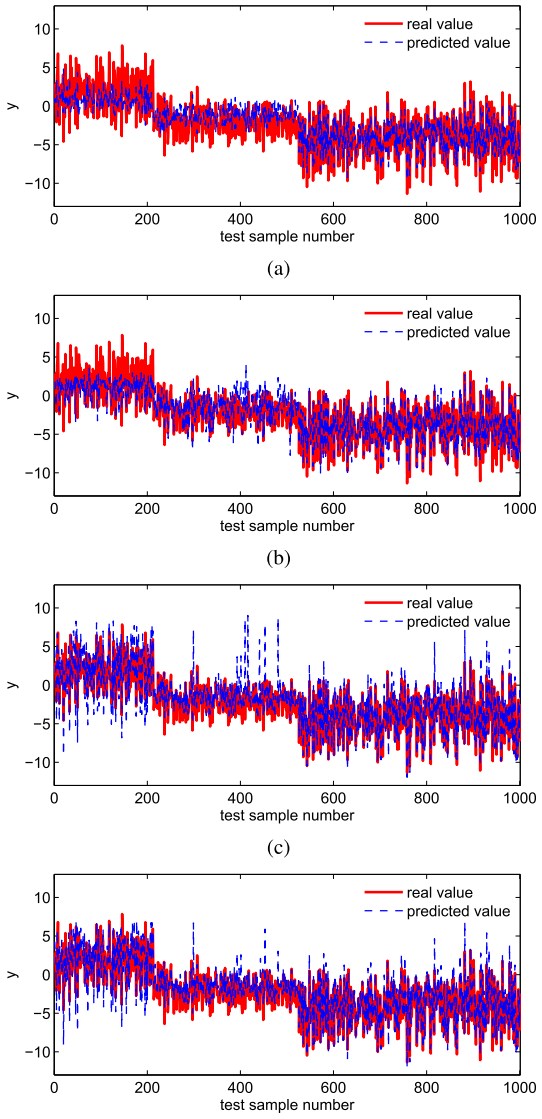


Fig. 2. Predictions of y obtained by various methods (LR = 10%). (a) Predictions of y obtained by PLS. (b) Predictions of y obtained by ELM. (c) Predictions of y obtained by GMR. (d) Predictions of y obtained by S^{2}GMR.

predictions by the PLS are more "stable," they deviate from the true values, especially for samples generated from first two modes. Despite that the ELM can make an improvement, its predictions for samples from the first mode are not good. In contrast, both the GMR and S^{2}GMR perform better than the PLS and ELM in all the three modes, but in contrast with the GMR, the S^{2}GMR can provide more accurate predictions in some areas such as those from the second mode.

Scatter plot comparisons among the PLS, GMR, ELM, and S^{2}GMR presented in Fig. 3 could provide us with more insights. It indicates that predictions obtained by the PLS and ELM are distributed more scattered (implying larger estimation error), and prediction biases are observed, while the GMR and S^{2}GMR have more balanced and tight scatters around the black diagonal line (meaning higher predictive accuracy). In addition, although the performance of GMR and S^{2}GMR is comparative when y values are below -5 (corresponding to the third mode), the S^{2}GMR is more accurate than the GMR in those areas, where y values exceed -5 (corresponding to

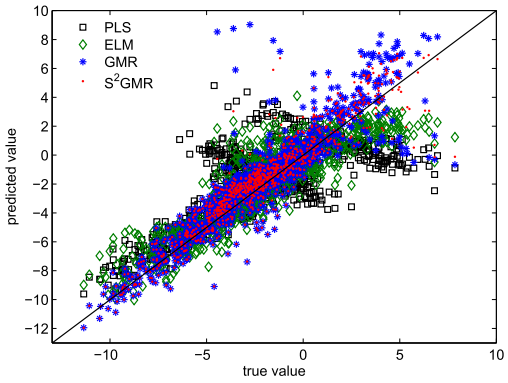


Fig. 3. Scatter plot comparisons among the PLS, ELM, GMR, and S^{2}GMR.

the first and second modes). This is because the third mode has the biggest prior probability (50%) which generated the largest number of samples. As a result, the third mode has

TABLE II

AVERAGE PREDICTIVE RMSES WITH SYNTHETIC DATA SET BY THE PLS, GMR, ELM, AND S^{2}GMM

LR	PLS	ELM	GMR	S^{2}GMR
10%	2.3020	1.5599	1.6011	1.2765
20%	2.3279	1.5333	1.4078	1.1927
30%	2.3059	1.4324	1.2783	1.1143
40%	2.2962	1.3641	1.0950	1.0407
50%	2.2980	1.3570	1.0301	1.0652

relatively more labeled samples while the other two modes have less which results in less accuracy. However, with the aid of those unlabeled samples, the S^{2}GMR outperforms the GMR in the first and second modes.

The above-mentioned qualitative comparisons and analyses indicate that the single PLS and ELM models cannot well handle the multimode process while the GMR and S^{2}GMR are more suitable, as they show more powerful abilities in capturing the multimode characteristics. However, the S^{2}GMR is superior to the GMR in those modes with smaller prior probabilities that have fewer samples for model training.

For more in-depth quantitative analyses, the predictive avgRMSEs obtained by various methods with different label rates (LRs) are tabulated in Table II. As can be seen, the addition of labeled samples does not help the PLS too much. Compared to the PLS, the ELM, GMR, and S^{2}GMR demonstrate higher accuracies, and the ELM outperforms the GMR with LR set as 10%, which is because the GMR suffers from overfitting while owing to the cross validation, the ELM does not. Furthermore, with LR set as 40% and 50%, the predictive accuracies of the GMR and S^{2}GMR are comparative. However, when label samples become fewer, the S^{2}GMR starts to show predictive advantages over the GMR. In particular, with LR decreased from 50% to 10%, the estimation performance of GMR has significantly deteriorated while the deterioration of S^{2}GMR is much less. This is because with insufficient labeled samples, the GMR has encountered the overfitting problem, which leads to inaccurate estimations of PDFs for each component, as illustrated in Table III, where LR $=$ 10%, and the parameters with symbols "$\hat{}$" and "$\check{}$" represent the parameters that estimated the GMR and S^{2}GMR, respectively.

Comparing these estimated parameters with their true values listed in Table I, we can find that the estimated mean vectors by GMR and S^{2}GMR are basically correct. However, the estimated prior probabilities and covariance matrices by GMR have significantly distorted, while those estimated $\{\alpha_k, \Sigma_k^{\boldsymbol{x}}\}_{k=1}^3$ by S^{2}GMR are very close to the true values, implying that the S^{2}GMR is more robust against overfitting than the GMR. In fact, the true log-likelihood function value for labeled samples is -103.20, while the estimated values by GMR and S^{2}GMR are -78.95 and -86.68, respectively.

For both the GMR and S^{2}GMR, the final estimations for a query sample are actually a weighted combination of estimations provided by each component model, as shown by (4) and (37), respectively. Therefore, accurate combination

TABLE III

COMPARISONS BETWEEN THE GMR AND S^{2}GMR IN TERMS OF THE ESTIMATED MARGINAL PDFS OF $\boldsymbol{x}$ FOR EACH COMPONENT

	$k=1$	$k=2$	$k=3$
$\hat{\alpha}_k$	0.200	0.350	0.450
$\check{\alpha}_k$	0.189	0.316	0.495
$\hat{\boldsymbol{\mu}}_k^{\boldsymbol{x}}$	$(-0.55, 2.39)^{\mathrm{T}}$	$(4.53, 6.37)^{\mathrm{T}}$	$(4.47, -0.59)^{\mathrm{T}}$
$\check{\boldsymbol{\mu}}_k^{\boldsymbol{x}}$	$(-0.45, 1.98)^{\mathrm{T}}$	$(4.15, 5.99)^{\mathrm{T}}$	$(3.87, 0.12)^{\mathrm{T}}$
$\hat{\boldsymbol{\Sigma}}_k^{\boldsymbol{x}}$	$\begin{pmatrix}0.50, 0.32\\0.32, 0.44\end{pmatrix}$	$\begin{pmatrix}0.41, 0.31\\0.31, 2.45\end{pmatrix}$	$\begin{pmatrix}1.19, -0.01\\-0.01, 0.65\end{pmatrix}$
$\check{\boldsymbol{\Sigma}}_k^{\boldsymbol{x}}$	$\begin{pmatrix}1.37, 0.70\\0.70, 0.80\end{pmatrix}$	$\begin{pmatrix}0.75, 0.48\\0.48, 2.00\end{pmatrix}$	$\begin{pmatrix}3.23, -1.08\\-1.08, 1.37\end{pmatrix}$

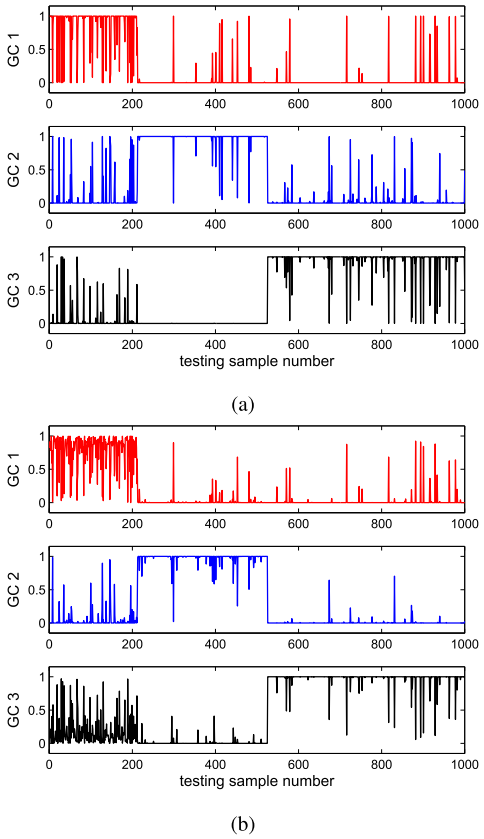


Fig. 4. With LR set as 10%, posterior probabilities of three GCs calculated by (a) GMR and (b) S^{2}GMR.

weights of component models are the prerequisite of precise estimation. For the GMR and S^{2}GMR, given a testing sample, the combinated weight of each component model's estimation is understood as the posterior probability of the corresponding component, which is visualized in Fig. 4, with 10% samples labeled in the training data set.

In the test data set, 213, 321, and 475 samples were generated from the first, second, and third component, respectively. Therefore, the true posterior probabilities of the first component for given test samples 1–213 are 1's and are 0's for given test samples 214–1000; for the second component, the true posterior probabilities for given test samples 214–525 are 1's and are 0's for given the rest ones;

TABLE IV

AVERAGE CPT (IN SECONDS) CONSUMED BY VARIOUS METHODS FOR THE NUMERICAL EXAMPLE

LR	$CPT_{trn}(\times 10^{-2}s)$				$CPT_{tst}(\times 10^{-2}s)$			
	PLS	ELM	GMR	S^2GMR	PLS	ELM	GMR	S^2GMR
10%	1.7	0.3	3.3	22.4	0.86	5.8	10.4	9.0
20%	1.7	0.3	4.7	15.6	0.90	6.1	10.5	9.1
30%	1.9	0.4	6.2	16.7	0.93	6.1	10.0	8.9
40%	1.6	0.5	6.5	15.1	0.84	6.1	10.0	9.1
50%	1.8	0.5	5.9	15.6	0.93	6.1	10.1	9.5

for the third component, the true posterior probabilities for given test samples 526–1000 are 1's and are 0's for given test samples 1–525. Such settings here are merely used for performance investigation. Fig. 4 manifests that the posterior probabilities of three GCs computed by both the GMR and S^2GMR are not 100% accurate, which results from the overlap of test samples in an input space as shown in Fig. 1(a). However, the S^2GMR could perform better, as the posterior probabilities of one component are less disturbed by the other two modes, which demonstrates the reliability of S^2GMR owing to more accurate estimations of marginal PDFs. In specific, for the first and second components, the accuracies of posterior probabilities obtained by the GMR and S^2GMR are almost comparable; while for the third component with the largest number of test samples among three GCs, the posterior probabilities computed by the S^2GMR are apparently more accurate than those computed by the GMR.

The CPT consumed by various methods is tabulated in Table IV. As can be seen, for all investigated methods, the online consumed time for making predictions can be negligible. In terms of offline model training, the ELM is the fastest as it only needs to solve a linear equation without iterative learning. Also, it is noted that the S^2GMR requires more time for parameter learning than the GMR because in the S^2GMR, calculating posterior distributions over latent variables associated with those unlabeled samples are necessary. However, the offline computational efficiency of S^2GMR is not an issue since its learning speed is fast enough (not more than 1 s with 200 samples).

B. Application to Industrial Primary Reformer

The primary reformer, which is illustrated in Fig. 5 [32], comes from the hydrogen manufacturing units (HMUs) of an ammonia synthesis process (ASP). Desulphurized natural gas stream flows successively into the prereformer, primary reformer, and secondary reformer in order to obtain hydrogen (H_2). According to technological design, the primary reformer is the unit where the chemical reactions are mainly set off based on the following transformation equations with nickel catalyst:

$$\begin{aligned} C_nH_{2n+2} + nH_2O &\xleftrightarrow{\Delta} nCO + (2n+1)H_2 \\ CH_4 + H_2O &\xleftrightarrow{\Delta} CO + 3H_2 \\ CO + H_2O &\xleftrightarrow{\Delta} CO_2 + H_2. \end{aligned} \tag{41}$$

The main product of HMU, namely H_2, is a very important source material for the ASP, whose product is ammonia and is

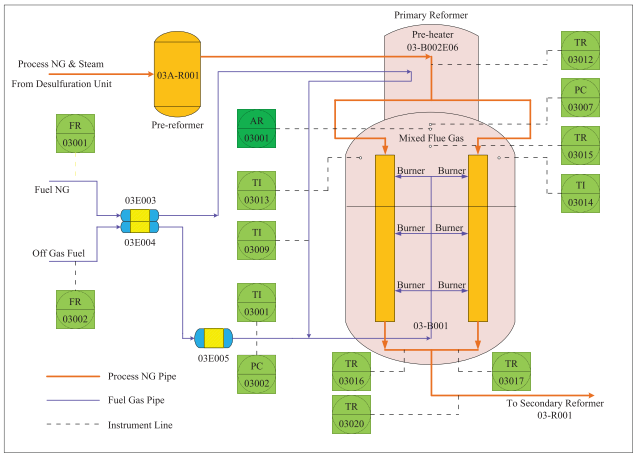


Fig. 5. Flowchart of the primary reformer.

TABLE V

DESCRIPTIONS OF SECONDARY VARIABLES FOR SOFT SENSING O_2 CONCENTRATION

Tags	Descriptions
FR03001.PV	Flow of fuel natural gas
FR03002.PV	Flow of fuel off gas
PC03002.PV	Pressure of fuel off gas
PC03007.PV	Pressure of furnace flue gas
TI03001.PV	Temperature of fuel off gas
TI03009.PV	Temperature of fuel natural gas
TR03012.PV	Temperature of process gas
TI03013.PV	Temperature of furnace flue gas
TI03014.PV	Temperature of furnace flue gas
TR03015.PV	Temperature of mixed furnace flue gas
TR03016.PV	Temperature of transformed gas
TR03017.PV	Temperature of transformed gas
TR03020.PV	Temperature of transformed gas

fed into the urea synthesis process to produce the final product (urea). Reaction temperature is a pivotal factor to keep transformation reactions in (41) carrying on stably. In an ammonia production plant, the reaction temperature should be controlled at around 580 °C by manipulating the burning condition of fuel gas in the dense burner. To this end, the technological design requires the concentration of oxygen (O_2) at the top of the primary reformer to be maintained within setting interval. In practice, O_2 concentration is measured by the expensive mass spectrometer (AR03001). Therefore, a soft sensor is desired for online estimating O_2 concentration for cutting down the production costs and calibrating the spectrometer. Secondary variables for soft sensing O_2 concentration are selected according to expert knowledge from field engineers, which are marked with light green rectangles in Fig. 5 and are explained in Table V.

Samples used for the soft-sensor development were collected from the database of distributed control systems in an industrial ASP, lasting from January 2015 to July 2015. After data preprocessing including outlier removal, missing value treatment, and time alignment, a total of 3480 samples have been obtained, which are evenly partitioned as the training data

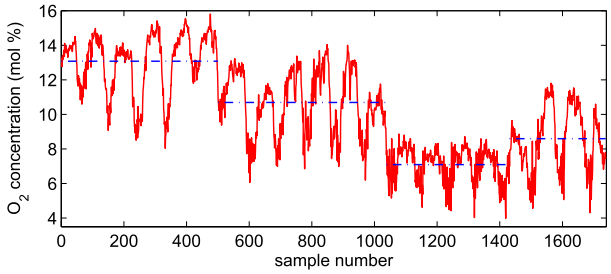


Fig. 6. Multimode characteristics of the primary reformer.

TABLE VI
AVERAGE PREDICTIVE RMSES OF VARIOUS SOFT SENSORS FOR THE PRIMARY REFORMER

LR	PLS	ELM	GMR	S^{2}GMR
10%	1.7418	1.5539	1.6103($K=2$)	1.4762($K=5$)
20%	1.7368	1.4954	1.5361($K=3$)	1.3513($K=9$)
30%	1.7377	1.3837	1.3732($K=5$)	1.2661($K=10$)
40%	1.7314	1.3516	1.3325($K=6$)	1.2362($K=11$)
50%	1.7294	1.3066	1.2593($K=7$)	1.1890($K=11$)

set and testing data set. To test the performance of PLS, GMR, ELM, and S^{2}GMR-based soft sensors with different amounts of labeled samples, we uniformly select a certain proportion of training samples and treat them as "unlabeled." Here, since the dimensionality of secondary variables is relatively high, the Bayesian regularization is employed for the S^{2}GMR, and for simplicity, the precision parameters are homogeneously set as 1's. In addition, the number of GCs is automatically determined using the BIC for soft sensors based on the GMR and S^{2}GMR.

The primary reformer is a process with multiple operating modes due to complex burning conditions as shown in Fig. 6, from which we can basically see the shifts of operating conditions marked with blue dot lines. Predictions by the PLS, GMR, ELM, and S^{2}GMR-based soft sensors for O_2 concentration with LR $= 50\%$ are presented in Fig. 7.

It is found from Fig. 7 that similar to the previous numerical example, the PLS-based soft sensor could not deal with any mode pertinently but tends to find a balance among the four operating conditions. In contrast, the other three soft sensors outperform the PLS-based one. However, the S^{2}GMR-based soft sensor intuitively demonstrates predictive advantages over those based on the ELM and GMR in tracking the dynamic changes of O_2 concentration in the first and fourth operating conditions. For further investigations, quantitative predictive accuracies of soft sensors based on various methods are compared in Table VI so are the selected numbers of GCs (i.e., K's) for the GMR and S^{2}GMR-based soft sensors. Collected predictive accuracies of four soft sensors indicate that they all get improved when the proportion of labeled samples raises from 10% to 50%, but improvements on the PLS-based one are slight; and the ELM-based soft sensor performs better with LR $>$ 30% than the GMR-based one, while the GMR-based soft sensor outperforms the ELM-based one when LR $<$ 30%. Moreover, the S^{2}GMR-based soft sensor always outperforms those based on the GMR and ELM.

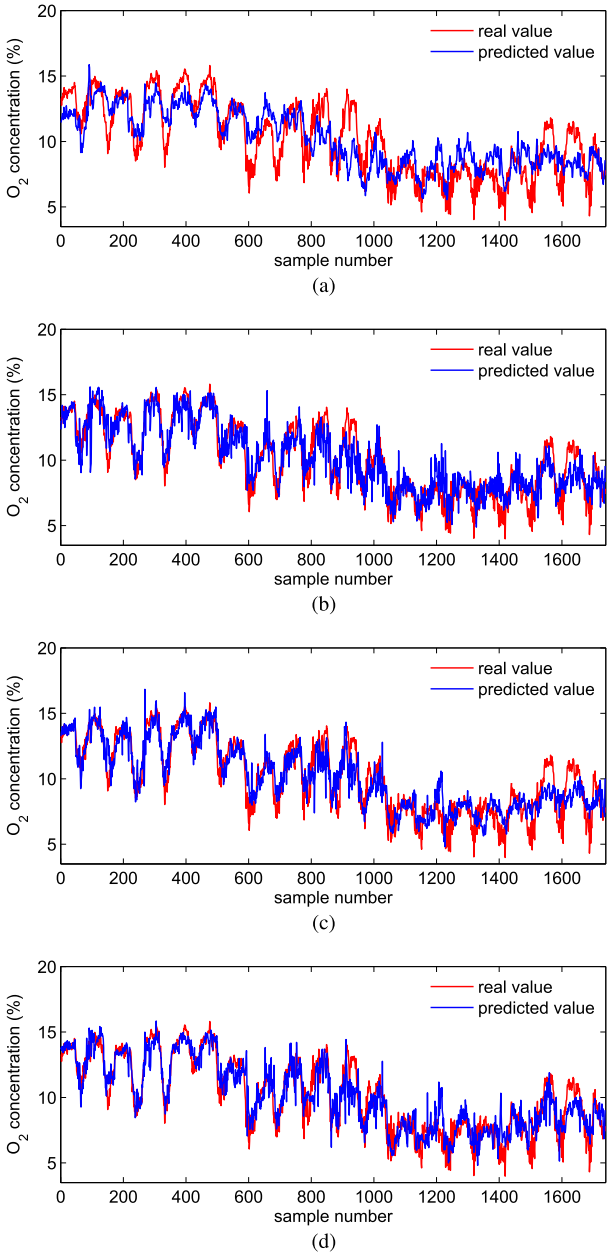
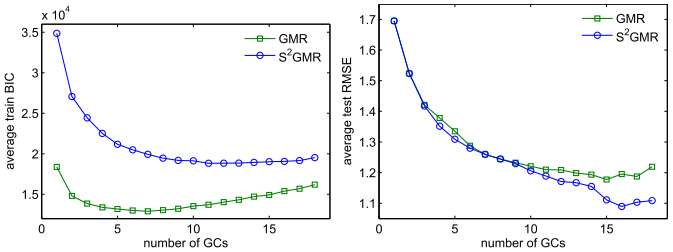


Fig. 7. Predictions of O_2 concentration obtained by soft sensors based on (a) PLS, (b) ELM, (c) GMR, and (d) S^{2}GMR.

It is interesting to note from Table VI that the GMR-based soft sensor tends to select fewer GCs compared with the S^{2}GMR-based one. Insufficient GCs results in underfitting, which as well as prevents high estimation accuracy. In fact, the underfitting phenomena do occur in the GMR-based soft sensor when performing model selection using the BIC, which can be seen from comparisons of the model selection process between the GMR and S^{2}GMR as presented in Figs. 8–12. Note that for some large values of K, the GMR and S^{2}GMR become disabled due to the singularity problem happened in calculating the inverse of covariance matrices.

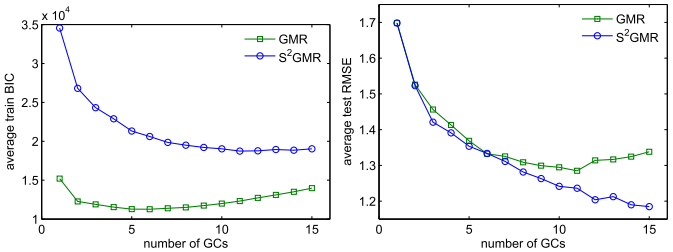
Comprehensively comparing Figs. 8–12, we have the following findings.

1) For the GMR-based soft sensor, the BIC always selects fewer GCs than the "optimal" one in terms



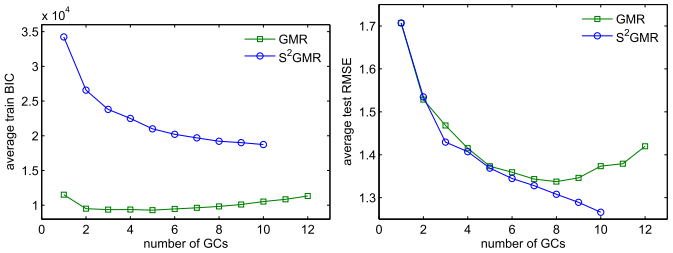
(a) (b)

Fig. 8. Model selection processes for soft sensors based on the GMR and S^{2}GMR with LR = 50%. (a) Dependence of train BIC on K. (b) Dependence of test RMSE on K.



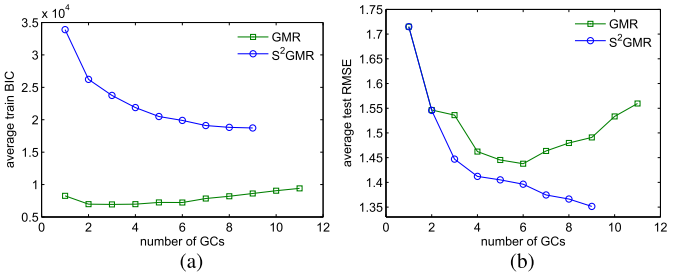
(a) (b)

Fig. 9. Model selection processes for soft sensors based on the GMR and S^{2}GMR with LR = 40%. (a) Dependence of train BIC on K. (b) Dependence of test RMSE on K.



(a) (b)

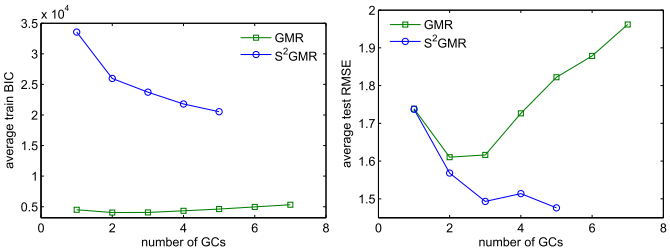
Fig. 10. Model selection processes for soft sensors based on the GMR and S^{2}GMR with LR = 30%. (a) Dependence of train BIC on K. (b) Dependence of test RMSE on K.



(a) (b)

Fig. 11. Model selection processes for soft sensors based on the GMR and S^{2}GMR with LR = 20%. (a) Dependence of train BIC on K. (b) Dependence of test RMSE on K.

of test RMSE, except when LR is set as 10%. In contrast, the BIC can basically perform model selection satisfactorily with available values of K for the S^{2}GMR-based soft sensor. This implies that with merely labeled samples, the BIC's penalty on model complexity is too heavy; however, with both labeled and unlabeled samples, the BIC's penalty becomes more proper.



(a) (b)

Fig. 12. Model selection processes for soft sensors based on the GMR and S^{2}GMR with LR = 10%. (a) Dependence of train BIC on K. (b) Dependence of test RMSE on K.

TABLE VII
AVERAGE CPT (IN SECONDS) CONSUMED BY VARIOUS SOFT SENSORS FOR THE PRIMARY REFORMER

LR	CPT_{trn}				CPT_{tst}			
	PLS	ELM	GMR	S^{2}GMR	PLS	ELM	GMR	S^{2}GMR
10%	0.12	0.02	0.14	0.44	0.08	0.06	0.04	0.28
20%	0.12	0.04	0.22	1.45	0.08	0.07	0.04	0.45
30%	0.11	0.06	0.40	2.03	0.07	0.08	0.05	0.47
40%	0.12	0.08	0.62	2.43	0.08	0.08	0.05	0.54
50%	0.11	0.11	0.85	3.36	0.08	0.09	0.05	0.55

2) Excluding the negative influence of underfitting on the GMR-based soft sensor, comparisons of test RMSE with the same K values reveal that the S^{2}GMR-based soft sensor outperforms the GMR-based one, and the superiority of S^{2}GMR becomes more significant as LR decreases and the number of GCs increases.
3) For the GMR-based soft sensor, when the number of GCs exceeds a certain value, test RMSE increases (particularly when labeled samples become fewer), which indicates the occurrence of overfitting. By contrast, through incorporating unlabeled samples, the overfitting problem is effectively alleviated by the S^{2}GMR-based soft sensor.
4) The number of available K values for the S^{2}GMR-based soft sensor seems fewer than that for the GMR-based one. This may be explained as follows. In the S^{2}GMR, separately manipulating the regression coefficients with (17) or (21) makes the secondary and primary variables more correlated. As a result, the way of calculating the covariance matrices of joint PDFs with (7) makes covariance matrices more prone to be singular.

Since the BIC makes the GMR underfitting in this case study, we have tried the AIC [23] for model selection. However, the AIC kept decreasing but never increased for the GMR-based soft sensor with all available K values and LR ranged from 50% to 10%, indicating that overfitting happened. In contrast, the variation trends of AIC for the S^{2}GMR are almost identical to those of BIC, which could still achieve satisfactory model selection for the S^{2}GMR-based soft sensor.

The CPT time consumed by various soft sensors are summarized in Table VII, from which we can see that although the two indices CPT_{trn} and CPT_{tst} for the S^{2}GMR-based soft sensor are larger than those for the other three soft sensors, the computational efficiency for S^{2}GMR-based soft sensor is still acceptable.

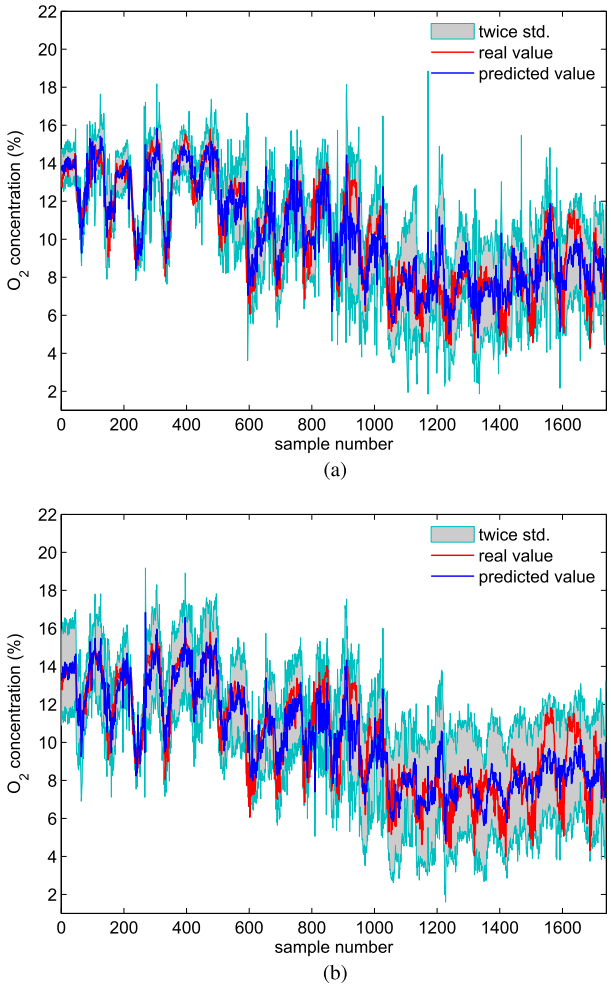


Fig. 13. Confidence intervals for O_2 concentration with $\pm$ twice standard deviations. (a) Confidence intervals provided by the S^{2}GMR. (b) Confidence intervals provided by the GMR.

In addition to the estimations of O_2 concentration, both the GMR and S^{2}GMR-based soft sensors can give the corresponding confidence intervals, which are shown in Fig. 13. As can be seen, the true values of O_2 concentrations are basically wrapped by confidence intervals provided by both the GMR and S^{2}GMR. However, one can find that confidence intervals provided by the S^{2}GMR are much narrower than those provided by the GMR. In specific, for the S^{2}GMR, the average distance between the upper and lower bounds of confidence intervals is 3.94, while that for the GMR is 5.02, which demonstrates the S^{2}GMR has a better performance assessment ability.

V. Conclusion

In this paper, in order to deal with the deficiency of GMR in soft-sensor applications, we have proposed an S^2 GMR including the novel semisupervised model structure and EM-based parameter learning algorithm, which could exploit the useful information contained in both labeled and unlabeled samples. Meanwhile, model selection for the S^{2}GMR can be performed automatically with the BIC. The performance of S^{2}GMR has been thoroughly evaluated based on synthetic and real-life industrial data sets, and the following three conclusions can be basically drawn.

1) The S^{2}GMR could provide more accurate estimations of target variable compared with the GMR, in particular when labeled samples are rare.
2) The S^{2}GMR can facilitate model selection by alleviating the problem of underfitting or overfitting.
3) The S^{2}GMR has better assessment ability from the perspective of modeling prediction uncertainties compared with the GMR.

However, model selection using BIC requires repeated operations for traversing all candidate numbers of GC. Besides, the precision parameters in the Bayesian regularization need to be manually predefined. Such drawbacks of BIC and S^{2}GMR can be overcome under the variational inference framework, which is able to automatically determine the number of GCs and heterogeneous precision parameters within one training round. Therefore, a variational S^{2}GMR is desirable and is expected to gain better performance, which will be our upcoming work.

Appendix

Detailed derivations for the complete data log-likelihood function in (12) and parameter updating formulas in (15)–(18) are presented as follows.

With the sum rule and product rule of probability and the independence assumption, $\mathcal{L}(\boldsymbol{\Theta})$ can rewritten as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\Theta}) &= \sum_{\boldsymbol{Z}_l,\boldsymbol{Z}_u} p(\boldsymbol{Z}_l,\boldsymbol{Z}_u|\mathcal{D}) \ln p(\mathcal{D},\boldsymbol{Z}_l,\boldsymbol{Z}_u) \\ &= \sum_{\boldsymbol{Z}_l} \widetilde{p}(\boldsymbol{Z}_l) \ln p(\boldsymbol{X}_l,\boldsymbol{Y}_l,\boldsymbol{Z}_l) + \sum_{\boldsymbol{Z}_u} \widetilde{p}(\boldsymbol{Z}_u) \ln p(\boldsymbol{X}_u,\boldsymbol{Z}_u) \\ &= \sum_{\boldsymbol{Z}_l} \widetilde{p}(\boldsymbol{Z}_l) \ln p(\boldsymbol{Y}_l|\boldsymbol{X}_l,\boldsymbol{Z}_l) + \sum_{\boldsymbol{Z}_l} \widetilde{p}(\boldsymbol{Z}_l) \ln p(\boldsymbol{X}_l|\boldsymbol{Z}_l) \\ &\quad + \sum_{\boldsymbol{Z}_l} \widetilde{p}(\boldsymbol{Z}_l) \ln p(\boldsymbol{Z}_l) + \sum_{\boldsymbol{Z}_u} \widetilde{p}(\boldsymbol{Z}_u) \ln p(\boldsymbol{X}_u|\boldsymbol{Z}_u) \\ &\quad + \sum_{\boldsymbol{Z}_u} \widetilde{p}(\boldsymbol{Z}_u) \ln p(\boldsymbol{Z}_u) \end{aligned} \tag{A.1}$$

where $\mathcal{D} = (\boldsymbol{X}_l,\boldsymbol{Y}_l,\boldsymbol{X}_u)$, $\widetilde{p}(\boldsymbol{Z}_l) = p(\boldsymbol{Z}_l|\boldsymbol{X}_l,\boldsymbol{Y}_l)$, $\widetilde{p}(\boldsymbol{Z}_u) = p(\boldsymbol{Z}_u|\boldsymbol{X}_u)$. Then, $\sum_{\boldsymbol{Z}_l} \widetilde{p}(\boldsymbol{Z}_l) \ln p(\boldsymbol{X}_l,\boldsymbol{Y}_l,\boldsymbol{Z}_l)$ can be expanded as

$$\begin{aligned}&\sum_{\boldsymbol{Z}_l} \widetilde{p}(\boldsymbol{Z}_l) \ln p(\boldsymbol{X}_l,\boldsymbol{Y}_l,\boldsymbol{Z}_l) \\ &= \sum_{z_1,\ldots,z_{n_l}} \left(\prod_{i=1}^{n_l} p(z_i|\boldsymbol{x}_i,y_i) \ln \prod_{i=1}^{n_l} p(y_i|\boldsymbol{x}_i,z_i) \right) \\ &= \sum_{z_1,\ldots,z_{n_l}} \left(\prod_{i=1}^{n_l} p(z_i|\boldsymbol{x}_i,y_i) \sum_{i=1}^{n_l} \ln p(y_i|\boldsymbol{x}_i,z_i) \right) \\ &= \sum_{i=1}^{n_l} \sum_{k=1}^{K} p(z_i=k|\boldsymbol{x}_i,y_i) \ln p(y_i|\boldsymbol{x}_i,z_i=k) \\ &= \sum_{i=1}^{n_l} \sum_{k=1}^{K} \gamma_k^i \ln p_k(y_i|\boldsymbol{x}_i) \end{aligned} \tag{A.2}$$

where we have used the facts that $\sum_{z_i} p(\boldsymbol{z}_i|\boldsymbol{x}_i, y_i) = 1$ for $i = 1, \ldots, n_l$. Similarly, using $\sum_{z_j} p(\boldsymbol{z}_j|\boldsymbol{x}_j) = 1$ for $j = n_l + 1, \ldots, n_l + n_u$, we can obtain that

$$\sum_{\boldsymbol{Z}_l} \widetilde{p}(\boldsymbol{Z}_l) \ln p(\boldsymbol{X}_l|\boldsymbol{Z}_l) = \sum_{i=1}^{n_l}\sum_{k=1}^{K} \gamma_k^i \ln p_k(\boldsymbol{x}_i) \tag{A.3}$$

$$\sum_{\boldsymbol{Z}_l} \widetilde{p}(\boldsymbol{Z}_l) \ln p(\boldsymbol{Z}_l) = \sum_{i=1}^{n_l}\sum_{k=1}^{K} \gamma_k^i \ln \alpha_k \tag{A.4}$$

$$\sum_{\boldsymbol{Z}_u} \widetilde{p}(\boldsymbol{Z}_u) \ln p(\boldsymbol{X}_u|\boldsymbol{Z}_u) = \sum_{j=n_l+1}^{n_l+n_u}\sum_{k=1}^{K} \gamma_k^j \ln p_k(\boldsymbol{x}_j) \tag{A.5}$$

$$\sum_{\boldsymbol{Z}_u} \widetilde{p}(\boldsymbol{Z}_u) \ln p(\boldsymbol{Z}_u) = \sum_{j=n_l+1}^{n_l+n_u}\sum_{k=1}^{K} \gamma_k^j \ln \alpha_k. \tag{A.6}$$

Substituting (A.2)–(A.6) into (A.1) leads to (12).

For learning models parameters $\{\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}, \boldsymbol{\omega}_k, \psi_k, \sigma_k^2\}_{k=1}^K$, (12) is simplified as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\Theta}) = &\sum_{i=1}^{n_l}\sum_{k=1}^{K} \gamma_k^i \ln \mathcal{N}\big(y_i\big|\widetilde{\boldsymbol{x}}_i^T\widetilde{\boldsymbol{\omega}}_k, \sigma_k^2\big) \\ &+ \sum_{i=1}^{n_l}\sum_{k=1}^{K} \gamma_k^i \ln \mathcal{N}\big(\boldsymbol{x}_i\big|\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\big) \\ &+ \sum_{j=n_l+1}^{n_l+n_u}\sum_{k=1}^{K} \gamma_k^j \ln \mathcal{N}\big(\boldsymbol{x}_j\big|\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}\big) + \mathbb{C}\end{aligned} \tag{A.7}$$

where those terms independent of $\{\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}, \boldsymbol{\omega}_k, \psi_k, \sigma_k^2\}_{k=1}^K$ are absorbed into the constant term $\mathbb{C}$.

Subsequently, the derivatives of $\mathcal{L}(\boldsymbol{\Theta})$ with respect to $\{\boldsymbol{\mu}_k, (\boldsymbol{\Sigma}_k)^{-1}, \widetilde{\boldsymbol{\omega}}_k, (\sigma_k^2)^{-1}\}_{k=1}^K$ can be obtained as

$$\frac{\partial \mathcal{L}(\boldsymbol{\Theta})}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{n_l} \gamma_k^i (\boldsymbol{\Sigma}_k)^{-1}\overline{\boldsymbol{x}}_k^i + \sum_{j=n_l+1}^{n_l+n_u} \gamma_k^j (\boldsymbol{\Sigma}_k)^{-1}\overline{\boldsymbol{x}}_k^j$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\Theta})}{\partial (\boldsymbol{\Sigma}_k)^{-1}} = \frac{1}{2}\sum_{i=1}^{n_l} \gamma_k^i \big(\boldsymbol{\Sigma}_k - \overline{\boldsymbol{x}}_k^i\big(\overline{\boldsymbol{x}}_k^i\big)^T\big) \tag{A.8}$$

$$+\frac{1}{2}\sum_{j=n_l+1}^{n_l+n_u} \gamma_k^j \big(\boldsymbol{\Sigma}_k - \overline{\boldsymbol{x}}_k^j\big(\overline{\boldsymbol{x}}_k^j\big)^T\big) \tag{A.9}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\Theta})}{\partial \widetilde{\boldsymbol{\omega}}_k} = \frac{1}{\sigma_k^2}\widetilde{\boldsymbol{X}}_l^T \boldsymbol{\Gamma}_k(\boldsymbol{Y}_l - \widetilde{\boldsymbol{X}}_l\widetilde{\boldsymbol{\omega}}_k) \tag{A.10}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\Theta})}{\partial (\sigma_k^2)^{-1}} = \frac{1}{2}\sum_{i=1}^{n_l} \gamma_k^i \big(\sigma_k^2 - \big(y_i - \widetilde{\boldsymbol{x}}_i^T\widetilde{\boldsymbol{\omega}}_k\big)^2\big). \tag{A.11}$$

Setting (A.8)–(A.11) to zeros leads to (15)–(18).

REFERENCES

[1] J. Yu, "Online quality prediction of nonlinear and non-Gaussian chemical processes with shifting dynamics using finite mixture model based Gaussian process regression approach," *Chem. Eng. Sci.*, vol. 82, pp. 22–30, Sep. 2012.

[2] W. Shao and X. Tian, "Adaptive soft sensor for quality prediction of chemical processes based on selective ensemble of local partial least squares models," *Chem. Eng. Res. Des.*, vol. 95, pp. 113–132, 2015.

[3] D. Wang, J. Liu, and R. Srinivasan, "Data-driven soft sensor approach for quality prediction in a refining process," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 11–17, Feb. 2010.

[4] M. Kano and M. Ogawa, "The state of the art in chemical process control in Japan: Good practice and questionnaire survey," *J. Process Control*, vol. 20, no. 9, pp. 969–982, Oct. 2010.

[5] F. Souza and R. Araújo, "Online mixture of univariate linear regression models for adaptive soft sensors," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 937–945, May 2014.

[6] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven soft sensors in the process industry," *Comput. Chem. Eng.*, vol. 33, no. 4, pp. 795–814, Apr. 2009.

[7] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data mining and analytics in the process industry: The role of machine learning," *IEEE Access*, vol. 5, pp. 20590–20616, 2017.

[8] F. A. A. Souza and R. Araújo, "Mixture of partial least squares experts and application in prediction settings with multiple operating modes," *Chemometrics Intell. Lab. Syst.*, vol. 130, no. 2, pp. 192–202, 2014.

[9] L. Zhou, J. Chen, Z. Song, Z. Ge, and A. Miao, "Probabilistic latent variable regression model for process-quality monitoring," *Chem. Eng. Sci.*, vol. 116, pp. 296–305, Sep. 2014.

[10] X. Yuan, Z. Ge, B. Huang, and Z. Song, "A probabilistic just-in-time learning framework for soft sensor development with missing data," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 1124–1132, May 2017.

[11] C. Shang, B. Huang, F. Yang, and D. Huang, "Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling," *AIChE J.*, vol. 61, no. 12, pp. 4126–4139, 2015.

[12] L. Yao and Z. Ge, "Locally weighted prediction methods for latent factor analysis with supervised and semisupervised process data," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 126–138, Jan. 2017.

[13] J. Yu, "Multiway Gaussian mixture model based adaptive kernel partial least squares regression method for soft sensor estimation and reliable quality prediction of nonlinear multiphase batch processes," *Ind. Eng. Chem. Res.*, vol. 51, no. 40, pp. 13227–13237, 2012.

[14] R. Grbić, D. Slišković, and P. Kadlec, "Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models," *Comput. Chem. Eng.*, vol. 58, no. 22, pp. 84–98, 2013.

[15] W. Xiong, W. Zhang, B. Xu, and B. Huang, "JITL based MWGPR soft sensor for multi-mode process with dual-updating strategy," *Comput. Chem. Eng.*, vol. 90, pp. 260–267, Jul. 2016.

[16] L. L. T. Chan and J. Chen, "Melt index prediction with a mixture of Gaussian process regression with embedded clustering and variable selections," *J. Appl. Polym. Sci.*, vol. 134, no. 40, p. 45237, 2017.

[17] L. Wang, H. Jin, X. Chen, J. Dai, K. Yang, and D. Zhang, "Soft sensor development based on the hierarchical ensemble of Gaussian process regression models for nonlinear and non-Gaussian chemical processes," *Ind. Eng. Chem. Res.*, vol. 55, no. 8, pp. 7704–7719, 2016.

[18] X. Yuan, Z. Ge, and Z. Song, "Soft sensor model development in multiphase/multimode processes based on Gaussian mixture regression," *Chemometrics Intell. Lab. Syst.*, vol. 138, no. 6, pp. 97–109, 2014.

[19] C. Mei, Y. Su, G. Liu, Y. Ding, and Z. Liao, "Dynamic soft sensor development based on Gaussian mixture regression for fermentation processes," *Chin. J. Chem. Eng.*, vol. 25, no. 1, pp. 116–122, 2017.

[20] J. Zhu, Z. Ge, and Z. Song, "Variational Bayesian Gaussian mixture regression for soft sensing key variables in non-Gaussian industrial processes," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 1092–1099, May 2017.

[21] Z. Ge, F. Gao, and Z. Song, "Mixture probabilistic PCR model for soft sensing of multimode processes," *Chem. Intell. Lab. Syst.*, vol. 105, no. 1, pp. 91–105, Jan. 2011.

[22] W. Shao and X. Tian, "Semi-supervised selective ensemble learning based on distance to model for nonlinear soft sensor development," *Neurocomputing*, vol. 222, pp. 91–104, Jan. 2017.

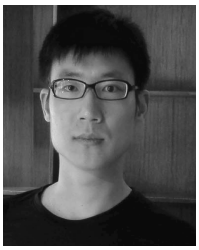
[23] H.-C. Yan, J.-H. Zhou, and C. K. Pang, "Gaussian mixture model using semisupervised learning for probabilistic fault diagnosis under new data categories," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 4, pp. 723–733, Apr. 2017.

[24] X. Xing, Y. Yu, H. Jiang, and S. Du, "A multi-manifold semi-supervised Gaussian mixture model for pattern classification," *Pattern Recognit. Lett.*, vol. 34, no. 16, pp. 2118–2125, 2013.

[25] H. Sung, "Gaussian mixture regression and classification," Ph.D. dissertation, Dept. Statist., Rice Univ., Houston, TX, USA, 2004.

[26] K. P. Burnham and D. R. Anderson, "Multimodel inference: Understanding AIC and BIC in model selection," *Sociol. Methods Res.*, vol. 33, no. 3, pp. 261–304, 2004.

[27] J. Liu, D.-S. Chen, and J.-F. Shen, "Development of self-validating soft sensors using fast moving window partial least squares," *Ind. Eng. Chem. Res.*, vol. 49, no. 22, pp. 11530–11546, 2010.
[28] H. Kaneko, M. Arakawa, and K. Funatsu, "Applicability domains and accuracy of prediction of soft sensor models," *AIChE J.*, vol. 57, no. 6, pp. 1506–1513, 2011.
[29] C. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. New York, NY, USA: Springer-Verlag, 2006.
[30] R. Lindgren, P. Geladi, and S. Wold, "The kernel algorithm for PLS," *J. Chemometrics*, vol. 7, no. 1, pp. 45–59, 1993.
[31] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.
[32] L. Yao and Z. Ge, "Moving window adaptive soft sensor for state shifting process based on weighted supervised latent factor analysis," *Control Eng. Pract.*, vol. 61, pp. 72–80, Apr. 2017.



Weiming Shao received the B.Eng. and Ph.D. degrees from the College of Information and Control Engineering, China University of Petroleum, Qingdao, China, in 2009 and 2016, respectively.

From 2014 to 2015, he was a Visiting Research Associate with the Department of Electrical Engineering, Petroleum Institute, Abu Dhabi, United Arab Emirates. He is currently a Post-Doctoral Research Fellow with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, China. His current research interests include machine learning and statistical learning methods and their applications to semisupervised, robust, and adaptive soft-sensor development.

Zhiqiang Ge (M'13–SM'17) received the B.Eng. and Ph.D. degrees from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively.

From 2010 to 2011, he was a Research Associate with the Department of Chemical and Biomolecular Engineering, Hong Kong University of Science Technology, Hong Kong. In 2013, he was a Visiting Professor with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada. From 2014 to 2017, he was an Alexander von Humboldt Research Fellow with the University of Duisburg–Essen, Duisburg, Germany. He is currently a Full Professor with the College of Control Science and Engineering, Zhejiang University. His current research interests include industrial big data, process monitoring, quality prediction, machine leaning, and Bayesian methods.

Zhihuan Song received the B.Eng. and M.Eng. degrees in industrial automation from the Hefei University of Technology, Hefei, China, in 1983 and 1986, respectively, and the Ph.D. degree in industrial automation from Zhejiang University, Hangzhou, China, in 1997.

Since 1997, he has been with the Department of Control Science and Engineering, Zhejiang University, where he was a Post-Doctoral Research Fellow, an Associate Professor, and currently a Professor. He has authored or co-authored over 200 papers in journals and conference proceedings. His current research interests include the modeling and fault diagnosis of industrial processes, analytics and applications of industrial big data, and advanced process control technologies.