Journal of Process Control 64 (2018) 123–131

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Semisupervised learning for probabilistic partial least squares regression model and soft sensor application

Junhua Zheng, Zhihuan Song*

State Key Laboratory of Industrial Control Technology, Institute of Industrial Process Control, College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China

ARTICLE INFO

Article history:
Received 5 January 2017
Received in revised form 21 January 2018
Accepted 21 January 2018

Keywords:
Probabilistic partial least squares
Regression modeling
Expectation-maximization
Semisupervised data modeling

ABSTRACT

Due to long sampling time and large measurement delay, variables such as melt index, concentrations of key components in the stream, and product quality variables are difficult to measure online. At the same time, routinely recorded variables such as flow, temperature and press are much easier to measure. As a result, only a small portion of data has values for all variables, while other large parts of data only have values for those routinely recorded variables. Focused on regression modeling between those two types of process variables with imbalanced sampling values, this paper develops a semisupervised form of the Probabilistic Partial Least Squares (PPLS) model. In this model, both labeled data samples (with values for both two types of variables) and unlabeled data samples (with values only for routinely recorded variables) can be effectively used. For parameter learning of the semisupervised PPLS model, an efficient Expectation-Maximization algorithm is designed. An industrial case study is provided as an example for soft sensor application, which is constructed based on the new developed model.

1. Introduction

In the past years, with the wide use of the distributed control systems in modern industrial processes, a large amount of data have been collected, which motivates use of various data-based methods for modeling, monitoring, and control. By mining and analyzing the patterns and relationships among process data, useful information can be extracted, based on which statistical models can be developed [1–4]. Those data models can be used for various applications, such as dimensionality reduction, data visualization, process monitoring, fault diagnosis, and soft sensing/quality prediction [5–8].

Compared to the routinely recorded process variables such as temperature, pressure and flow, some key performance indices and quality variables are much more difficult to be monitored and measured online, such as the melt index in the polypropylene production process, the viscosity index in the rubber production process, or component concentrations in the product stream. Typically, these variables are often obtained through expensive analyzers or lab analyses, both of which may introduce a significant time delay to the quality control system. In contrast, by building a regression model between some easy-to-measure process variables and those key indices, data-based soft sensors can provide continuous estimations for those important variables, which have become more and more popular in recent years.

To date, various data-based soft sensor modeling methods have been developed, including latent variable models such as principal component regression (PCR) and partial least squares (PLS) [9–13], artificial neural networks (ANN) and kernel-based models [14–17], and probabilistic and Bayesian methods [18]. Practically, the probabilistic soft sensor modeling method provides a natural mechanism for describing relationships among stochastic process variables, with considerations of both measurement and model uncertainties. Compared to deterministic modeling approaches, several additional advantages can be found by using the probabilistic modeling method [19]. First, most probabilistic models are based on Bayes' rule and can be trained through the expectation maximization (EM) learning mechanism. While Bayes' rule provides as the cornerstone for probabilistic inference, the EM algorithm provides an effective learning framework for most probabilistic models. Second, the problem of missing data and outliers which are quite common in practice can be solved straightforwardly under the probabilistic modeling framework. Third, the flexible probabilistic subspace can be easily generalized to mixture models,

* Corresponding author.
E-mail addresses: jhzheng@zju.edu.cn (J. Zheng), songzhihuan@zju.edu.cn (Z. Song).

https://doi.org/10.1016/j.jprocont.2018.01.008
0959-1524/© 2018 Elsevier Ltd. All rights reserved.

which can be used to deal with more complicated process data modeling problems. For the soft sensing purpose in industrial processes, several probabilistic data models have already been introduced, including probabilistic PCR [20], supervised latent factor analysis [21], relevant vector machine [22], Gaussian process regression [23], and so on [24–27]. More recently, the widely used PLS model has also been extended to the probabilistic form for soft sensing [28].

For a typical soft sensor model development, a fully labeled training dataset is needed, which means each sample in the training dataset should has input and output measurements for the soft sensor. However, while the routinely recorded input data such as temperature, pressure, and flowrate are easy to be measured and obtained, the output data of the soft sensor which correspond to key performance indices or product quality variables are usually difficult to obtain. As a result, we may only have a small number of labeled data samples for soft sensor modeling, and hold a large number of unlabeled data samples which lack of measurements for key process variables. This is actually a semisupervised learning problem from the viewpoint of machine learning [29]. Although the unlabeled dataset has no output values, it may contain important process information, based on which the estimation of the distribution of input variables could be significantly improved. In the past years, several semisupervised learning methods have already been introduced for process monitoring and soft sensor applications [30–34].

Under the probabilistic PLS model structure, the motivation of the present paper is to incorporate both labeled and unlabeled datasets for soft sensor modeling. In contrast to the basic probabilistic PLS model, the new model which incorporates both labeled and unlabeled datasets is termed as semisupervised probabilistic PLS model. For parameter learning of the semisupervised PPLS model, an efficient Expectation-Maximization algorithm is designed. Unlike the basic PPLS model, the new method has two modeling items for both labeled and unlabeled datasets, which is actually a combination of unsupervised probabilistic model and supervised probabilistic model. The model structure and parameter learning process are similar to those of the probabilistic PLS model.

2. Probabilistic PLS model (PPLS)

The main idea of the probabilistic PLS model is to use a part of latent variables of **x** to explain **y**, and keep the rest of the latent variables to explain its own information. Here is the generative model structure of probabilistic PLS [28]

$$\mathbf{x} = \boldsymbol{\mu}_x + \mathbf{P}\mathbf{t}^s + \mathbf{Q}\mathbf{t}^b + \mathbf{e}_x \tag{1}$$

$$\mathbf{y} = \boldsymbol{\mu}_y + \mathbf{C}\mathbf{t}^s + \mathbf{e}_y \tag{2}$$

where $\mathbf{P} \in R^{m\times q_s}$, $\mathbf{C} \in R^{r\times q_s}$ and $\mathbf{Q} \in R^{m\times q_b}$ are loading matrices, $\mathbf{t}^s \in R^{q_s\times 1}$ is the latent variable vector that used to explain the information of **y**, $\mathbf{t}^b \in R^{q_b\times 1}$ is the rest of the latent variable vector that used to explain **x**, $\boldsymbol{\mu}_x$and $\boldsymbol{\mu}_y$ are mean vectors of **x** and **y**, $\mathbf{e}_x \in R^{m\times 1}$ and $\mathbf{e}_y \in R^{r\times 1}$ are measurement noise of **x** and **y**.

In this model, it is assumed that both probability density functions of the latent variable and the measurement noise are Gaussian, thus $p(\mathbf{t}_s) = N(0, \mathbf{I})$, $p(\mathbf{t}_b) = N(0, \mathbf{I})$, $p(\mathbf{e}_x) = N(0, \Sigma_x)$, and $p(\mathbf{e}_y) = N(0, \Sigma_y)$. Here, heterogeneous noise variances have been assumed for both **x** and **y**, thus $\Sigma_x = diag\{\sigma_{x,u}^2\}_{u=1,2,\cdots,m}$ and $\Sigma_y = diag\{\sigma_{y,v}^2\}_{v=1,2,\cdots,r}$. Given datasets $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]^T \in R^{n\times m}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]^T \in R^{n\times r}$, the parameter set of the probabilistic PLS model $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_x, \boldsymbol{\mu}_y, \mathbf{P}, \mathbf{Q}, \mathbf{C}, \Sigma_x, \Sigma_y\}$ can be determined by maximizing the following log-likelihood function

$$L(\mathbf{X}, \mathbf{Y}|\boldsymbol{\mu}_x, \boldsymbol{\mu}_y, \mathbf{P}, \mathbf{Q}, \mathbf{C}, \Sigma_x, \Sigma_y) = \ln \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{y}_i|\boldsymbol{\mu}_x, \boldsymbol{\mu}_y, \mathbf{P}, \mathbf{Q}, \mathbf{C}, \Sigma_x, \Sigma_y) \tag{3}$$

which can be efficiently handled through the Expectation-Maximization algorithm. It is a fast numerical calculation method. Instead of maximizing the log-likelihood function directly, the EM algorithm tries to maximize the expected complete-data log-likelihood function, including the observed variables **x**, **y** and the latent variables $\mathbf{t}^s$, $\mathbf{t}^b$. The value of expected complete-data log-likelihood function of the dataset with respect to the latent variables can be calculated as follows [28]

$$\begin{aligned} E[L(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Theta})] &= \sum_{i=1}^{n} \int p(\begin{bmatrix} \boldsymbol{t}_i^s \\ \boldsymbol{t}_i^b \end{bmatrix} | \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix}, \boldsymbol{\Theta}_{old}) \ln[p(\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix}, \begin{bmatrix} \boldsymbol{t}_i^s \\ \boldsymbol{t}_i^b \end{bmatrix} |\boldsymbol{\Theta})]d \begin{bmatrix} \boldsymbol{t}_i^s \\ \boldsymbol{t}_i^b \end{bmatrix} \\ &= \sum_{i=1}^{n} \int p(\begin{bmatrix} \boldsymbol{t}_i^s \\ \boldsymbol{t}_i^b \end{bmatrix} | \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix}, \boldsymbol{\Theta}_{old}) \ln[p(\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix} | \begin{bmatrix} \boldsymbol{t}_i^s \\ \boldsymbol{t}_i^b \end{bmatrix}, \boldsymbol{\Theta}) p(\begin{bmatrix} \boldsymbol{t}_i^s \\ \boldsymbol{t}_i^b \end{bmatrix} |\boldsymbol{\Theta}]d \begin{bmatrix} \boldsymbol{t}_i^s \\ \boldsymbol{t}_i^b \end{bmatrix} \end{aligned} \tag{4}$$

Based on the formulation of the expected complete-data log-likelihood function, the EM algorithm can be iteratively carried out through two main steps: the Expectation-step (E-step) and the Maximization-step (M-step). It has been proved that the EM algorithm can guarantee the log likelihood value never decreases when this algorithm is carried out iteratively [35]. As a result, the optimal value of the parameter set $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_x, \boldsymbol{\mu}_y, \mathbf{P}, \mathbf{Q}, \mathbf{C}, \Sigma_x, \Sigma_y\}$ can be determined as soon as the EM algorithm converges. The main drawback of the EM algorithm is that it may get a local optimal value, for improvement, several different initialized values are suggested to train the model. Another limitation of this method is due to the Gaussian assumption of the latent variable and measurement noise. However, if the non-Gaussian distribution is assumed, the probabilistic PLS model structure cannot be determined, since the distribution forms of non-Gaussian variables are quite different from each other. Besides, the model training will become much more difficult, which is out the scope of the current work.

3. Semisupervised modeling of PLS (SSPPLS)

3.1. Model development

Similar to the basic probabilistic PLS model, the semisupervised probabilistic PLS model also deals with a pair of datasets $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n]^T \in R^{n\times m}$ and $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\cdots,\mathbf{y}_{n_1}]^T \in R^{n_1\times r}$, where m is the number of measured variables of $\mathbf{X}$, r is the number of measured variables of $\mathbf{Y}$. However, unlike the PPLS model, the number of measurements in $\mathbf{X}$ in the semisupervised PPLS model is much larger than that in the $\mathbf{Y}$ dataset, thus $n>n_1$. In other words, only a part of the measurements in $\mathbf{X}$ have their corresponding values in $\mathbf{Y}$. Therefore, the original two datasets $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n]^T \in R^{n\times m}$ and $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\cdots,\mathbf{y}_{n_1}]^T \in R^{n_1\times r}$ can be written as one labeled dataset $\{\mathbf{X}_1 \in R^{n_1\times m}, \mathbf{Y} \in R^{n_1\times r}\}$ and one unlabeled dataset $\mathbf{X}_2 \in R^{n_2\times r}$, where $n_1+n_2=n$.

The model structure of the semisupervised PPLS is similar to that of the basic PPLS model, which is given as follows

$$\mathbf{x}_i = \boldsymbol{\mu}_x + \mathbf{P}\mathbf{t}_i^s + \mathbf{Q}\mathbf{t}_i^b + \mathbf{e}_{x,i} \tag{5}$$

$$\mathbf{y}_j = \boldsymbol{\mu}_y + \mathbf{C}\mathbf{t}_j^s + \mathbf{e}_{y,j} \tag{6}$$

where $\mathbf{P} \in R^{m\times q_s}$, $\mathbf{C} \in R^{r\times q_s}$ and $\mathbf{Q} \in R^{m\times q_b}$ are loading matrices, $\mathbf{t}^s \in R^{q_s\times 1}$and $\mathbf{t}^b \in R^{q_b\times 1}$ are two latent variable vectors, $\boldsymbol{\mu}_x$and $\boldsymbol{\mu}_y$ are mean vectors, and $\mathbf{e}_{x,i} \in R^{m\times 1}$ and $\mathbf{e}_{y,j} \in R^{r\times 1}$ are measurement noise of $\mathbf{x}_i$ and $\mathbf{y}_i$. Similarly, it is also assumed in the semisupervised model that both of the latent variables and the measurement noise follow Gaussian distribution. thus $p(\mathbf{t}_s)=N(0,\mathbf{I})$, $p(\mathbf{t}_b)=N(0,\mathbf{I})$, $p(\mathbf{e}_x)=N(0,\Sigma_x)$, and $p(\mathbf{e}_y)=N(0,\Sigma_y)$, where $\Sigma_x = diag\{\sigma_{x,u}^2\}_{u=1,2,\cdots,m}$ and $\Sigma_y = diag\{\sigma_{y,v}^2\}_{v=1,2,\cdots,r}$.

Based on the model structure and prior definitions of latent variables and measurement noise, the conditional probability of $\mathbf{x}$ and $\mathbf{y}$ can be calculated as $p(\mathbf{x}|\boldsymbol{\mu}_x, \mathbf{t}^s, \mathbf{t}^b, \mathbf{P}, \mathbf{Q}, \Sigma_x)=N(\boldsymbol{\mu}_x+\mathbf{P}\mathbf{t}^s+\mathbf{Q}\mathbf{t}^b, \Sigma_x)$ and $p(\mathbf{y}|\boldsymbol{\mu}_y, \mathbf{t}^s, \mathbf{C}, \Sigma_y)=N(\boldsymbol{\mu}_y+\mathbf{C}\mathbf{t}^s, \Sigma_y)$. The marginal probability of the samples in the labeled and unlabeled datasets $p(\mathbf{x},\mathbf{y})$ and $p(\mathbf{x})$ can be further formulated by integrating out the latent factor, given as follows

$$p(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} | \begin{bmatrix}\mu_x\\ \mu_y\end{bmatrix}, \begin{bmatrix}\mathbf{PQ}\\ \mathbf{C0}\end{bmatrix}, \begin{bmatrix}\Sigma_x\\ \Sigma_y\end{bmatrix}) = \int p(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} | \begin{bmatrix}\mu_x\\ \mu_y\end{bmatrix}, \begin{bmatrix}\mathbf{t}^s\\ \mathbf{t}^b\end{bmatrix}, \begin{bmatrix}\mathbf{PQ}\\ \mathbf{C0}\end{bmatrix}, \begin{bmatrix}\Sigma_x\\ \Sigma_y\end{bmatrix})p(\begin{bmatrix}\mathbf{t}^s\\ \mathbf{t}^b\end{bmatrix})d\begin{bmatrix}\mathbf{t}^s\\ \mathbf{t}^b\end{bmatrix} \tag{7}$$

$$p(\mathbf{x}|\boldsymbol{\mu}_x, [\mathbf{PQ}], \Sigma_x) = \int p(\mathbf{x}|\boldsymbol{\mu}_x, \begin{bmatrix}\mathbf{t}^s\\ \mathbf{t}^b\end{bmatrix}, [\mathbf{PQ}], \Sigma_x)p(\begin{bmatrix}\mathbf{t}^s\\ \mathbf{t}^b\end{bmatrix})d\begin{bmatrix}\mathbf{t}^s\\ \mathbf{t}^b\end{bmatrix} \tag{8}$$

For the given dataset $\{\mathbf{X}_1 \in R^{n_1\times m}, \mathbf{Y} \in R^{n_1\times r}\}$ and $\mathbf{X}_2 \in R^{n_2\times r}$, the parameter set of the semisupervised probabilistic PLS model $\boldsymbol{\Theta}=\{\boldsymbol{\mu}_x, \boldsymbol{\mu}_y, \mathbf{P}, \mathbf{Q}, \mathbf{C}, \Sigma_x, \Sigma_y\}$ can be determined by maximizing the following log-likelihood function

$$\begin{aligned}
\boldsymbol{\Theta}^* &= arg\max\{L(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{\Theta})\}\\
\mathrm{L}(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{\Theta}) &= L(\boldsymbol{X}_1,\boldsymbol{Y}|\boldsymbol{\Theta}) + L(\boldsymbol{X}_2|\boldsymbol{\Theta})\\
&= \ln\prod_{i=1}^{n_1} p(\boldsymbol{x}_i,\boldsymbol{y}_i|\boldsymbol{\Theta}) + \ln\prod_{i=n_1+1}^{n} p(\boldsymbol{x}_i|\boldsymbol{\Theta}) = \sum_{i=1}^{n_1}\ln p(\boldsymbol{x}_i,\boldsymbol{y}_i|\boldsymbol{\Theta}) + \sum_{i=n_1+1}^{n}\ln p(\boldsymbol{x}_i|\boldsymbol{\Theta})
\end{aligned} \tag{9}$$

To maximize this log-likelihood function, the efficient EM algorithm can be used. The main advantage of the EM algorithm is that it can guarantee the log likelihood value never decreases when this algorithm is carried out iteratively. Therefore, the optimal value of the parameter set $\boldsymbol{\Theta}=\{\boldsymbol{\mu}_x, \boldsymbol{\mu}_y, \mathbf{P}, \mathbf{Q}, \mathbf{C}, \Sigma_x, \Sigma_y\}$ can be finally determined when the EM algorithm converges. Since we do not know the value of the latent variables, instead of maximizing the log-likelihood function directly, the maximization of the expected complete-data log-likelihood function is usually carried out, which consists of two iterative steps: the Expectation-step (E-step) and the Maximization-step (M-step) [35].

With respect to the latent variables $\mathbf{t}^s$ and $\mathbf{t}^b$, the expected complete-data log-likelihood function value can be determined as follows

$$\begin{aligned}
E[L(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\Theta})] &= \sum_{i=1}^{n_1}\int p(\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix} | \begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{bmatrix}, \boldsymbol{\Theta}_{old})\ln[p(\begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{bmatrix}, \begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\boldsymbol{\Theta})]d\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix} + \sum_{i=n_1+1}^{n}\int p(\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\boldsymbol{x}_i, \boldsymbol{\Theta}_{old})\ln[p(\boldsymbol{x}_i, \begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\boldsymbol{\Theta})]d\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}\\
&= \sum_{i=1}^{n_1}\int p(\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix} | \begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{bmatrix}, \boldsymbol{\Theta}_{old})\ln[p(\begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{bmatrix}, \begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\boldsymbol{\Theta})p(\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\boldsymbol{\Theta})]d\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix} + \sum_{i=n_1+1}^{n}\int p(\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\boldsymbol{x}_i, \boldsymbol{\Theta}_{old})\ln[p(\boldsymbol{x}_i|\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}, \boldsymbol{\Theta})p(\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\boldsymbol{\Theta})]d\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}
\end{aligned} \tag{10}$$

In the E-step of the EM algorithm, we are given the parameters Θ_{old} obtained in the previous M-step, the aim of this step is to determine the value of expected complete-data log-likelihood function, which resorts to the posterior probability of the latent variables. The posterior probabilities of the latent variables can be calculated based on the Bayesian rule. For the labeled and unlabeled datasets, respectively, the posterior probabilities of the latent variables are calculated as follows, in which the constant items of the Bayesian formula are omitted.

$$\begin{aligned}
&p(\begin{bmatrix}\mathbf{t}_i^s\\ \mathbf{t}_i^b\end{bmatrix} | \begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix}, \boldsymbol{\Theta}_{old}) \propto p(\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix} | \begin{bmatrix}\mathbf{t}_i^s\\ \mathbf{t}_i^b\end{bmatrix}, \boldsymbol{\Theta}_{old})p(\begin{bmatrix}\mathbf{t}_i^s\\ \mathbf{t}_i^b\end{bmatrix})\\
&i=1,2,\cdots,n_1
\end{aligned} \tag{11}$$

$$p(\begin{bmatrix}\mathbf{t}_i^s\\ \mathbf{t}_i^b\end{bmatrix}|\mathbf{x}_i,\boldsymbol{\Theta}_{old})\propto p(\mathbf{x}_i|\begin{bmatrix}\mathbf{t}_i^s\\ \mathbf{t}_i^b\end{bmatrix},\boldsymbol{\Theta}_{old})p(\begin{bmatrix}\mathbf{t}_i^s\\ \mathbf{t}_i^b\end{bmatrix})$$
$$i=n_1+1,n_1+2,\cdots,n \tag{12}$$

Since all terms in the right side of Eqs. (11) and (12) are Gaussian, the posterior distributions of the latent variables are also Gaussian, the two sufficient statistics (mean and variance) of which can be calculated as follows

For labeled dataset:

$$\mathrm{E}\left(\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{bmatrix},\boldsymbol{\Theta}_{old}\right)=\begin{bmatrix}\boldsymbol{P}&\boldsymbol{Q}\\ \boldsymbol{C}&0\end{bmatrix}^T\left(\begin{bmatrix}\Sigma_x&0\\ 0&\Sigma_y\end{bmatrix}+\begin{bmatrix}\boldsymbol{P}&\boldsymbol{Q}\\ \boldsymbol{C}&0\end{bmatrix}\begin{bmatrix}\boldsymbol{P}&\boldsymbol{Q}\\ \boldsymbol{C}&0\end{bmatrix}^T\right)^{-1}\begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{bmatrix} \tag{13}$$

$$E([\begin{matrix}\boldsymbol{t}_i^s\boldsymbol{t}_i^{sT}\\ \boldsymbol{t}_i^b\boldsymbol{t}_i^{bT}\end{matrix}]|[\begin{matrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{matrix}],\boldsymbol{\Theta}_{old})=\boldsymbol{I}-[\begin{matrix}\boldsymbol{P}&\boldsymbol{Q}\\ \boldsymbol{C}&0\end{matrix}]^T([\begin{matrix}\Sigma_x&0\\ 0&\Sigma_y\end{matrix}]+[\begin{matrix}\boldsymbol{P}&\boldsymbol{Q}\\ \boldsymbol{C}&0\end{matrix}][\begin{matrix}\boldsymbol{P}&\boldsymbol{Q}\\ \boldsymbol{C}&0\end{matrix}]^T)-1[\begin{matrix}\boldsymbol{P}&\boldsymbol{Q}\\ \boldsymbol{C}&0\end{matrix}]+\mathrm{E}([\begin{matrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{matrix}]|[\begin{matrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{matrix}],\boldsymbol{\Theta}_{old})E^T([\begin{matrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{matrix}]|[\begin{matrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{matrix}],\boldsymbol{\Theta}_{old}) \tag{14}$$

where $i=1,2,\cdots,n_1$, and

For unlabeled dataset:

$$\mathrm{E}(\begin{bmatrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{bmatrix}|\boldsymbol{x}_i,\boldsymbol{\Theta}_{old})=[\boldsymbol{P}\ \ \boldsymbol{Q}]^T(\Sigma_x+[\boldsymbol{P}\ \ \boldsymbol{Q}][\boldsymbol{P}\ \ \boldsymbol{Q}]^T)^{-1}\boldsymbol{x}_i \tag{15}$$

$$E([\begin{matrix}\boldsymbol{t}_i^s\boldsymbol{t}_i^{sT}\\ \boldsymbol{t}_i^b\boldsymbol{t}_i^{bT}\end{matrix}]|\boldsymbol{x}_i,\boldsymbol{\Theta}_{old})=\boldsymbol{I}-[\boldsymbol{P}\ \ \boldsymbol{Q}]^T\left(\Sigma_x+[\boldsymbol{P}\ \ \boldsymbol{Q}][\boldsymbol{P}\ \ \boldsymbol{Q}]^T\right)^{-1}[\boldsymbol{P}\ \ \boldsymbol{Q}]+\mathrm{E}([\begin{matrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{matrix}]|\boldsymbol{x}_i,\boldsymbol{\Theta}_{old})E^T([\begin{matrix}\boldsymbol{t}_i^s\\ \boldsymbol{t}_i^b\end{matrix}]|\boldsymbol{x}_i,\boldsymbol{\Theta}_{old}) \tag{16}$$

where $i=n_1+1,n_1+2,\cdots,n$.

In the M-step of the EM algorithm, the values of all parameters in $\boldsymbol{\Theta}=\{\boldsymbol{\mu}_x,\boldsymbol{\mu}_y,\mathbf{P},\mathbf{Q},\mathbf{C},\Sigma_x,\Sigma_y\}$ are updated by maximizing the likelihood function given in Eq. (10). This can be done by setting the partial derivatives of $E[L(\mathbf{X},\mathbf{Y},\boldsymbol{\Theta})]$ to zero with respect to each parameter. For the single mode of the semisupervised probabilistic PLS model, the mean values of both **x** and **y** can be determined before the implementation of the EM algorithm. Therefore, we do not need to determine those two values in this step. The results of other five updated parameters are given as follows

$$\frac{\partial E[L(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\Theta})]}{\partial\boldsymbol{P}}=0\ \Rightarrow$$
$$\boldsymbol{P}^{new}=[\sum_{i=1}^{n_1}\boldsymbol{x}_iE^T(\boldsymbol{t}_i^s|[\begin{matrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{matrix}])+\sum_{i=n_1+1}^{n}\boldsymbol{x}_iE^T(\boldsymbol{t}_i^s|\boldsymbol{x}_i)][\sum_{i=1}^{n_1}E(\boldsymbol{t}_i^s\boldsymbol{t}_i^{sT}|[\begin{matrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{matrix}])+\sum_{i=n_1+1}^{n}E(\boldsymbol{t}_i^s\boldsymbol{t}_i^{sT}|\boldsymbol{x}_i)]-1 \tag{17}$$

$$\frac{\partial E[L(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\Theta})]}{\partial\boldsymbol{Q}}=0\ \Rightarrow$$
$$\boldsymbol{Q}^{new}=[\sum_{i=1}^{n_1}\boldsymbol{x}_iE^T(\boldsymbol{t}_i^b|[\begin{matrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{matrix}])+\sum_{i=n_1+1}^{n}\boldsymbol{x}_iE^T(\boldsymbol{t}_i^b|\boldsymbol{x}_i)][\sum_{i=1}^{n_1}E(\boldsymbol{t}_i^b\boldsymbol{t}_i^{bT}|[\begin{matrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{matrix}])+\sum_{i=n_1+1}^{n}E(\boldsymbol{t}_i^b\boldsymbol{t}_i^{bT}|\boldsymbol{x}_i)]-1 \tag{18}$$

$$\frac{\partial E[L(\mathbf{X},\mathbf{Y},\boldsymbol{\Theta})]}{\partial\mathbf{C}}=0\Rightarrow\mathbf{C}^{new}=[\sum_{i=1}^{n_1}\mathbf{y}_iE^T(\mathbf{t}_i^s|\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix})][\sum_{i=1}^{n_1}E(\mathbf{t}_i^s\mathbf{t}_i^{sT}|\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix})]^{-1} \tag{19}$$

$$\frac{\partial E[L(\mathbf{X},\mathbf{Y},\boldsymbol{\Theta})]}{\partial\Sigma_x}=0\Rightarrow$$
$$\Sigma_x^{new}=\frac{1}{n}diag\{\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T-\left[\mathbf{P}^{new}\mathbf{Q}^{new}\right][\sum_{i=1}^{n_1}E(\begin{bmatrix}\mathbf{t}_i^s\\ \mathbf{t}_i^b\end{bmatrix}|\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix})\mathbf{x}_i^T+\sum_{i=n_1+1}^{n}E(\begin{bmatrix}\mathbf{t}_i^s\\ \mathbf{t}_i^b\end{bmatrix}|\mathbf{x}_i)\mathbf{x}_i^T]\} \tag{20}$$

$$\frac{\partial E[L(\mathbf{X},\mathbf{Y},\boldsymbol{\Theta})]}{\partial\Sigma_y}=0\Rightarrow\Sigma_y^{new}=\frac{1}{n_1}diag\{\sum_{i=1}^{n_1}[\mathbf{y}_i\mathbf{y}_i^T-\mathbf{C}^{new}E(\mathbf{t}_i^s|\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix})\mathbf{y}_i^T]\} \tag{21}$$

3.2. A simple example

Based on the developed method, a simple numerical example is illustrate to evaluate the performance of the semisupervised PPLS model. This example is constructed as follows

$$\mathbf{x}=\mathbf{A}_1\mathbf{t}_1+\mathbf{A}_2\mathbf{t}_2+\mathbf{e}_x$$
$$\mathrm{y}=\mathbf{Ct}_1+\mathrm{e}_\mathrm{y} \tag{22}$$

Table 1
Estimated noise variances through two modeling methods.

Method	PPLS			Method	SSPPLS		
Numbers of labeled samples	20	100	500	Numbers of labeled samples	20	100	500
x_1	0.0395	0.0191	0.0093	x_1	0.0156	0.0126	0.0098
x_2	0.0658	0.0511	0.0352	x_2	0.0367	0.0441	0.0413
x_3	0.1547	0.1059	0.0852	x_3	0.0798	0.0849	0.0929
x_4	0.0678	0.0487	0.0439	x_4	0.0497	0.0443	0.0428
x_5	0.0299	0.0168	0.0156	x_5	0.0203	0.0161	0.0128
y	0.0289	0.0157	0.0081	y	0.0147	0.0078	0.0119

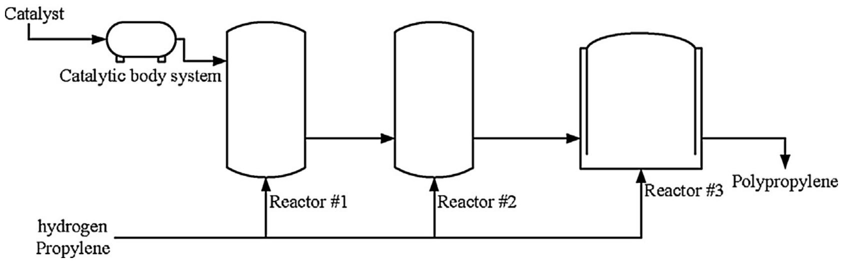

Fig. 1. Flowchart of the polypropylene production process.

Table 2
Selected variables in polypropylene production process for quality estimation.

No.	Variable description	No.	Variable description
1	Hydrogen concentration of the first reactor	8	Propylene feed of the first reactor
2	Hydrogen concentration of the second reactor	9	Propylene feed of the second reactor
3	Density of the first reactor	10	Power for the first reactor
4	Density of the second reactor	11	Power for the second reactor
5	TEAL flow	12	Lever of the second reactor
6	DONOR flow	13	Temperature of the first reactor
7	Atmer-163 flow	14	Temperature of the second reactor

where $\mathbf{A}_1 \in R^{5\times 2}$, $\mathbf{A}_2 \in R^{5\times 2}$ and $\mathbf{C} \in R^{1\times 2}$ are loading matrices, $\mathbf{t}_1 \in R^{2\times 1}$ and $\mathbf{t}_2 \in R^{2\times 1}$ are two latent variable vectors, and $\mathbf{e}_x \in R^{5\times 1}$ and $e_y \in R^{1\times 1}$ are measurement noise of $\mathbf{x}$ and y. In this example, different levels of measurement noise are assumed, given as follows

$$\begin{aligned} &\{\mathbf{e}_x\} \sim N(0, \boldsymbol{\Sigma}_e), \ \boldsymbol{\Sigma}_e = diag([0.1^2 0.2^2 0.3^2 0.2^2 0.1^2]) \\ &\{e_y\} \sim N(0, 0.1^2) \end{aligned} \tag{23}$$

A total of 1000 samples are generated through Eq. (22). In order to test the performance of both PPLS and semisupervised PPLS models, different numbers of labeled samples can be assumed for model developments. Detailed estimated variances of measurement noise are given in Table 1, which is average values of 100 running times. It can be seen that the estimation results become better with additional unlabeled samples. With the increased number of labeled samples, the two methods show close estimation results of the training dataset.

4. Industrial case study

Polypropylene is an important industrial material, which has been widely used in many fields, such as chemical industry, light industry, and medical industry. A Spheripol craft polypropylene device contains a catalytic body system, which comprises of TiCl4, triethylaluminum (TEAL), and diphenyldimethoxysilane (DONOR). To produce different brands of productions, three reactors are connected in series. The flowchart of the polypropylene production process is given in Fig. 1. The whole process includes four major units, and over 40 variables are measured. As a key performance index of the product quality, the melt index plays an important role in quality control in this process. Typically, this important variable is measured through lab analysis, which may take several hours. As a result, there are no measurements during the gap between two lab analyses for the melt index. This is quite harmful to both of the quality monitoring and control systems. In order to obtain continuous measurements for the melt index in this process, a data-based soft sensor should be developed.

A total of 14 easy-to-measure process variables have been selected as the inputs of the soft sensor, which are highly correlated with the melt index. Detailed descriptions of those 14 variables are provided in Table 2. In order to examine the data characteristics, three key process variables and the melt index of the training dataset are shown in Fig. 2(a) and (b), respectively. For development and testing of the semisupervised probabilistic PLS model, a total of 800 data samples have been obtained from a real industrial process, which is located in Nanjing, China, and the sampling time of each data sample is 1 min. Among those 800 data samples, half of them are used as the training dataset which are sampled every other point in the original dataset, while the rest part is used as the testing dataset.

To simulate the semisupervised data structure, different numbers of labeled samples are assumed. Following the modeling procedure of the developed method, given the training dataset, the parameter set of the PPLS model can be obtained through the EM algorithm, given in Section 3. For performance comparison, the basic probabilistic PLS model based soft sensor has also been developed, in which only the

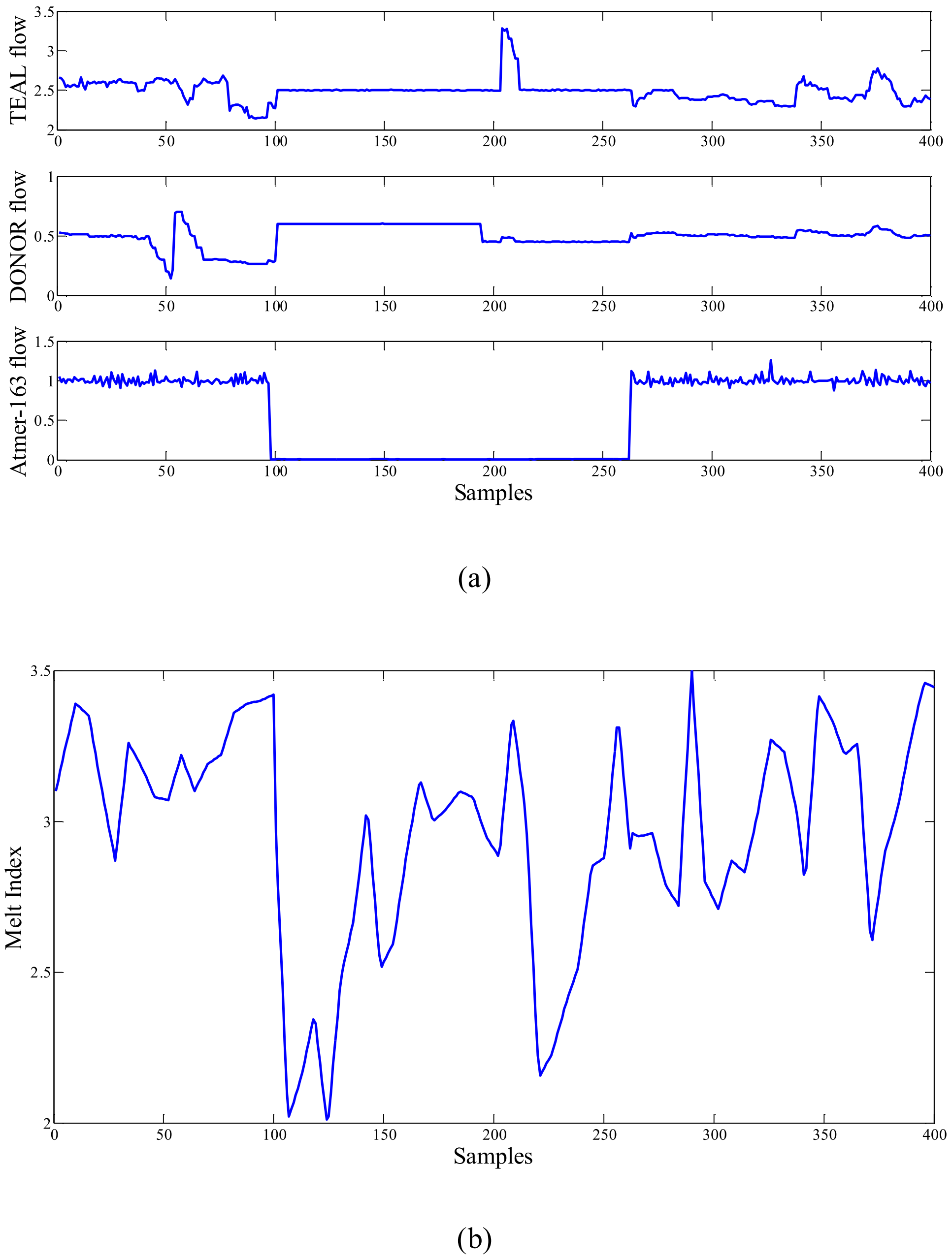

Fig. 2. Data characteristics, (a) Process variables; (b) Quality variable.

labeled data samples are incorporated. For a new data sample $\mathbf{x}_{new}$ from the process, the latent variables can be first calculated based on the soft sensor model, given as

$$\hat{\mathbf{t}}_{new} = [\mathbf{PQ}]^T\left(\Sigma_x + [\mathbf{PQ}]\,[\mathbf{PQ}]^T\right)^{-1}(\mathbf{x}_{new} - \boldsymbol{\mu}_x) \tag{24}$$

Then, the output of the soft sensor can be determined as

$$\hat{\mathbf{y}}_{new} = \mathbf{C}\hat{\mathbf{t}}_{new} = \mathbf{C}[\mathbf{PQ}]^T\left(\Sigma_x + [\mathbf{PQ}]\,[\mathbf{PQ}]^T\right)^{-1}(\mathbf{x}_{new} - \boldsymbol{\mu}_x) \tag{25}$$

where $\boldsymbol{\mu}_x$ is the mean value of the process variables determined in the modeling phase.

For performance evaluation of the developed soft sensors, the root mean square error (RMSE) criterion is typically used, which is defined as follows

$$RMSE = \sqrt{\frac{\sum_{j=1}^{L} \|\mathbf{y}_j - \hat{\mathbf{y}}_j\|^2}{L}} \tag{26}$$

where $j = 1, 2, \cdots, L$, $\mathbf{y}_j$ and $\hat{\mathbf{y}}_j$ are real and predicted values, respectively, and L is the total number of testing data samples.

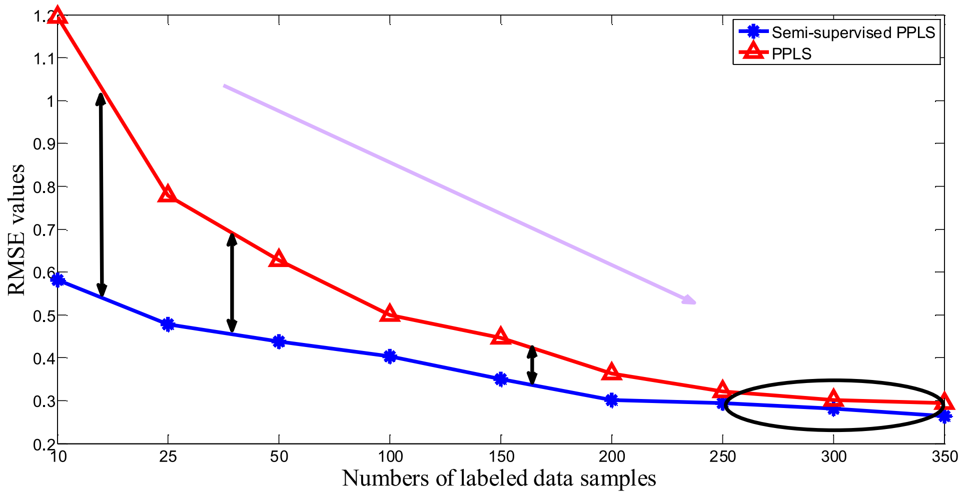

Fig. 3. RMSE values of the two methods for different cases.

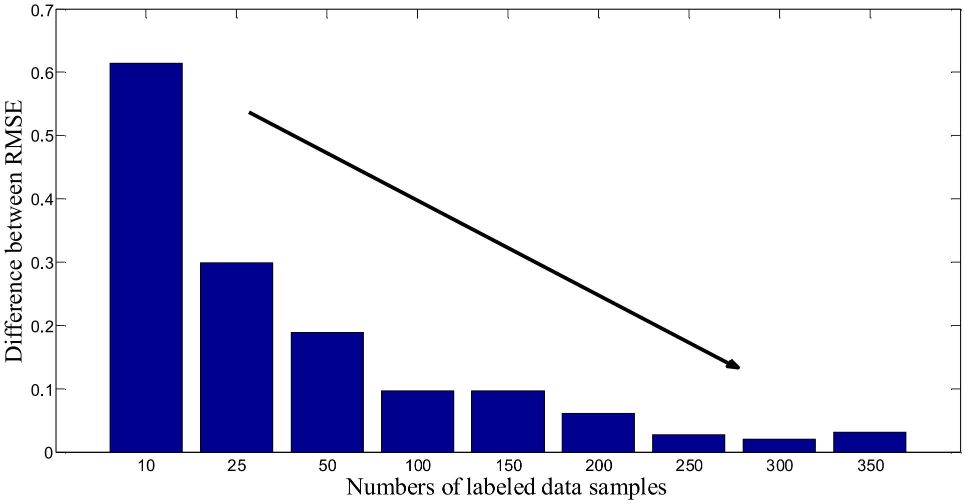

Fig. 4. Difference between RMSE values of the two methods for different cases.

Table 3
RMSE values of different semisupervised methods with 50% labeled samples.

Methods	SSPPLS	Co-training PLS	SSPPCR	Self-training PLS	Self-training PCR
RMSE	0.3121	0.3322	0.3678	0.3512	0.3916

In terms of the RMSE index, comparative results between the probabilistic PLS model and semisupervised probabilistic PLS model based soft sensors are given in Fig. 3. As can be seen in this figure, different numbers of labeled data samples have been assumed in the training dataset, which are between 10 and 350. The quality prediction performance has been greatly improved by using the semisupervised model, particularly when the number of labeled data samples is small compared to the whole size of the training dataset. With the increasing of the number of labeled data samples in the training dataset, the soft sensing performance will be improved for both of the two soft sensors. Simultaneously, the performance disparity between the semisupervised and the basic probabilistic PLS model based soft sensors become smaller. This is because more information has been included in both of the two soft sensor models. In this case, the importance of incorporating more unlabeled data samples into the data model becomes less significant. The RMSE disparities between the two soft sensors in different cases are shown in Fig. 4. Fig. 5 gives detailed estimation results of the melt index for the whole testing dataset, in which the number of labeled data samples is half of the whole training data samples. For comparison, the detailed estimation results of the basic probabilistic PLS model based soft sensor are shown in Fig. 6. Besides, Fig. 7 presents the results of both methods in comparison with the real values, in which it can be seen that the new method performance better for most testing data samples. Furthermore, in order to compare the performance of the proposed method with other semisupervised modeling methods, RMSE values of the testing dataset generated by several different semisupervised modeling methods are summarized in Table 3, including Semisupervised PPCR model, co-training PLS model, self-training PLS model, and self-training PCR model. It can be seen that the developed method has obtained the best result. Compared to models based on PCR, those developed through the PLS model structure provide better results.

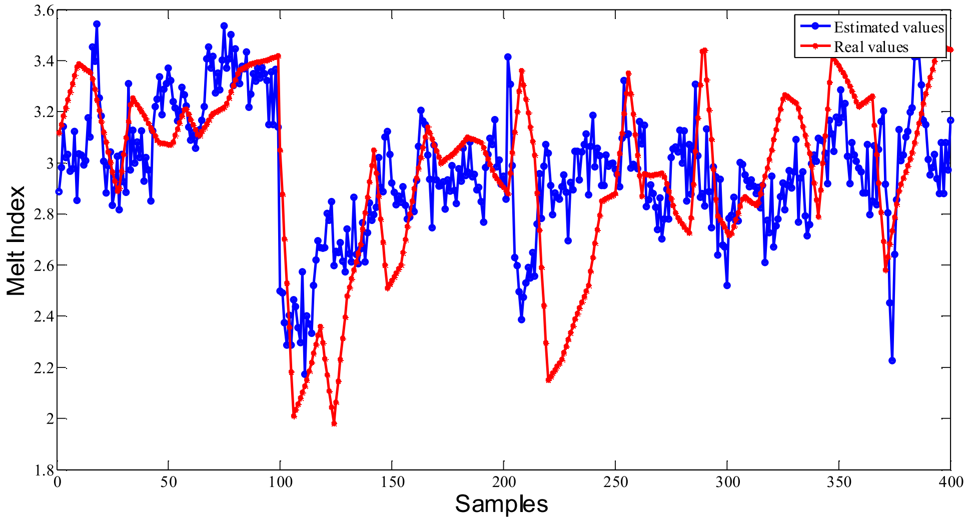

Fig. 5. Quality prediction results of the semisupervised probabilistic PLS model.

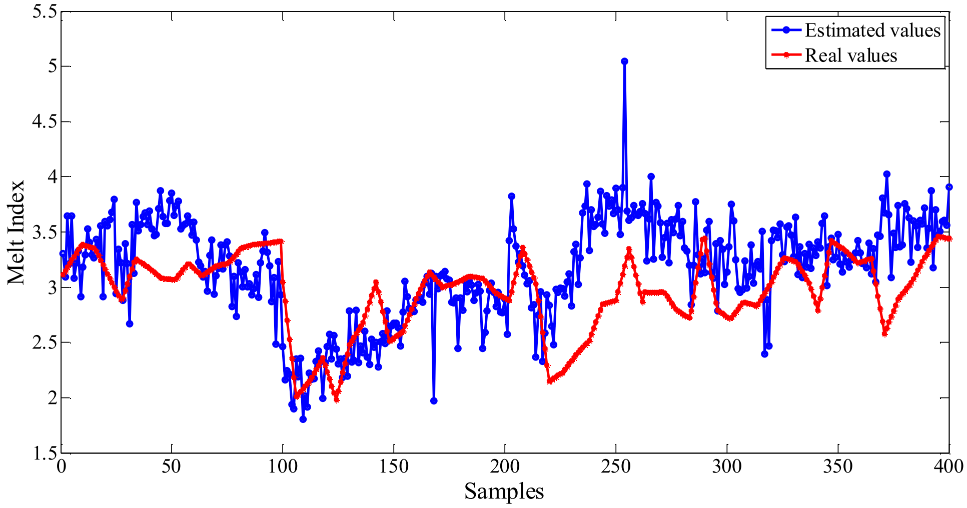

Fig. 6. Quality prediction results of the probabilistic PLS model.

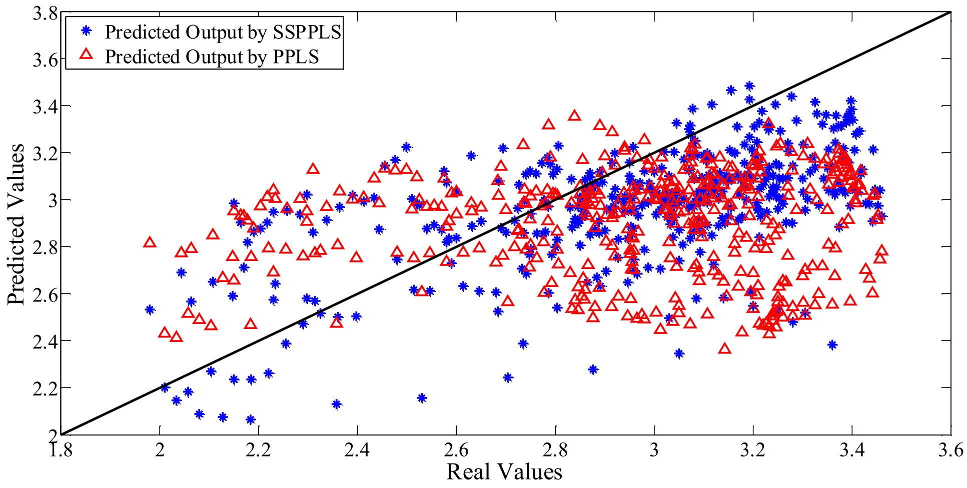

Fig. 7. Scatter plot of both SSPPLS and PPLS methods with 50% labeled samples.

5. Conclusions

In this paper, a semisupervised form of the probabilistic PLS model has been developed for soft sensor application in industrial processes. An efficient EM algorithm was designed for model training and parameter learning. With the incorporation of additional unlabeled data samples for modeling, the performance of the developed soft sensor can be significantly improved. A real industrial example has been provided for detailed comparative studies between the semisupervised PPLS model and the basic PPLS model. It has been demonstrated that the semisupervised model based soft sensor obtained better results, particularly when the number of labeled samples are much smaller than that of the unlabeled samples.

Although the developed method show superior performance to the basic supervised modeling method, it still has spaces to improve it. For example, the nonlinearity among process variables has not been considered, as well as the dynamical nature of process data. For a semisupervised modeling method, it is critical to determine how many labeled samples we need to guarantee its performance, however, this may differ from case to case. Besides, different semisupervised modeling methods have their own strengths and shortcomings in different situations, the performance may be greatly improved by combining them together through an efficient way.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (NSFC) (61673337).

References

[1] Z. Ge, Z. Song, S.X. Ding, B. Huang, Data mining and analytics in the process industry: the role of machine learning, IEEE Access 5 (2017) 20590–20616.
[2] Z. Ge, Review on data-driven modeling and monitoring for plant-wide industrial processes, Chemom. Intell. Lab. Syst. 171 (2017) 16–25.
[3] J. MacGregor, A. Cinar, Monitoring fault diagnosis, fault-tolerant control and optimization: data driven methods, Comput. Chem. Eng. 47 (2012) 111–120.
[4] Z. Ge, Z. Song, F. Gao, Review of recent research on data-based process monitoring, Ind. Eng. Chem. Res. 52 (2013) 3543–3562.
[5] X. Wang, Data Mining and Knowledge Discovery for Process Monitoring and Control, Springer, London, 2012.
[6] L. Chiang, R. Braatz, E. Russell, Fault Detection and Diagnosis in Industrial Systems, Springer, 2001.
[7] U. Kruger, L. Xie, Statistical Monitoring of Complex Multivariate Processes, JohnWiley & Sons Ltd, West Sussex, UK, 2012.
[8] Z. Ge, Z. Song, Multivariate Statistical Process Control: Process Monitoring Methods and Applications, Springer, London, 2013.
[9] K. Fujiwara, M. Kano, S. Hasebe, A. Takinami, Soft-sensor development using correlation-based just-in-time modeling, AIChE J. 55 (2009) 1754–1765.
[10] H. Kaneko, K. Funatsu, Classification of the degradation of soft sensor models and discussion on adaptive models, AIChE J. 59 (2013) 2339–2347.
[11] H. Kaneko, T. Okada, K. Funatsu, Selective use of adaptive soft sensors based on process state, Ind. Eng. Chem. Res. 53 (2014) 15962–15968.
[12] H. Galicia, Q. He, J. Wang, Comparison of the performance of a reduced-order dynamic PLS soft sensor with different updating schemes for digester control, Contr. Eng. Pract. 20 (2012) 747–760.
[13] W. Ni, S. Tan, W. Ng, S. Brown, Localized adaptive recursive partial least squares regression for dynamic system modeling, Ind. Eng. Chem. Res. 51 (2012) 8025–8039.
[14] J. Yu, A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses, Comput. Chem. Eng. 41 (2012) 134–144.
[15] K. Zhang, H. Hao, Z. Chen, S. Ding, K. Peng, A comparison and evaluation of key performance indicator-based multivariate statistics process monitoring approaches, J. Process Contr. 33 (2015) 112–126.
[16] Y. Liu, N. Hu, H. Wang, P. Li, Soft chemical analyzer development using adaptive least-squares support vector regression with selective pruning and variable moving window size, Ind. Eng. Chem. Res. 48 (2009) 5731–5741.
[17] 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. 82 (2012) 22–30.
[18] S. Khatibisepehr, B. Huang, F. Xu, A. Espejo, A Bayesian approach to design of adaptive multi-model inferential sensors with application in oil sand industry, J. Process Contr. 22 (2012) 1913–1929.
[19] M. Tipping, C. Bishop, Probabilistic principal component analysis, J. R. Stat. Soc.: Ser. B 61 (1999) 611–622 (Statistical Methodology).
[20] Z. Ge, F. Gao, Z. Song, Mixture probabilistic PCR model for soft sensing of multimode processes, Chemom. Intell. Lab. Syst. 105 (2011) 91–105.
[21] Z. Ge, Supervised latent factor analysis for process data regression modeling and soft sensor application, IEEE Trans. Control Syst. Technol. 24 (2016) 1004–1011.
[22] Z. Ge, Z. Song, Nonlinear soft sensor development based on relevance vector machine, Ind. Eng. Chem. Res. 49 (2010) 8685–8693.
[23] Z. Ge, T. Chen, Z. Song, Quality prediction for polypropylene production process based on CLGPR model, Control Eng. Pract. 19 (2011) 423–432.
[24] Y. Liu, J. Chen, Integrated soft sensor using just-in-time support vector regression and probabilistic analysis for quality prediction of multi-grade processes, J. Process Contr. 23 (2013) 793–804.
[25] J. Zhu, Z. Ge, Z. Song, Robust Supervised Probabilistic Principal Component Analysis model for soft sensing of key process variables, Chem. Eng. Sci. 122 (2015) 573–584.
[26] M. Ma, S. Khatibisepehr, B. Huang, A. Bayesian, framework for real-time identification of locally weighted partial least squares, AIChE J. 61 (2) (2015) 518–529.
[27] C. Shang, B. Huang, F. Yang, et al., Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling, AIChE J. 61 (12) (2015) 4126–4139.
[28] J. Zheng, Z. Song, Z. Ge, Probabilistic learning of partial least squares regression model: theory and industrial applications, Chemom. Intell. Lab. Syst. 158 (2016) 80–90.
[29] B. Chapelle, Semisupervised Learning (O. Chapelle, et al., Eds. 2006) IEEE Transactions on Neural Networks, 20, 3, (2009) 542.
[30] Z. Yan, C. Huang, Y. Yao, Semisupervised mixture discriminant monitoring for chemical batch processes, Chemom. Intell. Lab. Syst. 134 (2014) 10–22.
[31] Z. Ge, Z. Song, Semisupervised Bayesian method for soft sensor modeling with unlabeled data samples, AIChE J. 57 (2011) 2109–2119.
[32] L. Bao, X. Yuan, Z. Ge, Co-training partial least squares model for semisupervised soft sensor development, Chemom. Intell. Lab. Syst. 147 (2015) 75–85.
[33] S.Y. Zhong, Q.J. Wen, Z.Q. Ge, Semisupervised Fisher discriminant analysis model for fault classification in industrial processes, Chemom. Intell. Lab. Syst. 138 (2014) 203–211.
[34] X. Yuan, Z. Ge, B. Huang, Z. Song, Y. Wang, Semisupervised JITL framework for nonlinear industrial soft sensing based on locally semisupervised weighted PCR, IEEE Trans. Ind. Inf. 13 (2017) 532–541.
[35] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B 39 (1977) 1–39.