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/

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

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 $\mathbf{x}$ and $\mathbf{y}$, thus $\Sigma_x = diag\{\sigma^2_{x,u}\}_{u=1,2,\cdots,m}$ and $\Sigma_y = diag\{\sigma^2_{y,v}\}_{v=1,2,\cdots,r}$. 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 $\mathbf{x}$, $\mathbf{y}$ 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]

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

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.