Journal of the Taiwan Institute of Chemical Engineers 67 (2016) 20–28

Contents lists available at ScienceDirect

Journal of the Taiwan Institute of Chemical Engineers

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

Development of a soft sensor for processes with multiple operating regimes using adaptive multi-state partial least squares regression

Wei Guo, Tianhong Pan*, Zhengming Li

School of Electrical Information Engineering, Jiangsu University, Zhenjiang 212013, China

ARTICLE INFO

Article history:
Received 7 January 2016
Revised 3 June 2016
Accepted 19 July 2016
Available online 29 July 2016

Keywords:
Soft sensor
Data-driven
Multi-state partial least squares
Adaptive scheme

ABSTRACT

This paper proposes an adaptive multi-state partial least squares (MSPLS) algorithm for multivariate chemical processes over a wide range of operating conditions. In the proposed algorithm, the state variable with the maximum variation is first selected from the defined key variables. The system is then divided into several states according to the rank of this state variable. The deviation is subtracted from the process variables in each state, resulting in a set of unified process variables that are then combined to form the PLS model. Finally, an adaptive scheme is designed to generalize the performance of MSPLS. Applications to a continuous stirred tank reactor and a real industrial process are used to evaluate the proposed algorithm.

1. Introduction

In industrial processes, some response variables cannot be measured online, or can only be measured at very low frequencies. However, these variables are essential for monitoring and timely control. Consequently, inferential models (*i.e.*, soft sensors) developed using knowledge-based or data-driven strategies have been designed to predict these unmeasurable response variables [1,2]. Over recent decades, soft sensors have been widely used in many industrial processes involving petrochemicals, fermentation, and polymers [3–6].

Knowledge-based strategies can describe plant behavior under all operating conditions [7]. However, it is difficult to compute the exact model parameters, because some a priori physical knowledge and the chemical background are required [8]. Furthermore, the calculation of such parameters is time-consuming [9].

Concomitant with developments in computer science and measurement techniques, the collection and storage of operating data are more convenient than ever before. Accumulated historical records contain a wealth of useful information, facilitating the construction of inferential models using data-driven methods,*e.g.*, artificial neural networks (ANNs), support vector machines (SVMs), multiple linear regression (MLR), principal component regression (PCR), and partial least squares (PLS) regression [10–13].

An ANN model is a type of universal function approximator based on the empirical risk minimization (ERM) principle. ANNs can approximate any nonlinear function with an appropriate number of nodes and hidden layers [14]. Unfortunately, the optimal topology of the network is usually unknown, and over-fitting is a serious problem in ANNs [15]. Furthermore, ANNs are prone to becoming stuck in local optima during the learning phase [16]. Unlike ANNs, SVMs utilize the structural risk minimization (SRM) principle, which balances the quality of the learning data against the complexity of the approximating function [17]. However, the training complexity of SVMs is increased by a large dataset. Although some fast training algorithms have been presented to overcome this issue, the user should choose a suitable optimization strategy in accordance with specific conditions [18].

Traditional MLR techniques such as the least squares (LS) approach are basic regression models. These may suffer from numerical problems when the process variables are strongly collinear [19]. PCR/PLS models overcome this issue by projecting the multidimensional space of the original process variables onto a subspace. The resulting projection is defined by the orthogonal principal components (PCs) or latent variables (LVs) [20]. However, PCR obtains the PCs without any reference to the response variables. As a result, the derived PCs, which explain a considerable amount of the variation in the process variables, may not be related to the variation of the response variables [21,22]. In contrast to PCR, PLS simultaneously decomposes the process variables and response variables with constrained LVs. These LVs contain the maximum variation and correlation between the process variables and response variables.

However, the accuracy of traditional PLS may be degraded when new online process information is introduced, because the

* Corresponding author.
E-mail address: thpan@live.com (T. Pan).

http://dx.doi.org/10.1016/j.jtice.2016.07.018
1876-1070/

PLS model is built offline using historical data, and thus cannot handle these new data. To rectify this problem, recursive partial least squares (RPLS) regression was developed [23,24]. RPLS algorithms recursively update the PLS model without increasing the size of the data matrices, thereby avoiding the need to store large amounts of numerical data and perform computationally expensive mathematical operations.

PLS is usually based on a global predictive model with the aim of achieving a universal generalization performance. For multiple operating regimes, process variables often need to vary over a wide range to cover the diverse range of products [25]. The change across entire operating regimes is greater than the variation in one region, which may lead to inaccurate predictions in some local regions where the process characteristics have changed [26]. Using divide-and-conquer techniques, local modeling methods such as multiple models and just-in-time learning (JITL) have been proposed to deal with this issue [26,27]. However, the computational burden of JITL and the lack of adaptive schemes for multiple models mean that local modeling methods are defectively used with multiple operating regime systems [28,29]. Consequently, this paper proposes a soft sensor based on multi-state partial least squares (MSPLS) regression, and discusses its adaptability.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of the PLS model. Section 3 describes the determination of mean metrics for search deviations and the adaptive MSPLS modeling procedure. Section 4 applies the model to a simulated continuous stirred tank reactor (Section 4.1) and a real industrial process (Section 4.2). Finally, Section 5 concludes this paper.

2. Partial least squares model

Consider a pair of datasets $X \in R^{K \times N}$ and $Y \in R^{K \times L}$, where X, Y are the process and response variables, respectively, and $\bar{X}$ and $\bar{Y}$ (zero mean and unit variance) are the centered and scaled forms of X, Y. The traditional PLS model projects $\bar{X}$, $\bar{Y}$ onto subset LVs that are defined by an input score matrix T and an output score matrix U:

$$\begin{cases} \bar{X} = \sum_{j=1}^{a} t_j p_j' + E_a = TP' + E_a \\ \bar{Y} = \sum_{j=1}^{a} u_j q_j' + F_a = UQ' + F_a \end{cases} \tag{1}$$

where $T = [t_1, t_2, \ldots, t_a]$ (unit length), $P = [p_1, p_2, \ldots, p_a]$ and $Q = [q_1, q_2, \ldots, q_a]$ are the loading matrices of X and Y. E, F represent residual matrices. Eq. (1) contains bilinear terms that represent the outer model of $\bar{X}$ and $\bar{Y}$. These terms require a connection known as the inner model between T and U:

$$\hat{U} = BT \tag{2}$$

$$b_i = \frac{u_i' t_i}{t_i' t_i} \qquad i = 1, \ldots, a \tag{3}$$

where $B = diag[b_1, b_2, \ldots, b_a]$ is the linear model assumed to relate T and U. Thus, the PLS model is built, and the prediction of $\bar{Y}$ is given by

$$\hat{\bar{Y}} = BTQ' = B\bar{X}W\left(P'W\right)^{-1}Q' \tag{4}$$

The total number of LVs in the PLS model is N, but it is not necessary for all N to contribute to the estimated output. Several methods can be used to determine the ideal number of LVs for regression. One such method is to choose the number of LVs present when $\|F_{a-1}\| - \|F_a\|$ falls below a certain threshold [21]. Another is k-fold cross-validation, in which the optimal number of LVs can be chosen with the minimum predicted error sum of squares (PRESS) [30].

3. Adaptive multi-state partial least squares method

The proposed adaptive MSPLS algorithm is used for multiple operating regime systems, in which some process variables may change over a wider range than others to cooperate with different regime operations. The basic idea underlying the adaptive MSPLS algorithm is to use the deviation matrices $\{\hat{X}, \hat{Y}\}$ through $\{X, Y\}$ to build an adaptive PLS model. Unlike the PLS/RPLS models, the MSPLS algorithm eliminates the differences among multiple operating regimes (multi-states). First, the mean of the operating regime is subtracted from the samples in each operating regime (centered). The centered samples are then combined, and a PLS model is constructed. The crucial step in MSPLS is to obtain the mean matrices. This proceeds as follows:

(1) select a process variable as a basis state for dividing the operating regimes;
(2) classify samples according to the selected process variable;
(3) calculate mean metrics using these classified samples.

In general, a process variable either maintains a constant level or changes dynamically. Further, the variance can always account for the change in the response variable. Thus, this process variable is called the state variable x^s, which is determined as

$$x^s = \arg\max_{x_i} \left(\mathrm{var}(x_i)\right) \qquad i = 1, \ldots, N \tag{5}$$

The number of states 'M' can be defined based on the distribution of x^s. Further, the state number divides x^s into 'M' equal segments called the state span (D):

$$D = \frac{\max(x^s) - \min(x^s)}{M} \tag{6}$$

where $\max(x^s)$ and $\min(x^s)$ are the maximum and minimum of x^s. Accordingly, the respective bounds of each state are ξ_j^l, ξ_j^h:

$$\begin{cases} \xi_j^l = \min(x^s) + (j-1) \times D \\ \xi_j^h = \xi_j^l + D \end{cases} \qquad j = 1, \ldots, M \tag{7}$$

The state classification of sample k, *i.e.*, $x^s(k = 1, \ldots, K)$, is then defined as if $x^s \in (\xi_j^l, \xi_j^h)$, $x^s \in j^{th}$ state. Following the classification of the samples, the mean of each state is calculated as:

$$\begin{cases} \tilde{x}_j = \left(\frac{X' S e_j}{n_j}\right)' \\ \tilde{y}_j = \left(\frac{Y' S e_j}{n_j}\right)' \end{cases} \qquad j = 1, \ldots, M \tag{8}$$

where $e_j \in R^{M \times 1}$ is the unit norm vector of zeros except for the j^{th} element, and S is the membership matrix:

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1M} \\ S_{21} & S_{22} & \cdots & S_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ S_{K1} & S_{K2} & \cdots & S_{KM} \end{bmatrix} \tag{9}$$

$$S_{kj} = \begin{cases} 1 & x_k \in j^{th} \text{ state} \\ 0 & x_k \notin j^{th} \text{ state} \end{cases} \tag{10}$$

where $k = 1, \ldots, K$, $j = 1, \ldots, M$. It is obvious that $X'S$ and $Y'S$ represent the sums of the samples under different states, and e_j can eliminate other sums in addition to the jth state. The obtained means are arranged into mean matrices:

$$\tilde{X} = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_M \end{bmatrix}, \tilde{Y} = \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_M \end{bmatrix} \tag{11}$$