Chemometrics and Intelligent Laboratory Systems 146 (2015) 55–62

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

Covariance-based locally weighted partial least squares for high-performance adaptive modeling

Koji Hazama [1], Manabu Kano *

Department of Systems Science, Kyoto University, Kyoto 606-8501, Japan

ARTICLE INFO

Article history:
Received 19 March 2015
Received in revised form 4 May 2015
Accepted 5 May 2015
Available online 13 May 2015

Keywords:
Just-in-time modeling
Locally weighted partial least squares
Soft-sensor
Process analytical technology
Calibration

ABSTRACT

Locally weighted partial least squares (LW-PLS) is one of Just-in-Time (JIT) modeling methods; PLS is used to build a local linear regression model every time when output variables need to be estimated. The prediction accuracy of local models strongly depends on the definition of similarity between a newly obtained sample and past samples stored in a database. To calculate the similarity, the Euclidean distance and the Mahalanobis distance have been widely used, but they do not take account of the relationship between input and output variables. This fact limits the achievable performance of LW-PLS and other locally weight regression methods. Thus, in the present work, covariance-based locally weighted PLS (CbLW-PLS) is proposed by integrating LW-PLS and a new similarity index based on the covariance between input and output variables. CbLW-PLS was applied to two industrial problems: soft-sensor design for estimating unreacted NaOH concentration in an alkali washing tower in a petrochemical process, and process analytical technology (PAT) for estimating concentration of a residual drug substance in a pharmaceutical process. The proposed similarity index was compared with six conventional indexes based on distances, correlations, or regression coefficients. The results have demonstrated that CbLW-PLS achieved the best prediction performance of all in both case studies.

1. Introduction

Real-time monitoring and control of product quality are difficult in most manufacturing processes because product quality is not always measured in real time. On the other hand, the number of measured variables and the amount of data stored in databases are rapidly increasing. Such a situation has motivated us to predict the difficult-to-measure product quality from easily measurable process variables and to use predicted values instead of measurements for real-time monitoring and control. In other words, virtual sensing technology is crucial in predicting product quality or other important variables when online analyzers are not available [1]. Virtual sensing technology has been successfully applied to various processes in various industries. It is known as soft-sensors in the refinery/petrochemical industry, process analytical technology (PAT) in the pharmaceutical industry, and virtual metrology (VM) in the semiconductor industry.

In recent years, Just-in-Time (JIT) modeling has attracted a lot of attention in order to prevent deterioration of prediction accuracy due to changes in process characteristics and operating conditions. In fact, Kano and Ogawa reported in 2009 that the maintenance of models is the most critical issue concerning soft-sensors on the basis of the questionnaire survey of process control applications [2]. More than 30% of the engineers pointed out the necessity to cope with changes in process characteristics and operating conditions in order to keep the prediction performance of soft-sensors.

To cope with changes in process characteristics and operating conditions, various recursive methods have been proposed and their applications have been reported. A review of adaptation techniques was given by Kadlec et al. [3]. The concept drift theory was exploited to classify the algorithms into three different types: 1) moving windows techniques, 2) recursive adaptation techniques, and 3) ensemble-based methods. Recursive methods can adapt models to new operating conditions gradually, but the model may adapt excessively and not function in a sufficiently wide range of operating conditions when a process is operated within a narrow range for a certain period of time. An approach to prevent excessive recursive PLS update is minimizing the number of recursive PLS update runs while maintaining the model [4]. A more serious drawback of recursive methods is that they cannot cope with abrupt changes in process characteristics.

In such situations, JIT modeling is desirable. JIT modeling technique constructs a model every time when prediction is required so that it can adapt the model to time-varying process characteristics and operating conditions. It constructs a local model by weighting samples in a database according to the similarity between a newly obtained sample

* Corresponding author.
E-mail address: manabu@human.sys.i.kyoto-u.ac.jp (M. Kano).
[1] Present affiliation: A. T. Kearney, Tokyo, Japan.

http://dx.doi.org/10.1016/j.chemolab.2015.05.007
0169-7439/

(query) and past samples. The JIT modeling concept has been integrated with linear and nonlinear regression methods such as multiple regression analysis (MRA) [5] and support vector regression (SVR) [6,7]. JIT modeling and its industrial applications were recently surveyed by Kano and Fujiwara [1].

In particular, JIT modeling integrated with partial least squares (PLS) is called locally weighted PLS (LW-PLS) [8], which has been successfully applied to various industrial processes. In the pharmaceutical industry, for example, LW-PLS has been applied to estimation of active pharmaceutical ingredients (API) content with near infrared (NIR) spectroscopy [8], estimation of the amount of residual drug substances in cleaning processes with infrared-reflection absorption spectroscopy (IR-RAS) [9], and NIR-based real-time monitoring of ingredient concentration during blending [10]. Other applications of LW-PLS include inferential control of product quality in the petrochemical industry [11], maize hardness characterization in the food industry [12], VM in the semiconductor industry [13], and determination of clinical parameters in human serum samples with Fourier transform infrared (FTIR) spectroscopy [14]. In addition, several updating strategies including LW-PLS were compared in the prediction accuracy by using an NIR dataset of gasoline [15]. Furthermore, LW-PLS algorithm was extended to improve the estimation performance or to cope with different problems. Such extension includes locally weighted partial least squares-discriminant analysis (LW-PLS-DA) for non-linear classification [16] and a Bayesian framework providing a systematic way for real-time parameterization of the similarity function, selection of the local PLS model structure, and estimation of the corresponding model parameters [17].

The definition of similarity plays a crucial role in improving the prediction accuracy of JIT modeling technique including LW-PLS. Similarity indexes are usually defined on the basis of the Euclidean distance or the Mahalanobis distance [8,18]. Other similarity indexes take account of the angles between a query and samples in a database [6,19]. In addition, the prediction accuracy can be significantly improved by using the similarity index based on the weighted distance, whose weights are derived from physical properties of target material [9]. There have been various works that calculate the weighted distance based on the regression coefficients of MRA, PLS, and LW-PLS [5,20]. However, these methods require constructing a regression model in advance to calculate the similarity index, therefore the computational load is heavy.

In the present work, focusing on LW-PLS, we propose a new similarity index that takes account of the relationships both among input variables and among input and output variables with suppressing an increase in computational load. The proposed method is referred to as covariance-based LW-PLS (CbLW-PLS). Case studies are conducted through two different operation data in real plants to compare the proposed similarity index with other similarity indexes in the prediction performance of LW-PLS.

This paper is organized as follows: LW-PLS is described in Section 2, and the new similarity index is proposed in Section 3 The distribution of each similarity index is visualized through a numerical experiment in Section 4. The case studies are shown in Section 5 to demonstrate the effectiveness of the proposed method. Finally, the conclusion is given in Section 6.

2. Locally weighted partial least squares (LW-PLS)

In this section, PLS and LW-PLS are briefly explained.

2.1. Partial least squares (PLS)

In general, PLS is preferable to multiple regression or ordinary least squares (OLS) when a linear regression model is built from process data, because PLS can deal with multicollinearity that prevents from obtaining a reliable model by using OLS. Multicollinearity appears in a situation where input variables are nearly or completely linearly dependent; such a situation is common in process data analysis. To address this issue, PLS derives latent variables as linear combinations of input variables and uses them to predict output variables.

Suppose data of input variables and an output variable are given as $\mathbf{X} \in \Re^{N \times M}$ and $\mathbf{y} \in \Re^{N}$. These variables are mean-centered and properly scaled, e.g. normalized. A PLS model with K latent variables is expressed as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \tag{1}$$

$$\mathbf{y} = \mathbf{T}\mathbf{q} + \mathbf{f} \tag{2}$$

where $\mathbf{T} \in \Re^{N \times K}$ is a score matrix consisting of latent variables $\mathbf{t}_k \in \Re^{N}$ $(k = 1, 2, \ldots, K)$, $\mathbf{P} \in \Re^{M \times K}$ consisting of $\mathbf{p}_k \in \Re^{M}$ is a loading matrix of $\mathbf{X}$, $\mathbf{q} \in \Re^{K}$ is a regression coefficient vector from latent variables to the output variable, and $\mathbf{E}$ and $\mathbf{f}$ are residuals.

In PLS, the model is constructed in an iterative manner through the NIPALS algorithm [21]. After $\mathbf{X}_1 = \mathbf{X}$ and $\mathbf{y}_1 = \mathbf{y}$ are set, the variable matrices at the k th iteration $(k \geq 2)$ are written as

$$\mathbf{X}_k = \mathbf{X}_{k-1} - \mathbf{t}_{k-1}\mathbf{p}_{k-1}^{\mathrm{T}} \tag{3}$$

$$\mathbf{y}_k = \mathbf{y}_{k-1} - \mathbf{t}_{k-1}q_{k-1} \quad . \tag{4}$$

The k th latent variable $\mathbf{t}_k$ is expressed as

$$\mathbf{t}_k = \mathbf{X}_k\mathbf{w}_k \tag{5}$$

where the k th weighting vector $\mathbf{w}_k$, the kth column of the weighting matrix $\mathbf{W}$, is determined so that the inner product between $\mathbf{t}_k$ and $\mathbf{y}_k$ is maximized under the constraint $\|\mathbf{w}_k\| = 1$. The Lagrange multiplier method enables us to derive $\mathbf{w}_k$, $\mathbf{p}_k$, and q_k as follows.

$$\mathbf{w}_k = \frac{\mathbf{X}_k^{\mathrm{T}}\mathbf{y}_k}{\left\|\mathbf{X}_k^{\mathrm{T}}\mathbf{y}_k\right\|} \tag{6}$$

$$\mathbf{p}_k = \frac{\mathbf{X}_k^{\mathrm{T}}\mathbf{t}_k}{\mathbf{t}_k^{\mathrm{T}}\mathbf{t}_k} \tag{7}$$

$$q_k = \frac{\mathbf{y}_k^{\mathrm{T}}\mathbf{t}_k}{\mathbf{t}_k^{\mathrm{T}}\mathbf{t}_k}. \tag{8}$$

This procedure is repeated until k reaches the number of adopted latent variables K. This PLS algorithm is knows as PLS1 because the number of output variables is one; PLS2 is available when multiple output variables need to be predicted simultaneously.

2.2. Locally weighted partial least squares (LW-PLS)

LW-PLS is a JIT modeling method that constructs a local regression model according to the similarity between a query (target sample) and past samples stored in a database [8]. It has attracted much attention as a tool for virtual sensing since it can cope with changes in operating conditions and process characteristics.

Here the algorithm of LW-PLS is explained. $\{x_{nm}\}$ and $\{y_{nl}\}$ $(n = 1, 2, \ldots, N;\ m = 1, 2, \ldots, M;\ l = 1, 2, \ldots, L)$ are preprocessed measurements of input and output variables, where M and L are the numbers of input and output variables, respectively. As the preprocess, an adequate scaling is necessary to achieve high prediction performance. The same preprocess should be applied both to samples in the database and to the query. The n th sample is expressed as

$$\mathbf{x}_n = [x_{n1}, x_{n2}, \ldots, x_{nM}]^{\mathrm{T}} \tag{9}$$

$$\mathbf{y}_n = [y_{n1}, y_{n2}, \ldots, y_{nL}]^{\mathrm{T}} \quad . \tag{10}$$