Chemometrics and Intelligent Laboratory Systems 158 (2016) 138–145

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

# On-line quality prediction of batch processes using a new kernel multiway partial least squares method

Xichang Wang [a], Pu Wang [a], Xuejin Gao [a,*], Yongsheng Qi [b]

[a] College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China
[b] College of Electric Power, Inner Mongolia University of Technology, Hohhot 010051, China

ARTICLE INFO

*Article history:*
Received 18 December 2014
Received in revised form 14 June 2016
Accepted 21 June 2016
Available online 23 June 2016

*Keywords:*
Batch process
Partial least squares
Kernel partial least squares
Feature vector selection
Online prediction
Process monitoring

ABSTRACT

The conventional data-driven soft sensor methods such as multiway partial least squares have been encountering nonlinear problems in predictions of batch processes, and kernel methods have been used to deal with these problems. In this work, a new data-driven soft sensor method is proposed by developing a Reduced Dual Kernel multiway partial least squares algorithm. First, the number of kernel vectors is reduced by the feature vector selection method. Then, by projecting both input data and the output data into two reduced kernel spaces, dual kernel matrices are established. These two matrices can be used to build PLS models. Finally, the predicted data in the kernel space can be reversely projected onto its original space during online prediction. Comparisons were made among the proposed method and some pervious algorithms through a numerical example and an *Escherichia coli* fermentation batch process.



## 1. Introduction

Modern industrial batch processes are important, efficient and high-value-added, thus, sensors should be added to monitor the whole process and predict their product quality. During the operation process, various useful variables can be measured online by physical sensors and a set of enormous historical databases can be formed. These databases can be used for soft sensor technologies to estimate some key variables that cannot be easily acquired [1,2]. Without these technologies, operators may affect the monitoring of the whole process and even lead to operation delay and faulty judgments, which may cause troublesome outcomes and system failure. Soft sensor technologies such as partial least squares (PLS) have been successfully applied in industrial batch processes [3]. However, one assumption of PLS is that the monitored data is linear, which cannot be well satisfied in practice because of the nonlinearity nature of industrial variables. Rosipal [4] proposed a new kernel PLS (KPLS) algorithm, which has shown its efficiency in dealing with the nonlinear problems, however, the traditional KPLS methods have the dimension disaster problem that hard to process when historical data is to large, additionally focus only on the kernel projection of input variables, without considering the output variables. In this work, a reduced dual multiway kernel partial least squares algorithm is proposed focusing on the nonlinear problems of both input variables and output variables in batch processes, while solving the dimension problem.

This paper is organized as follows. In Section 2, the preliminaries of multiway kernel partial least squares and feature vector selection (FVS) are introduced, including data expanding method as well as the combination of basic FVS kernel trick and the basic MPLS algorithm. Section 3 describes the Reduced Dual Kernel MPLS algorithm, including the reverse projection algorithm for data in a FVS kernel space. The comparisons of different related algorithms are illustrated through a numerical example and an experiment in Section 4. Section 5 is the conclusions.

## 2. Preliminaries

### 2.1. Multiway partial least squares

Partial least squares (PLS) [5] is a data-driven mathematical algorithm used to build prediction models [6]. It has been widely used in many areas such as economics, sociology and chemometrics. The purpose of PLS is to analyze relationships between input data $X$ and output data $Y$. The general underlying model of PLS is:

$$X = TP^T + E \tag{1}$$

* Corresponding author.
*E-mail address:* gaoxuejin@bjut.edu.cn (X. Gao).

http://dx.doi.org/10.1016/j.chemolab.2016.06.017
0169-7439/

**Table 1**
NIPLS algorithm.

(1) Set $i=1, E_0=X, F_0=Y, u_1=y_1$; where $i$ is the index of the latent variable, $y_1$ is the first target variable;
(2) $w_i=E_{i-1}^T u_i/u_i^T u_i$;
(3) $t_i=E_{i-1}w_i/\|E_{i-1}w_i\|$;
(4) $q_i=F_{i-1}^T t_i/\|F_{i-1}^T t_i\|$;
(5) $u_i=F_{i-1}q_i$, go to step (2) until convergence is reached;
(6) $p_i=E_{i-1}^T t_i$;
(7) $b_i=u_i^T t_i$;
(8) $E_i=E_{i-1}-t_i p_i^T$;
(9) $F_i=F_{i-1}-u_i q_i^T$;
(10) $i=i+1$;
(11) Repeat from step (2) until all of the latent variables are calculated.

$$Y = UQ^T + F \tag{2}$$

where $X$ is an $n \times m$ matrix, $Y$ is an $n \times p$ matrix; $T$ and $U$ are $n \times R$ matrices that are, scores of $X$ and $Y$ respectively; $P$ and $Q$ are $m \times R$ and $p \times R$ loading matrices respectively; superscript $T$ represents the transpose operator. $E$ and $F$ are residuals. $R$ is the number of latent variables, which can be determined by cross validation method [7]. Table 1 shows the principle of the NIPLS (Nonlinear Iterative Partial Least Squares [8]) algorithm [2].

Multiway PLS is an extension of PLS to analyze data taken from batch processes. Since the common data format of batch processes is arranged in terms of batch number, variable number and sample number (or time number), this three-dimensional data array should be unfolded into a two-dimensional matrix before applying PLS. P. Nomikos and J. F. MacGregor [9] proposed an expanding method, which expands the batch data on the batch direction. However, due to the data filling problems it has, Aguado et al. [10] proposed a new expanding method, AT approach in short. The principles of this method are described as follows: (1) Unfold $X(I \times J \times K)$ to $X(I \times KJ)$ as Fig. 1a, (2) Normalize the data $X(I \times KJ)$ to have zero mean and standard deviation, and (3) Transform the data $X(I \times KJ)$ to $X(IK \times J)$ as shown in Fig. 1b.

### 2.2. Kernel PLS

Traditional PLS has been successfully applied under the linear assumption, however, generally the actual industrial processes have serious nonlinear problem. To solve this problem, kernel tricks (or kernel method) have been implemented in the fields of pattern analysis, process monitoring and image processing [11], and have been integrated with data-driven methods such as Principal Component Analysis and Partial Least Squares [2,12]. Kernel tricks such as Gaussian Kernel trick [4] are used to calculate some specific distance between different data and project the distance information into a high dimensional space to reduce data's nonlinear property. After the application of kernel tricks, previous data-driven methods can be then used.

For the application of kernel tricks, an appropriate projection function $f$ should be constructed to project vector $\boldsymbol{x}$ into a high dimensional space:

$$f : \boldsymbol{x} \in \mathbb{R}^n \longmapsto f(\boldsymbol{x}) \in F \subseteq \mathbb{R}^N \tag{3}$$

where $\mathbb{R}^n$ is an $n$-dimensional real space, and $\mathbb{R}^N$ is an $N$-dimensional real space. The kernel function calculates the inner product in the high dimensional space F, as shown in Eq. (4).

$$\mathrm{k}(\boldsymbol{x}, \boldsymbol{z}) = \langle f(\boldsymbol{x}), f(\boldsymbol{z}) \rangle \tag{4}$$

where $\boldsymbol{x}$, $\boldsymbol{z}$ denote different vectors with the same element number.

In this paper, the Gaussian kernel function (or Radial Basis Function in other words) [13–18] is used as one step of the data pre-processing:

$$\mathrm{k}(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}\right) \tag{5}$$

where $\sigma$ is a kernel parameter.

In the conventional KPLS method, the input variable matrix $X_n$ can be replaced by matrix $K_n$:

$$K_n = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix} \tag{6}$$

### 2.3. Feature vector selection method for kernel tricks

As can be seen from Eq. (6), conventional kernel trick needs to compute the inner product among all vectors, and result in a matrix with size of $N \times N$, thus dimension disaster is inevitable when measurement data contain enormous samples. Meanwhile, data projected into a high dimensional space is in general mapped to some submanifold [19], therefore it is better to extract the data from these submanifold feature spaces. Baudat et al. [20] claimed that data matrix in a kernel space is not of full rank, some of the kernel vectors can be represented by other kernel vectors. They proposed an algorithm called feature vector selection (FVS) to extract these vectors from the feature space and reduce the size of kernel matrix. The principle of FVS can be briefly described as follows, for a kernel matrix as shown in Eq. (7):

$$K_i = \begin{bmatrix} K_{i-1} & \boldsymbol{k}_{i-1,i} \\ \boldsymbol{k}_{i,i-1} & k_{i,i} \end{bmatrix} \tag{7}$$

where $K_i$ denotes the kernel matrix of $X_i$ ($i < n$), $\boldsymbol{k}_{i\text{-}1,i}$ and $\boldsymbol{k}_{i,i\text{-}1}$ are a column vector and a row vector with $i$-1 elements respectively. When the rank of $K_i$ is equal to $i$-1, we can say that the additional part of $K_i$, in comparison with $K_{i\text{-}1}$, can be linearly described by $K_{i\text{-}1}$. Eq. (8) can be used to justify whether the kernel matrix $K_i$ is of full rank or not.

$$\delta_i = 1 - \frac{\boldsymbol{k}_{i-1,1}^T \boldsymbol{K}_{\boldsymbol{i}-1}^{-1} \boldsymbol{k}_{\boldsymbol{i}-1,\boldsymbol{i}}}{k_{i,i}} \tag{8}$$

Theoretically, $\delta_i$ should be equal to 0 in order to make $K_i$ a singular matrix, however, the real data collected from actual process is with noise generally. As a result $\delta_i$ could hardly be zero. So the threshold of $\delta_i$ need to be a non-zero positive number $\varepsilon$.

After acquiring matrix $X_n = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^{\mathrm{T}}$ that expanded and normalized from the original three-dimensional data array, the algorithm used in this paper can be briefly described in Table 2 (details can be referred to [18]). Matrix $S$ includes vectors selected from $X$ while kernel matrix $K_{span}$ can be calculated by equation $\mathrm{K}(X,S) = \langle f(X), f(S) \rangle$. We can use $K_{span}$ to replace the original kernel matrix $\mathrm{K}(X,X)$.

## 3. Reduced Dual Kernel MPLS algorithm

### 3.1. Principles of Reduced Dual Kernel MPLS algorithm

The MKPLS algorithm has shown great advantages on solving the nonlinear problem [6,14,15], but its basic principles have not changed a lot: (1) preprocess matrix $X$ and $Y$, (2) project input data $X$ into kernel matrix $K$, then (3) use $K$ and output data $Y$ to form a PLS model (as shown in Fig. 2b). These methods can handle the nonlinear problem of input data $X$ that the traditional PLS (Fig. 2a) cannot, but the output data $Y$ still remains unchanged, but the fact is that the output data $Y$ can also be transferred into kernel matrix $K_Y$ in the high dimensional space through the kernel trick which can be then used to construct the PLS model (as Fig. 2c) with kernel matrix $K_X$ (data $X$ in kernel