Chemometrics and Intelligent Laboratory Systems 96 (2009) 132–143

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

Moving window kernel PCA for adaptive monitoring of nonlinear processes

Xueqin Liu [a,b], Uwe Kruger [c,*], Tim Littler [a], Lei Xie [b,*], Shuqing Wang [b]

[a] School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Ashby Building, Stranmillis Road, Belfast BT9 5AH, UK
[b] State Key Lab of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, PR China
[c] Department of Electrical Engineering, The Petroleum Institute, P.O. Box 2533, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Article history:
Received 13 October 2008
Received in revised form 25 December 2008
Accepted 7 January 2009
Available online 23 January 2009

Keywords:
Nonlinear process
Kernel PCA
Moving window
Multivariate statistical process control
Adaptive
Numerically efficient

ABSTRACT

This paper discusses the monitoring of complex nonlinear and time-varying processes. Kernel principal component analysis (KPCA) has gained significant attention as a monitoring tool for nonlinear systems in recent years but relies on a fixed model that cannot be employed for time-varying systems. The contribution of this article is the development of a numerically efficient and memory saving moving window KPCA (MWKPCA) monitoring approach. The proposed technique incorporates an up- and downdating procedure to adapt (i) the data mean and covariance matrix in the feature space and (ii) approximates the eigenvalues and eigenvectors of the Gram matrix. The article shows that the proposed MWKPCA algorithm has a computation complexity of $O(N^2)$, whilst batch techniques, e.g. the Lanczos method, are of $O(N^3)$. Including the adaptation of the number of retained components and an l-step ahead application of the MWKPCA monitoring model, the paper finally demonstrates the utility of the proposed technique using a simulated nonlinear time-varying system and recorded data from an industrial distillation column.

1. Introduction

The demand for effective quality monitoring and safe operation in the petrochemical industry has propelled research into statistical-based fault detection and diagnosis methods over the past few decades. Multivariate statistical methods such as principal component analysis (PCA) [1,42,47], partial least squares [24,32,46] and more recently independent component analysis [10,27,30] have been developed and applied for this purpose. Among them, PCA is the most popular one, which relates to its conceptual simplicity. However, such methods collectively assume linear variable interrelationships, which hamper their application if these relationships are nonlinear.

Kramer [22] developed one of the first nonlinear extensions of PCA, which rely on auto-associative neural networks (ANNs). Reference [9] proposed a simplification that incorporated principal curves into this neural network structure. However, the nonlinear function is approximated by a linear combination of several univariate nonlinear functions, which represents a restriction of generality [19]. This was addressed by introducing input-training neural networks [19,38]. A detailed review in Reference [25] suggested that neural network based techniques may not represent a generic nonlinear extension of PCA.

The analysis in Reference [25] also yielded that KPCA [37] is a generic nonlinear PCA extension, which can efficiently compute PCs in a high-dimensional feature space using integral operators and nonlinear kernel functions. The core idea of KPCA is to first map the data space into a feature space using a nonlinear mapping and then compute the PCs in the feature space. It should also be noted that KPCA only requires the solution of an eigenvalue problem, and, since it can incorporate different kernel functions, KPCA can handle a wide range of nonlinearities. In addition, KPCA does not require a pre-estimate of the number of retained PCs.

Despite recently reported KPCA-based monitoring applications [8,16,26,45], the following problems arise: (i) the identification of a KPCA monitoring model requires the storage of the symmetric *kernel matrix*, whose dimension is given by the number of reference samples and (ii) the monitoring model is fixed which may produce false alarms if the process is naturally time-varying. The latter problem has been addressed by a recursive KPCA formulation [5,21,44], similar in approach to the work on linear recursive PCA [29,43]. However, the kernel matrix grows in size each time a new data point becomes available which is practically problematic (memory and computational requirements).

Implementing a moving window approach, as proposed by Hoegaerts et al. [17], overcomes this problem and produces a constant size of the kernel matrix and a constant speed of adaptation [42]. The adaptation of the kernel matrix relies on a simultaneous up- and downdating, which is memory efficient compared to a recalculation of the kernel matrix. It is important to note, however, that alterations in the sample mean of the original as well as the transformed variable set in the feature space were not considered in [17]. This, however, can

* Corresponding authors. Kruger is to be contacted at Department of Electrical Engineering, The Petroleum Institute, P.O. Box 2533, Abu Dhabi, United Arab Emirates. Xie, State Key Lab of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, PR China. Tel.: +971 2607 5150; fax: +971 2607 5200.
E-mail addresses: ukruger@pi.ac.ae (U. Kruger), leix@iipc.zju.edu.cn (L. Xie).

0169-7439/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.chemolab.2009.01.002

reduce the sensitivity for detecting fault conditions as discussed in Reference [44]. An adaptive monitoring scheme is therefore required for the mean of the transformed variable set in the feature space and for incorporating this change into the kernel matrix prior to the adaptation of the KPCA model.

This article proposes the adaptation of the mean and covariance matrix in the feature space and addresses the following issues, which the literature has not considered yet: (i) how to adapt the eigendecomposition of the Gram matrix numerically efficient, (ii) how to determine changes in the number of retained PCs, (iii) how to adapt the statistical confidence limits based on the adapted KPCA model and (iv) how to prevent the KPCA model from adapting incipiently developing faults. Compared to conventional methods, such as Rank-1 modification [3,11], inverse iteration [12] or the Lanzcos method [34,35] which this article shows to be of $O(N^3)$, the proposed method is of $O(N^2)$.

The paper is organized as follows. Preliminaries of KPCA-based monitoring are presented prior to the adaptation of the mean and covariance matrix in the feature space. Section 4 then presents the adaptation of the MWKPCA model. This is followed by summarizing the adaptation of the number of retained PCs and the univariate monitoring statistics. Next, a simulation example (Section 6) and the analysis of recorded data from an industrial distillation unit (Section 7) are given to demonstrate the effectiveness of the MWKPCA technique. Finally, Section 8 provides a concluding summary of this article.

2. Preliminaries

KPCA maps a set of M observations $\mathbf{x}\in\mathbb{R}^n$, n and $M\in\mathbb{N}$, $M>n$, into a high-dimensional feature space $\boldsymbol{\Phi}(\mathbf{x})\in F$ and subsequently performs a PCA on $\boldsymbol{\Phi}(\mathbf{x})$. Let $\mathbf{x}_i$ be the ith sample, the covariance matrix for $\boldsymbol{\Phi}(\mathbf{x})$ is:

$$\mathbf{C}_\Phi = \frac{1}{M-1}\sum_{i=1}^{M}(\boldsymbol{\Phi}(\mathbf{x}_i)-\mathbf{m}_\Phi)(\boldsymbol{\Phi}(\mathbf{x}_i)-\mathbf{m}_\Phi)^T = \frac{1}{M-1}\overline{\boldsymbol{\Phi}}(\mathbf{X})\overline{\boldsymbol{\Phi}}T(\mathbf{X}), \quad (1)$$

where $\mathbf{m}_\Phi = \frac{1}{M}\boldsymbol{\Phi}(\mathbf{X})1_M$ is the sample mean in the feature space, 1_M is an M-dimensional vector of ones, $\boldsymbol{\Phi}(\mathbf{X})=[\boldsymbol{\Phi}(\mathbf{x}_1),\boldsymbol{\Phi}(\mathbf{x}_2),\ldots,\boldsymbol{\Phi}(\mathbf{x}_M)]$, $\overline{\boldsymbol{\Phi}}(\mathbf{X}) = \boldsymbol{\Phi}(\mathbf{X}) - \frac{1}{M}\boldsymbol{\Phi}(\mathbf{X})\mathbf{E}_M$, $\mathbf{E}_M = 1_M\times 1_M^T$, is the mean centered feature matrix and $\mathbf{X}=[\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_M]$. Next, an eigendecomposition of $\mathbf{C}_\Phi$ is computed:

$$\mathbf{C}_\Phi\mathbf{u}_k = \frac{1}{M-1}\overline{\boldsymbol{\Phi}}(\mathbf{X})\overline{\boldsymbol{\Phi}}^T(\mathbf{X})\mathbf{u}_k = \lambda_k\mathbf{u}_k \qquad k=1,2,\ldots,M, \quad (2)$$

where λ_k and $\mathbf{u}_k$ represent the kth eigenvalue-eigenvector pair of $\mathbf{C}_\Phi$. Given that the explicit mapping function $\boldsymbol{\Phi}(\mathbf{x})$ is unknown, KPCA circumvents the use of $\boldsymbol{\Phi}(\mathbf{x})$ by utilizing the eigendecomposition of the centered Gram matrix $\mathbf{G}=\overline{\boldsymbol{\Phi}}^T(\mathbf{X})\overline{\boldsymbol{\Phi}}(\mathbf{X})\in\mathbb{R}^{M\times M}$:

$$\overline{\boldsymbol{\Phi}}^T(\mathbf{X})\overline{\boldsymbol{\Phi}}(\mathbf{X})\mathbf{v}_k = \zeta_k\mathbf{v}_k, \quad (3)$$

where $\zeta_k\in\mathbb{R}$ and $\mathbf{v}_k\in\mathbb{R}^M$ represent the kth eigenvalue-eigenvector pair of $\mathbf{G}$. Introducing the kernel definition $K(\mathbf{x}_i,\mathbf{x}_j)=\boldsymbol{\Phi}^T(\mathbf{x}_i)\boldsymbol{\Phi}(\mathbf{x}_j)$, $\mathbf{G}$ can be computed from the kernel matrix $\mathbf{K}=\boldsymbol{\Phi}^T(\mathbf{X})\boldsymbol{\Phi}(\mathbf{X})\in\mathbb{R}^{M\times M}$

$$\mathbf{G} = \mathbf{K} - \frac{1}{M}\mathbf{K}\mathbf{E}_M - \frac{1}{M}\mathbf{E}_M\mathbf{K} + \frac{1}{M^2}\mathbf{E}_M\mathbf{K}\mathbf{E}_M. \quad (4)$$

After constructing the PCA model in the feature space, the KPCA score vector, $\mathbf{t}\in\mathbb{R}^r$, for a new sample $\mathbf{x}\notin\mathbf{X}$ is given by:

$$t = \mathbf{U}^T\overline{\boldsymbol{\Phi}}(\mathbf{x}) = \mathbf{A}^T\boldsymbol{\Phi}^T(\mathbf{X})\left(\boldsymbol{\Phi}(\mathbf{x}) - \frac{1}{M}\boldsymbol{\Phi}(\mathbf{X})1_M\right) = \mathbf{A}^T\left(\mathbf{k}(\mathbf{X},\mathbf{x}) - \frac{1}{M}\mathbf{K}1_M\right). \quad (5)$$

Here, $\overline{\boldsymbol{\Phi}}(\mathbf{x})=\boldsymbol{\Phi}(\mathbf{x})-\mathbf{m}_\Phi$, r is the number of retained PCs, $\mathbf{U}=[\mathbf{u}_1\,\mathbf{u}_2\ldots\mathbf{u}_r]\in F$, $\mathbf{A}=\left[\mathbf{I}-\frac{1}{M}\mathbf{E}_M\right]\mathbf{V}\in\mathbb{R}^{M\times r}$, $\mathbf{V}=\left[\frac{\mathbf{v}_1}{\sqrt{\zeta_1}}\frac{\mathbf{v}_2}{\sqrt{\zeta_2}}\cdots\frac{\mathbf{v}_r}{\sqrt{\zeta_r}}\right]\in\mathbb{R}^{M\times r}$ and $\mathbf{k}(\mathbf{X},\mathbf{x})\in\mathbb{R}^M$ represents the kernel vector constructed from $\mathbf{X}$ and $\mathbf{x}$:

$$\mathbf{k}(\mathbf{X},\mathbf{x}) = \begin{pmatrix}K(\mathbf{x}_1,\mathbf{x}) & K(\mathbf{x}_2,\mathbf{x}) & \cdots & K(\mathbf{x}_M,\mathbf{x})\end{pmatrix}^T. \quad (6)$$

For process monitoring, KPCA relies on Hotelling's T^2 and Q statistics:

$$T^2 = \mathbf{t}^T\boldsymbol{\Lambda}^{-1}\mathbf{t} \qquad Q = \overline{\boldsymbol{\Phi}}(\mathbf{x})^T\left[\mathbf{I}-\mathbf{U}\mathbf{U}^T\right]\overline{\boldsymbol{\Phi}}(\mathbf{x}), \quad (7)$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix storing the variances of the score variables and $Q = K(\mathbf{x},\mathbf{x}) - \frac{2}{M}1_M^T\mathbf{k}(\mathbf{X},\mathbf{x}) + \frac{1}{M^2}1_M^T\mathbf{K}1_M - \mathbf{t}^T\mathbf{t}$.

3. Adaptation of mean and covariance matrix

This section describes the first contribution of this article and relates to the adaptation of the mean and covariance matrix in the feature space. Conventional moving window PCA (MWPCA) work [42] utilizes a two-step procedure that involves removing the oldest sample, further referred to as downdating, and then adding the newly available sample, defined as updating. This is followed by recomputing the PCA decomposition including the adapted variable mean and covariance matrix in a numerically efficient way, e.g. by applying inverse iteration or Lanczos method [29,42].

For the transformed data in the feature matrix, $\boldsymbol{\Phi}(\mathbf{X})$, applying the same two-step procedure involves the utilization of an intermediate window, which discards the effect of the oldest sample, and the "new" window that includes the impact of the newly available sample. Setting the window length to $N\in\mathbb{N}$ and defining the feature matrices that stores the transformed data of the intermediate window and the new window by $\boldsymbol{\Phi}(\widetilde{\mathbf{X}})$ and $\boldsymbol{\Phi}(\widehat{\mathbf{X}})$, respectively, the adaptation of the mean and covariance matrix of the transform variable set conceptually relies on the following procedure: $\boldsymbol{\Phi}(\mathbf{X})\Rightarrow\boldsymbol{\Phi}(\widetilde{\mathbf{X}})\Rightarrow\boldsymbol{\Phi}(\widehat{\mathbf{X}})$, where $\boldsymbol{\Phi}(\widetilde{\mathbf{X}})=[\boldsymbol{\Phi}(\mathbf{x}_2),\cdots,\boldsymbol{\Phi}(\mathbf{x}_N)]$, $\boldsymbol{\Phi}(\widehat{\mathbf{X}})=[\boldsymbol{\Phi}(\mathbf{x}_2),\cdots,\boldsymbol{\Phi}(\mathbf{x}_N),\boldsymbol{\Phi}(\mathbf{x}_{N+1})]$. Here, $\widetilde{\mathbf{X}}=[\mathbf{x}_2\,\mathbf{x}_3\cdots\mathbf{x}_N]$, $\widehat{\mathbf{X}}=[\mathbf{x}_2\cdots\mathbf{x}_N\,\mathbf{x}_{N+1}]$, and $\boldsymbol{\Phi}(\mathbf{x}_{N+1})$ is the newly recorded sample transformed into the feature space.

As shown in Eq. (1), a KPCA model is constructed from the covariance matrix of the process data transformed into the feature space. Its adaptation requires the adaptation of the mean vector, $\mathbf{m}_\Phi$, and the covariance matrix, $\mathbf{C}_\Phi$, by the following two-step procedure [42]. It should be noted that the development of the adaptation algorithms for the mean, covariance matrix and the subsequent KPCA model relies on the first shift of the moving window, that is when the first "new" sample, $\mathbf{x}_{N+1}$, becomes available. This is for convenience only and does not represent a restriction of generality.

Step 1. *Downdating* ($\boldsymbol{\Phi}(\mathbf{X})\rightarrow\boldsymbol{\Phi}(\widetilde{\mathbf{X}})$): The mean vector in the feature space of the intermediate window, $\widetilde{\mathbf{m}}_\Phi$, can be expressed by that of the "old" window, $\mathbf{m}_\Phi$, and the removal of the impact of the oldest sample, $\boldsymbol{\Phi}(\mathbf{x}_1)$:

$$\widetilde{\mathbf{m}}_\Phi = \frac{N}{N-1\mathbf{m}_\Phi - \frac{1}{N-1}\boldsymbol{\Phi}(\mathbf{x}_1)} \quad (8)$$

Incorporating Eq. (8) into the definition of the covariance matrix gives rise to:

$$\widetilde{\mathbf{C}}_\Phi = \frac{N-1}{N-2}\left[\mathbf{C}_\Phi - \frac{N}{(N-1)^2}(\boldsymbol{\Phi}(\mathbf{x}_1)-\mathbf{m}_\Phi)(\boldsymbol{\Phi}(\mathbf{x}_1)-\mathbf{m}_\Phi)^T\right]. \quad (9)$$

Step 2. *Updating* $\boldsymbol{\Phi}(\widetilde{\mathbf{X}})\rightarrow\boldsymbol{\Phi}(\widehat{\mathbf{X}})$: The mean vector in the feature space of the new window, $\widehat{\mathbf{m}}_\Phi$, can be computed from the intermediate window matrix, $\widetilde{\mathbf{m}}_\Phi$, and the new observations, $\boldsymbol{\Phi}(\mathbf{x}_{N+1})$:

$$\widehat{\mathrm{m}}_\Phi = \frac{N-1}{N}\widetilde{m}_\Phi + \frac{1}{N}\boldsymbol{\Phi}(\mathbf{x}_{N+1}) \quad (10)$$

Using the above equation, the covariance matrix of the transformed data in the new window becomes:

$$\widehat{\mathbf{C}}_\Phi = \frac{1}{N-1}\overline{\boldsymbol{\Phi}}(\widehat{\mathbf{X}})\overline{\boldsymbol{\Phi}}(\widehat{\mathbf{X}})^T = \frac{N-2}{N-1}\widetilde{\mathbf{C}}_{\boldsymbol{\Phi}} + \frac{1}{N}(\boldsymbol{\Phi}(\mathbf{x}_{N+1})-\widetilde{\mathbf{m}}_\Phi)(\boldsymbol{\Phi}(\mathbf{x}_{N+1})-\widetilde{\mathbf{m}}_\Phi)^T,$$