Simultaneous Estimation of the Number of Principal Components and Kernel Parameter in KPCA*

Yujia Fu, Hongfeng Tao and Huizhong Yang

Abstract—This article proposed a novel method to determine the number of principal components and the optimal values of tuning factors for kernel principal component models. Existing work predominantly relies on ad-hoc rules or cross-validated approaches to estimate. To guarantee statistical independence, the proposed technique incorporates a two-fold cross-validated approach by omitting one variable in turn, which is predicted by the remaining ones. For these regressions, the number of principal components varies. This finally yields an optimum selection for the parameters, which application and the analysis of recorded industrial data from a glass melter process confirm.

Index Terms—kernel principal component analysis, the number of principal components, cross validations

I. INTRODUCTION

Kernel principal component analysis (KPCA) [1] is a widely used nonlinear method as the nonlinear extension of principal component analysis (PCA) in control engineering applications, including process modeling [2], process monitoring [3], fault detection [4], feature extraction [5] and dimensionality reduction [6].

To develop a KPCA model, it is necessary to select the optimal number of principal components (PCs) which is the dimension of the KPCA model. If the model includes more than the required number of PCs, it is over-parameterized and may incorporate information that is not related to the source or common cause variables. In contrast, containing too few components results in a KPCA model that cannot completely represent the information encapsulated in the common cause variation.

To determine the number of PCs $n$, the proposed methods in the literature includes the cumulative percent variance (CPV) [7], which measures the variance percentage explained by the first $n$ PCs. The performance of this approach, however, is strongly relied on the selection of the threshold. Another technique is kernel parallel analysis (KPA) [8], which is an automated technique and an kernel extension of parallel analysis for PCA. More recently, a technique that is based on the reconstruction error (RE) was proposed [9]. This technique determines $n$ such that the residual error is sufficiently small, which is also subjective.

*This work was supported by National Natural Science Foundation of China (No.61273070), 111 project (No.B12018), the Fundamental Research Funds for the Central Universities (JUSRP1733B) and Jiangsu University Graduate Innovation Research Project (No.KY15-1168).

The authors are with Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education, Jiangnan University, Wuxi 214122, P. R. China fuyujialw@126.com(Y.J. Fu); taohongfeng@hotmail.com(H.F. Tao); yhz@jiangnan.edu.cn(H.Z. Yang)

Besides the number of PCs, the parameter of the kernel function $\sigma$ can strongly affect the performance of a KPCA model. Despite this, the literature has only proposed ad hoc and heuristic methods to select an appropriate value for this parameter. Associate methods include the Kernel target alignment (KTA) [10], which was developed as a kernel matrix evaluation criteria, is used to measure the degree of linear dependency between the kernel matrix and target. A different evaluation criteria is the feature space-based kernel matrix evaluation measure (FSM) [11]. However, both KTA and FSM may fail to determine an appropriate kernel parameter for small sample sizes and have a tendency of overfitting [12]. To address this, Yang et al. [12] developed a technique based on the largest variance criteria for the optimal kernel parameter. Kim et al. [13] proposed a method which selects the kernel parameter as the product of variable number, variance of the variables, the number of variables and a proper constant which varies between 1 and 10. However, again, it is subjective. Kenig et al. [14] suggested using the 0.2 quantile of the distances between the samples as a guide to select $\sigma$. Teixeira et al. [15] proposed the solution by calculating the maximum distance between each sample and the mean of the data set. Ni et al. [16] selected the kernel parameters as the sum of the difference of the maximum and minimum for each variable in the data matrix.

Despite the recent advances in selecting $n$ and $\sigma$, the literature has only proposed ad hoc and heuristic methods and the selection of both parameters is still an open question. Moreover, the literature has not proposed a simultaneous estimation of both parameters in an optimal fashion. This is the main objective of this article, which introduces a simultaneous estimation of both based on the variance of prediction error.

The article is divided into the following sections. The next section presents a brief summary of KPCA. Section III introduces the working on the proposed method for linear PCA. Section IV then develops this method to be applicable to KPCA and gives a theoretical foundation for the working of the proposed method. Finally, Section V presents an industrial case study to demonstrate the working and accuracy of the proposed technique and Section VII provides a concluding summary.

II. PRELIMINARIES

Typical noncausal data structures for a set of recorded process variables stored in a random vector $x \in \mathbb{R}^N$ are:
Here, \( s \in \mathbb{R}^n, n < N \), is a random vector storing \( n \) common cause variables that describe the dominant variation in \( x \). \( e \) is a random error vector describing measurement noise and other random variation that is not encapsulated in the source variable set \( s \). Eqs (1a) and (1b) describe linear and nonlinear relationships between \( s \) and \( x \), respectively, where \( A \) is a parameter matrix and \( \psi(\cdot) \) is a smooth nonlinear function.

Provided that \( E[\{a_i^T s\}^2] \ll E[\{e_i^2\}] \) for \( i = 1, \ldots, N \), where \( a_i^T \) is the \( i \)th row vector of \( A \) and \( E[\cdot] \) the expectation operator, the eigendecomposition of the data covariance matrix reveals \( n \) for Eq (1a). The aim of this article is to introduce a novel method to simultaneously estimate \( n \) and kernel parameter for KPCA, a generic nonlinear extension of PCA to obtain nonlinear principal components with respect to Eq (1b).

KPCA first maps the random vector \( x \) into a high-dimensional feature space \( \phi(x) \), for a total of \( L \) recorded samples. Following from the properties of reproducing kernels, the elements of \( K \) can be expressed by various kernel functions, for example the Gaussian kernel function that is of the form:

\[
K_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{\sigma^2} \right). 
\] (7)

For the finite sample size, \( L \), the accuracy of expressing the scalar product terms depends on the selection of \( \sigma \). As highlighted in Section I, the literature has, thus far, only reported heuristic and ad hoc techniques to determine \( l \) and \( \sigma \). In addition to that, the work reviewed in Section I is designed to either estimate \( l \) or \( \sigma \) but not simultaneously. Moreover, most techniques do not rely on cross validatory schemes that allows an identification of the KPCA model and an independent validation of the performance of the identified KPCA model.

### III. Concept of estimating \( n \) for PCA

This section shows the working of the proposed two-fold cross-validatory method for PCA. It is not intended, however, to introduce a new technique to estimate \( n \) for Eq (1a), as this is a well-researched topic. The proposed method relies on segmenting \( \tilde{x} \) and \( \tilde{X} \), which Subsections III-A and III-B describe, respectively. Subsection III-C introduces the two-fold cross-validatory method and Subsection III-D shows a simulation example.

#### A. Segmenting the variable set \( x \)

The first step is removing one variable from \( \tilde{x} \) and utilizing the remaining variables to predict the removed one. Removing the \( i \)th variable reduces Eq (1a) to become:

\[
\begin{align*}
\bar{\tilde{x}}_1 &= \begin{bmatrix} \tilde{x}_1 \\
\vdots \\
\tilde{x}_{i-1} \\
\tilde{x}_{i+1} \\
\vdots \\
\tilde{x}_N \end{bmatrix} \\
\bar{\tilde{x}}_i &= \begin{bmatrix} a_{i-1}^T \\
a_{i+1}^T \\
\vdots \\
a_N^T \end{bmatrix} s + \begin{bmatrix} e_{i-1} \\
e_{i+1} \\
\vdots \\
e_N \end{bmatrix}.
\end{align*}
\] (8)

By denoting \( \bar{\tilde{x}}_i = (\tilde{x}_1, \cdots, \tilde{x}_{i-1}, \tilde{x}_{i+1}, \cdots, \tilde{x}_N) \), \( A_{i} = [ a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_N] \) and \( e_{i} = (e_1, \cdots, e_{i-1}, e_{i+1}, \cdots, e_N) \), we have:

\[
\bar{\tilde{x}}_i = A_{i} s + e_{i}.
\] (9)

The data structure for \( i \)th variable is then:

\[
\bar{\tilde{x}}_i = a_i^T s + e_i.
\] (10)
B. Segmenting the set of samples stored in \( \mathbf{X} \)

The second step is segmenting the recorded set of samples. This guarantees evaluating the performance of the PCA model employing different samples to those used to identify the model and requires partitioning \( \mathbf{X} \) as follows:

\[
\mathbf{X} = \begin{bmatrix}
\xi_1^{(1)} & \ldots & \xi_1^{(m)} \\
\xi_2^{(1)} & \ldots & \xi_2^{(m)} \\
\vdots & \ddots & \vdots \\
\xi_N^{(1)} & \ldots & \xi_N^{(m)} \\
\end{bmatrix}, \quad (11)
\]

where \( m \) is the number of segments. Moreover, \( p = L/m \) denotes the number of samples in each segment. Each vector \( \xi_j^{(i)} \) contains a number of samples of the \( i \)th variable, i.e. \( \xi_j^{(i)} \) stores the samples \( \mathbf{x}_j^{(i-1)p+1}, \mathbf{x}_j^{(i-1)p+2}, \ldots, \mathbf{x}_j^{(i)p} \).

The two-fold cross-validatory method requires defining the following matrices based on Eq (11):

\[
\mathbf{X}^{(-j)} = \begin{bmatrix}
\xi_1^{(1)} & \ldots & \xi_1^{(m-1)} & \xi_1^{(m)} \\
\xi_2^{(1)} & \ldots & \xi_2^{(m-1)} & \xi_2^{(m)} \\
\vdots & \ddots & \vdots & \vdots \\
\xi_N^{(1)} & \ldots & \xi_N^{(m-1)} & \xi_N^{(m)} \\
\end{bmatrix}
\]

\[
\mathbf{x}_{-i}^{(-j)} = \begin{bmatrix}
\xi_1^{(j)} \\
\xi_2^{(j)} \\
\vdots \\
\xi_N^{(j)} \\
\end{bmatrix}, \quad (12a)
\]

\[
\mathbf{p}_{-i}^{(-j)} = \begin{bmatrix}
\xi_1^{(j-1)} \\
\xi_2^{(j-1)} \\
\vdots \\
\xi_N^{(j-1)} \\
\end{bmatrix}, \quad (12b)
\]

\[
\mathbf{x}_{-i}^{(j)} = \begin{bmatrix}
\xi_1^{(j)} & \xi_2^{(j)} & \ldots & \xi_N^{(j)} \\
\end{bmatrix}, \quad (12c)
\]

\[
\mathbf{p}_{-i}^{(j)} = \xi_i^{(j)}, \quad (12d)
\]

C. Proposed two-fold cross-validatory method

Using Eq (12a) allows:

- estimating the covariance matrix of the variable set \( \mathbf{x}_{-i}^{(-j)} \):
  \[
  \hat{\mathbf{S}}_{\mathbf{x}_{-i}^{(-j)}} = \frac{1}{(m-1)p-1} \mathbf{X}^{(-j)T} \mathbf{X}^{(-j)} \quad (13)
  \]
- computing the eigendecomposition of \( \hat{\mathbf{S}}_{\mathbf{x}_{-i}^{(-j)}} \):
  \[
  \hat{\mathbf{S}}_{\mathbf{x}_{-i}^{(-j)}} = \hat{\mathbf{P}}_{-i} \hat{\mathbf{A}}_{-i} \hat{\mathbf{P}}_{-i}^T
  \]
  \[
  \hat{\mathbf{P}}_{-i} = \begin{bmatrix}
  \hat{p}_1 & \hat{p}_2 & \cdots & \hat{p}_{N-1}
  \end{bmatrix}_{-i}
  \]
- and calculating the first \( \ell \) principal component scores:
  \[
  \begin{bmatrix}
  \hat{\tau}_1 \\
  \hat{\tau}_2 \\
  \vdots \\
  \hat{\tau}_\ell_{-i}
  \end{bmatrix}^{(-j)} = \mathbf{X}^{(-j)} \hat{\mathbf{P}}_{-i}
  \]

The scores are now used to setup a regression equation, where the scores stored in \( \mathbf{\hat{T}}^{(-j)}_{-i} = [\hat{\tau}_1, \hat{\tau}_2, \ldots, \hat{\tau}_\ell_{-i}] \in \mathbb{R}^{(m-1)p \times \ell} \) represent samples of the predictor variable set and the set stored in \( \mathbf{\hat{p}}^{(-j)}_{-i} \) are samples of the response set. The unknown model parameters can be computed as follows:

\[
\hat{\beta}^{(-j)}_{-i} = \left[\mathbf{\hat{T}}^{(-j)T}_{-i} \mathbf{\hat{T}}^{(-j)}_{-i}\right]^{-1} \mathbf{\hat{T}}^{(-j)T}_{-i} \mathbf{\hat{p}}^{(-j)}_{-i}.
\]

(16)

The identified regression model is now applied to the samples stored in \( \mathbf{\hat{X}}^{(-j)}_{-i} \) and \( \mathbf{\hat{p}}^{(j)}_{-i} \) — defined in Eqs (12c) and (12d):

\[
\hat{\mathbf{x}}^{(-j)}_{-i} = \left[\hat{\mathbf{\hat{T}}}_{-i}^{(-j)} \hat{\beta}^{(-j)}_{-i}\right] \mathbf{\hat{p}}^{(-j)}_{-i} = \mathbf{\hat{p}}^{(-j)}_{-i} - \hat{x}^{(-j)}_{-i}.
\]

(17)

Recall that the segmentation of \( \mathbf{X} \), shown in Eq (12), allows a statistically independent performance evaluation of the regression model, as the samples stored in \( \mathbf{\hat{X}}^{(-j)}_{-i} \) and \( \mathbf{\hat{p}}^{(j)}_{-i} \) have not been used to identify the regression model. Using a cross-validatory approach, the variance of the prediction error for predicting the \( \ell \)th random variable is:

\[
\hat{\sigma}^2_{\epsilon_{-i}}(\ell) = \frac{1}{\ell} \sum_{j=1}^{m} \epsilon^{(j)}_i \epsilon^{(j)}_i.
\]

(18)

As the regression model relies on retaining \( \ell \) principal components, so does the variance of the prediction error. The cross-validatory approach based on the variable segmentation in Eqs (9) and (10) yields the following average variance:

\[
\hat{\sigma}^2_{\epsilon_{-i}}(\ell) = \frac{1}{\ell} \sum_{i=1}^{N} \hat{\sigma}^2_{\epsilon_{-i}}(\ell) = \frac{1}{\ell} \sum_{i=1}^{N} \sum_{j=1}^{m} \epsilon^{(j)}_i \epsilon^{(j)}_i.
\]

(19)

The above statistic can be evaluated for \( \ell = 1, \ldots, N-1 \).

D. Simulation study

The following example shows that the two-fold cross-validatory concept can correctly estimate \( n \) in PCA. The simulation is based on Eq (1a) for \( N = 10, n = 1 \) to 3, \( e \sim N(0, 0.05I) \), \( s \sim N(0, I) \) and \( \mathbf{A} \) containing randomly selected elements between -1 and 1 that were drawn from a uniform distribution.

Table I summarizes the results of the proposed method for selecting \( \ell \) from 1 to 5, \( m = L \Rightarrow p = 1 \) leave-one-out cross-validation, and for various sample sizes, including \( L = 500, 1000, 2000, 5000 \) and 10000.

Table I confirms that the proposed method correctly estimate \( n \), irrespective of \( L \). For the proposed method, it is also interesting to note that as the number of samples increases, the estimated value \( \hat{\sigma}^2_{\epsilon_{-i}}(\ell) \) changes less and less significant for \( \ell = n, n+1, n+2 \) etc. Asymptotically, \( \hat{\sigma}^2_{\epsilon_{-i}} \), \( \epsilon_i \sim \pi_i - \beta_{\pi_i} - t_{-i} \), is given by:

\[
\hat{\sigma}^2_{\epsilon_{-i}} = \sigma^2_{\epsilon_{-i}} = s_{x_{-i}} - s_{x_{-i} x_{-i}} \mathbf{P}_{-i}^{-1} \mathbf{P}_{-i}^T s_{x_{-i}},
\]

(20)

Here, the covariances are \( s_{x_{-i} x_{-i}} = s_{x_{-i} x_{-i}} - \mathbf{A}_{-i} \mathbf{A}_{-i}^T \) and \( S_{x_{-i}} = \mathbf{A}_{-i} \mathbf{A}_{-i}^T + \sigma^2 \mathbf{I} \). Moreover, \( \hat{\sigma}^2_{\epsilon_{-i}} \) is identical to the least squares solution for \( \pi_i = \gamma_{\pi_i} \mathbf{P}_{-i} + \epsilon_i \), i.e. \( \hat{\sigma}^2_{\epsilon_{-i}} = \sigma^2_{\epsilon_{-i}} = s_{x_{-i} x_{-i}} \mathbf{P}_{-i}^{-1} s_{x_{-i} x_{-i}}, \) which follows from the fact that
the random vector $s$, describing the common cause variation in $x$, is encapsulated within the first $n$ retained components:

$$t_{-i} = P^T_{-i} (A_{-i} s + e_{-i}) \Rightarrow s = [P^T_{-i} A_{-i}]^{-1} t_{-i} - P^T_{-i} e_{-i}$$

(21)

Adding the discarded $N - n$ components to $t_{-i}$ incorporates uncorrelated information, as any additional eigenvectors are orthogonal to the column space spanned by $A_{-i}$. Asymptotically, it follows that $\sigma^2_{\ell}$ for $\ell = n, n + 1, \ldots, N - 1$ is given by Eq. (20), i.e. $\sigma^2_{\ell} = 1/N \sum_{i=1}^{\ell} \sigma^2_{\ell}$. The estimate for $n$ is then the smallest $\ell$ for which Eq. (19) asymptotically converges to $\sigma^2_{\ell}$. It is also important to note that although $\sigma_{\ell}^2(n) \rightarrow \sigma_{\ell}^2$, as $L \rightarrow \infty$, Eq. (19) is a biased estimate, as it is not based on the correct number of degrees of freedom. A detailed discussion of this issue, however, is beyond the scope here.

IV. SIMULTANEOUS ESTIMATION OF KPCA PARAMETERS

Using the two-fold cross-validatory approach, introduced in Section III, this section develops a method to optimally determine $n$ and $\sigma$. This method relies on the discussion in Ref [1]: "to determine a reconstruction mapping requires a suitable regression method for estimating the reconstruction mapping from kernel-based principal components to the input", which forms the basis for kernel partial least squares (KPLS) [13] and kernel principal component regression (KPCR) [17].

Following from Eqs (9) and (10), partitioning Eq (1b) by removing the $i$th variable yields:

$$\bar{X}_i = \psi_i(s) + e_i$$

(22a)

$$\bar{X}_{-i} = \psi_{-i}(s) + e_{-i}$$

(22b)

Following from the properties of reproducing kernels, the nonlinear score variables contain the information of the common cause variables $s$. Hence, it is possible to apply the approach detailed in Section III-C for PCA also for KPCA.

This, however, requires the use of KPCR:

$$\bar{x}_i^{(-j)} = \bar{\phi}_i(\bar{X}_{-i}^{(-j)}) \beta_i^{(-j)} + \epsilon_i^{(-j)}$$

(23)

Note that based on the definition $F = \bar{\phi}(X)$ in Section II, $\bar{\phi}_i(\bar{X}_{-i}^{(-j)}) = F_{-i}^{(-j)}$. For Eq (23), the nonlinear transformation $\bar{\phi}_i(\bar{X}_{-i}^{(-j)})$ is usually not known. To overcome this, KPCR relies on computing principal components from the Gram matrix that utilizes scalar products between feature vectors. This yields:

$$\bar{x}_i^{(-j)} = B_{-i}^{(-j)} \omega^{(-j)} + e_i^{(-j)}$$

(24a)

$$B_{-i}^{(-j)} = F_{-i}^{(-j)} U_{-i}^{(-j)}$$

(24b)

Here, $B_{-i}^{(-j)}$ is a matrix of regressors and $U_{-i}^{(-j)}$ stores the eigenvectors of $S_{-i}^{(-j)} = (1/L) F_{-i}^{(-j)T} F_{-i}^{(-j)}$. By comparing Eqs (23) and (24), it follows that $\beta_i^{(-j)} = U_{-i}^{(-j)T} \omega_i^{(-j)}$.

Appendix shows how $\epsilon_i^{(-j)}$ depends on the number of retained components $\ell$ and the set of kernel parameters $\sigma^2$:

$$\epsilon_i^{(-j)} = \bar{x}_i^{(-j)T} - B_{-i}^{(-j)} \omega_i^{(-j)}$$

$$= \bar{x}_i^{(-j)T} - G_{-i}^{(-j)} V_{-i}^{(-j)} S_{-i}^{(-j)-1} \omega_i^{(-j)}$$

(25)

To guarantee a statistically independent evaluation of the regression model, it is applied to the samples of the $j$th segment of $X$, defined in Eqs (12c) and (12d), which yields:

$$\epsilon_i^{(j)} = \bar{x}_i^{(j)} - B_j^{(-j)} \omega_i^{(-j)} = \bar{x}_i^{(j)} - G_j^{(-j)} V_{-i}^{(-j)} S_{-i}^{(-j)-1} \omega_i^{(-j)}$$

(27)

Eq (27) yields an estimate of $\sigma^{2}_{\epsilon}$. The average variance over all $N$ variables is then given in Eq. (19). Different to Eq (17), however, the residual vector $\epsilon_i^{(j)}$ depends not only on $\ell$ but also on $\sigma$. For KPCA, Eq (18) and (19) become:

$$\sigma^{2}_{\epsilon}(\ell, \sigma) = \frac{1}{\ell} \sum_{i=1}^{m} \epsilon_i^{(j)T} \epsilon_i^{(j)}$$

(28a)

$$\sigma^{2}_{\epsilon}(\ell, \sigma) = \frac{1}{\ell} \sum_{i=1}^{N} \sigma^{2}_{\epsilon}(\ell, \sigma)$$

(28b)

V. ALGORITHM FOR DETERMINING KPCA PARAMETERS

According to Eq (1b), the variance of each random variable in $e$ is significantly smaller than the variance of any random variable in $\psi(s)$. This, in turn, implies that $\sigma^2_{\epsilon}(\ell < n, \sigma) > \sigma^2_{\epsilon}(\ell = n, \sigma)$, as the $\ell < n$ retained nonlinear PCs cannot describe the information of the random vector $s$. As for PCA, we can asymptotically conclude that:

$$\lim_{L \rightarrow \infty} \sigma^{2}_{\epsilon}(\ell < n, \sigma) > \lim_{L \rightarrow \infty} \sigma^{2}_{\epsilon}(\ell = n, \sigma) = \lim_{L \rightarrow \infty} \sigma^{2}_{\epsilon}(\ell > n, \sigma)$$

(29)
Algorithm 1 shows the steps of the proposed technique, which determines the parameters of the KPCA model, i.e. \( n \) and \( \sigma \), in an iterative fashion.

Algorithm 1 Simultaneous estimation of KPCA parameters
1: Set values for initial parameters \( \ell \) and \( \sigma \)
2: Set \( i = 1, j = 1 \), define \( m \) and \( p = L/m \)
3: Set \( \hat{\sigma}^2_\ell = 0 \) and all \( \hat{\sigma}^2_\ell(i) = 0 \)
4: While \( i \leq N \)
5: Set \( J = 0 \)
6: While \( j \leq m \)
7: Setup \( \mathbf{X}_{-i}^{(j)} \), \( \mathbf{x}_{-i}^{(j)} \), \( \mathbf{X}_i^{(j)} \) and \( \mathbf{x}_i^{(j)} \)
8: Construct Gram matrices \( G_{-i}^{(j)} \) and \( G_i^{(j)} \)
9: Compute \( \mathbf{V}_{-i}^{(j)} \) and \( S_{-i}^{(j)} \)
10: Estimate regression vector \( \omega_i^{(-j)} \), \( \hat{\omega}_i^{(-j)} \)
11: Determine \( \mathbf{B}_{i}^{(j)} = G_i^{(j)} \mathbf{V}_{-i}^{(j)} S_{-i}^{(j)} \)
12: Compute prediction error \( \mathbf{e}_i^{(j)} = \mathbf{x}_i^{(j)} - \mathbf{B}_i^{(j)} \hat{\omega}_i^{(-j)} \)
13: Update performance index \( J = J + ||\mathbf{e}_i^{(j)}||^2 \)
14: Set \( j = j + 1 \).
15: End
16: \( \hat{\sigma}^2_\ell(i) = J/L \)
17: \( \hat{\sigma}^2_\ell = \hat{\sigma}^2_\ell + \hat{\sigma}^2_\ell(i)/N \)
18: Set \( i = i + 1 \)
19: End
20: Check for convergence, if unsatisfactory goto Step 21
21: Update parameter set and return to Step 3 by setting \( \hat{\sigma}^2_\ell = 0 \), all \( \hat{\sigma}^2_\ell(i) = 0 \), \( i = 1 \) and \( j = 1 \)
22: Terminate iteration

VI. APPLICATION STUDIES

This section presents an application of the proposed method to determine the number of source variables for an industrial glass melter process.

The process uses molten glass to clad waste material, in the form of powder. Row glass frit is introduced in batches every two minutes, whilst the master material continuously enters the melter vessel. Four induction coils that are positioned around the vessel heat the binary composition. The constant filling results in an increasing liquid level inside the vessel. Upon reaching a certain height, the molten mixture is poured out and the next cycle of filling and heating begins.

Samples of the following variables were recorded every 5 minutes: (i) temperatures inside the melter vessel at 15 different positions, (ii) the power in each of the four induction coils, (iii) the voltage applied to the induction coils and (iv) the viscosity of the molten glass.

From this process, a total of \( L = 7500 \) samples were recorded, covering a number of filling cycles. After subtracting the estimated mean value from each of the 21 variables and scaling them to unit variance, we applied the two-fold cross-validatory approach for \( m = 1500 \) segments of 5 samples each. Using the Gaussian kernel function in Eq (7) and varying the parameter \( \sigma \) between 1 and 15 produced the results shown in Figure 1.

The optimal value of \( \sigma \) for \( \ell = 1 \) to 9 was obtained using a grid search, where the difference between two grid points is initially 0.25. After obtaining minimum using this coarse grid for each \( \ell \), a refined search around the optimal \( \sigma \) followed for a resolution of 0.01 between two grid points. The proposed two-fold cross-validatory method estimates \( n = 5 \). The optimal value for the kernel parameter \( \sigma = 10.75 \).

VII. CONCLUSIONS

This paper develops a method for simultaneously determining both the number of PCs and kernel parameter in KPCA model based on a review of the existing approaches in the literature. The application to a glass melter process confirms that the proposed two-fold cross-validatory approach, summarized in Algorithm 1, can optimally estimate these parameters.

APPENDIX

Based on Eq (24a), the least squares estimate of \( \omega_i^{(-j)} \) is:

\[
\hat{\omega}_i^{(-j)} = \left[ B_{i}^{(-j)T} B_{i}^{(-j)} \right]^{-1} B_{i}^{(-j)T} \mathbf{e}_i^{(-j)}
\]

For the \( k \)th sample the \( i \)th variable, i.e. \( \mathbf{x}_i^{(k)} \), we have:

\[
\mathbf{x}_i^{(k)} = \hat{\omega}_i^{(-j)} (\mathbf{e}_i^{(k)})^{(-j)} + \epsilon_i^{(k)}
\]

Next, defining \( U_i^{(-j)} = [ u_i^{(1)} u_i^{(2)} \ldots u_i^{(L-p)} ]^{(-j)} \) allows expressing \( U_i^{(-j)T} \mathbf{e}_i^{(-j)} \) in the form:

\[
U_i^{(-j)T} \mathbf{e}_i^{(-j)} (\mathbf{x}_i^{(k)}) = \begin{pmatrix} u_i^{(1)T} \\ u_i^{(2)T} \\ \vdots \\ u_i^{(L-p)T} \end{pmatrix} \begin{pmatrix} \hat{\omega}_i^{(-j)(\mathbf{x}_i^{(k)})} \\ \hat{\omega}_i^{(-j)(\mathbf{x}_i^{(k)})} \\ \vdots \\ \hat{\omega}_i^{(-j)(\mathbf{x}_i^{(k)})} \end{pmatrix} = \mathbf{e}_i^{(-j)(\mathbf{x}_i^{(k)})}.
\]

Based on the singular value decomposition of \( \mathbf{F}_{i}^{(-j)} \), we get:

\[
u_i^{(q)} = F_{i}^{(-j)T} u_i^{(q)} \hat{\omega}_i^{(-j)(\mathbf{x}_i^{(k)})},
\]

where \( u_i^{(q)} \) and \( u_i^{(q)} \) are the \( q \)th left and right singular vector of \( \mathbf{F}_{i}^{(-j)} \), respectively, and \( \hat{\omega}_i^{(-j)} \) is the corresponding singular value. Given that the eigenvectors of \( S_{i}^{(-j)} \) and the right
singular vector of $F_{-i}^{(-j)}$ are identical, the $q$th element of

$$
\Phi_{-i}^{(q)}(x(k)_{-i}) = \frac{v_{(q)}^{T}}{s_{(q)}^{-i}} \Phi_{-i}(x(k)_{-i})
$$

(34)

where $s_{(q)}^{-i}$ is the singular value associated with the $q$th singular vector. Therefore, the vector $\Phi_{-i}(x(k)_{-i})$ can be expressed using the mean centered kernel functions:

$$
\Phi_{-i}(x(k)_{-i}) = \frac{v_{(1)}^{T}}{s_{(1)}^{-i}} \Phi_{-i}(x(k)_{-i}) \cdots \frac{v_{(L)}^{T}}{s_{(L)}^{-i}} \Phi_{-i}(x(k)_{-i}) \times \begin{bmatrix}
\frac{1}{s_{(1)}} & 0 & \cdots & 0 \\
0 & \frac{1}{s_{(2)}} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{1}{s_{(L-p)}} \\
\end{bmatrix} \times \begin{bmatrix}
\Phi_{-i}(x(k)_{-i}) - k_{-i}^{(-j)} \\
\Phi_{-i}(x(k)_{-i}) - k_{-i}^{(-j)} \\
\vdots \\
\Phi_{-i}(x(k)_{-i}) - k_{-i}^{(-j)} \\
\end{bmatrix}
$$

Based on Eq (35), $B_{-i}^{(-j)}$ in Eq (24a) now becomes:

$$
B_{-i}^{(-j)} = \begin{bmatrix}
\Phi_{-i}(x(k)_{-i}) U_{-i}^{(-j)} \\
\vdots \\
\Phi_{-i}(x(k)_{-i}) U_{-i}^{(-j)} \\
\end{bmatrix} = \begin{bmatrix}
\Phi_{-i}(x(k)_{-i}) - k_{-i}^{(-j)} \\
\Phi_{-i}(x(k)_{-i}) - k_{-i}^{(-j)} \\
\vdots \\
\Phi_{-i}(x(k)_{-i}) - k_{-i}^{(-j)} \\
\end{bmatrix} = F_{-i}^{(-j)} \Phi_{-i}^{(-j)} V_{-i}^{(-j)} S_{-i}^{(-j)-1}
$$

Using Eq (36), the estimation of the regression vector $\hat{\omega}_{-i}$ according to Eq (30), now becomes:

$$
\hat{\omega}_{-i}^{(-j)} = S_{-i}^{(-j)} \left[ V_{-i}^{(-j)T} G_{-i}^{(-j)T} V_{-i}^{(-j)} \right]^{-1} V_{-i}^{(-j)T} G_{-i}^{(-j)T} F_{-i}^{(-j)}
$$

(37)

Note that $G_{-i}^{(-j)}$ is symmetric.

The size of $B_{-i}^{(-j)} = G_{-i}^{(-j)T} V_{-i}^{(-j)T} \hat{\omega}_{-i}^{(-j)}$ depends on how many nonlinear PCs are included. In the ideal case, there are $L - p$ nonzero eigenvalues, resulting in an $L - p$ dimensional regression vector $\hat{\omega}_{-i}^{(-j)} = S_{-i}^{(-j)} G_{-i}^{(-j)T} \pi_{-i}^{(-j)}$.

In most practical situations, however, the rank of $G_{-i}^{(-j)}$ is significantly smaller than its size $[2]$. By retaining the eigenvectors associated with the first $\ell$ largest eigenvalues of $G_{-i}^{(-j)}$, the dimension of the matrices in Eq (37) reduces to $S_{(-j)}^{(-i)} \in \mathbb{R}^{(L-p)\times \ell}$ and $V_{(-j)}^{(-i)} \in \mathbb{R}^{(L-p)\times \ell}$.

According to Eqs (23), (30) and (37), the model residuals $\varepsilon_{i}$, therefore, also depend on $\ell$ and $b$:

$$
\varepsilon_{i}^{(-j)} = \left[ I - G_{-i}^{(-j)T} V_{(-j)}^{(-j)T} G_{-i}^{(-j)T} \right]^{-1} \times

V_{(-j)}^{(-j)T} G_{-i}^{(-j)T} \pi_{-i}^{(-j)}
$$

(38)

REFERENCES


