PROJECT REPORT

KERNEL PRINCIPAL COMPONENT ANALYSIS

SUBMITTED BY:

DEEPANSHU SHARMA | 2017A1PS0674P

DATE OF SUBMISSION: 27/06/2020

Prepared in the fulfilment of the

Course No. - CHE F377

AT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

ACKNOWLEDGEMENT

Firstly, I would like to express my special thanks of gratitude to all those who provided me any sort of help related to my project. I would also like to give a hearty gratitude to my project mentor Professor Ajaya Kumar Pani, under whose guidance I got the exposure about the Kernel Principal Component Analysis and it's usage in Chemical Engineering. This project intrigued me to do the research on the given topic and how is it been used in Chemical Engineering. I also came to know about some other interesting topics linked to Kernel PCA.

TABLE OF CONTENTS

ABSTRACT

Principal component analysis (PCA) is the key technique to reduce the dimensionality of linearly separable data in input space and extract the key features. The non-linear variant of PCA, that is, Kernel PCA uses the complicated spatial structure of high-dimensional features. The underlying idea behind kernel PCA is to map the given data from input space into a feature space using some nonlinear mapping and then calculating the principal components in that feature space. In this report, I firstly review the basic ideas of PCA and its drawbacks. Then I focus on the detailed explanation of kernel PCA. I have also reviewed some research papers to understand how different datasets require different modifications to kernel PCA.

I have found that kernel PCA has precedence over other non-linear methods due to the following reasons:

i. It does not involve any nonlinear optimization, unlike principal curves approach.
ii. The calculations required to apply kernel PCA are as simple as for standard PCA.
iii. We do not need to specify the number of principal components prior to its application.

1. INTRODUCTION

Principal component analysis (PCA) is the popular method used to reduce the number of variables in the input space and extract important features with the aim of retaining as much information as possible. This technique combines the highly correlated variables together to generate lesser variables which are also called, principal components. The method works by solving an eigenvalue problem (of correlation matrix) that generates the principal components. In this way, it's able to summarize the information present in various variables by a smaller set of components that can be easily viewed and analyzed. The most important use of PCA is to present the multidimensional data as lesser number of variables (principal components) which helps us to observe clusters and trends present within the data. Moreover, this summarization may also help in uncovering the relationships between input variables and observation values for these variables. Also, the principal component analysis technique helps to discover the correlations among the variables. Hence, we get an idea about two kinds of relationship, that is, relationship between input variables with its observations and also the relation among different input variables. The extracted principal components can be realized as the new axes in the given dataset that maximize the variance along those axes (represented by the eigenvectors of the covariance matrix). So, basically PCA aims to find the axes along which maximum variance is present (or, in other words, along which the data is most spread).

For the datasets having a large no. of variables (or dimensions), we need to have very high computation power to process the whole of the data. Generally, there are variables in the data which do not contribute (or contribute to a very little extent) to any variance present, hence they do not play any significant role in the further processing. So, we apply the PCA on the given dataset as a pre-processing step which generates the principle components. By using the knowledge of how much variance we want to preserve in our data, we choose the no. of principle components to do further processing.

The standard PCA technique uses the linear projection approach that only performs well if the data can be linearly separated. However, for the data which cannot be linearly separated, we need to use some nonlinear technique for such case.

For instance, it sometimes may be the case that if the d-dimensional data points x_n, where n = 1, 2, 3 … are nonlinearly mapped into a higher dimensional space y = f (x), different clusters in the dataset can be better separated than in the original space. For example, suppose that we have two groups of data values given in a 2-D space (x_1, x_2) in which they form two circles centered at the origin. As it is impossible to linearly separate these two groups of points in the 2-D space, these 2-D points are non-linearly mapped into a 3-D space (x_1, x_2, $x_1^2+x_2^2$), the two groups of points can be easily separated linearly in the 3rd dimension which actually corresponds to the radius of the circle.

Hence, Kernel PCA is used on non-linearly separable data containing highly correlated variables.

2. DETAILED EXPLANATION OF KERNEL PCA

Kernel PCA is said to be the nonlinear generalization of PCA because of the following two reasons:

(i) It applies PCA, but in feature space which can be of any large dimension.
(ii) On applying the kernel k(x, x') = <x, x'> (<a, b> represents the dot product of a and b), the standard algorithm for PCA can be retrieved.

Kernel PCA have several pros and cons in comparison to other nonlinear methods. The crucial edge of this method over others is that it do not require any nonlinear optimization. Moreover, we don't even need to specify the number of principal components that we want to obtain prior to modeling. However, Kernel PCA is difficult to comprehend in input space, unlike the method of principal curves. Also, this method have the limitation that if there are huge number of observations present in the given dataset, then it will lead to the formation of a huge kernel/gram matrix which may pose a challenge to the computational power of the device. Though we may still apply sparse greedy methods to overcome this limitation and can carry out Kernel PCA approximately. (More details on comparison with other methods are given in Section 2.3)

2.1 Kernel PCA as an Eigen Value Problem

Consider the non-linear mapping of input data points from x to f (x) into the feature space F of higher dimensions. In this feature space, the covariance matrix will be calculated as:

$$\bar{C} = \frac{1}{N}\sum_{n=1}^{N} f(x_n) f(x_n)^T$$

(Assuming the data remain centered in the feature space)

Putting the covariance matrix into the Eigen equation (i.e., when any given matrix gets multiplied by some specific vectors ϕ_i , then the equation returns some multiple of that

specific vector ϕ_i. These vectors are called eigenvectors and the multiples are called eigenvalues.):

$$\bar{C}\phi_i = \lambda_i \phi_i$$

So, we get:

$$\left[\frac{1}{N}\sum_{n=1}^{N} f(x_n) f(x_n)^T\right]\phi_i = \frac{1}{N}\sum_{n=1}^{N} (f(x_n).\phi_i)\, f(x_n) = \lambda_i \phi_i$$

Now, as the linear combination of the N given data points mapped into the feature space gives the eigenvector ϕ_i , we may write:

$$\phi_i = \frac{1}{\lambda_i N}\sum_{n=1}^{N} (f(x_n).\phi_i)\, f(x_n) = \sum_{n=1}^{N} a_n^{(i)} f(x_n)$$

Therefore, we get:

$$a_n^{(i)} = \frac{1}{\lambda_i N}(f(x).\phi_i)$$

By multiplying $f(x_m)^T$ to the above equation, we get:

$$(f(x_m).\phi_i) = \lambda_i N a_m^{(i)} = \sum_{n=1}^{N} a_n^{(i)} (f(x_m).f(x_n)) = \sum_{n=1}^{N} a_n^{(i)} k(x_m, x_n)$$

Where

$$k(x_m, x_n) = \big(f(x_m).f(x_n)\big), \qquad where, m, n = 1,2, \dots, N$$

Represents the gram matrix which is the inner-product of two vectors in the feature space F. So, if we take m = 1 …, N, then we can convert the above scalar equation into a vector equation where the above equation represents the m-th component of the following vector equation:

$$\lambda_i N a_i = \boldsymbol{K} a_i$$

Where, $\boldsymbol{K} = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & k(x_i, x_j) & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}_{Nx\ N}$

Is a matrix of N x N kernel elements $k(x_i, x_j), (i, j = 1,2, \ldots, N)$ and $a_i = [a_1^{(i)}, \ldots\ldots, a_N^{(i)}]^T$ (i = 1, … , N) can be calculated by solving the eigenvalue equation of **K** and these a_i will represent the N eigen vectors of **K**. Now, these a_i will act as the principal components, selecting only those which have higher corresponding eigenvalues.

Also, we can map any new data point x to f(x) in the feature space F by retrieving its n-th component from its projection on the n-th eigenvector $\phi_i = \sum_{n=1}^{N} a_n^{(i)} f(x_n)$ which can be written as:

$$(f(x).\phi_i) = \left(f(x).\sum_{n=1}^{N} a_n^{(i)} f(x_n)\right) = \sum_{n=1}^{N} a_n^{(i)} \big(f(x).f(x_n)\big) = \sum_{n=1}^{N} a_n^{(i)} k(x, x_n)$$

It's important to note that the mapping f (x) in the kernel PCA is not specified directly, neither is the dimension of the feature space F. Hence, we use kernel function to apply kernel PCA and determine the parameters of the kernel function empirically (generally). The procedure to apply the Kernel PCA is shown in the next section (section 3) step by step.

Note: The explanation above assumed that the given data points are centered in original space as well as in the feature space. However, the raw data is generally not centered, so we need to center our dataset before doing any analysis which is shown in the next section (section 3).

2.2 Properties of Kernel PCA

i. There is no correlation among the principal components.

ii. The first n (where n may vary from 1 to m, m being the number of observations) principal components retain more variance than any other subset of n components.

iii. The first n principal components (ordered in the decreasing magnitude of eigenvalues) have maximum info regarding the input data. (This property is only

valid under Gaussian assumption in F and hence depends on the choice of Kernel and given dataset.)

iv. The first n principal components give minimum MSE value while depicting the input data points in F.

v. This technique calculates the number of principal components that may be even higher than the dimension of input dataset, unlike regular PCA. For instance, let the number of data points in the given dataset are m, and this value is higher than the dimension of the input dataset, N. Here, standard PCA can claim maximum of N non-zero eigenvalues. However, on the other side, Kernel PCA can determine up to m eigenvalues (that are not equals to 0).

vi. As regular PCA is based on the transformation of basis, it's feasible to reconstruct the initial patterns from the extracted principal components by doing expansion in the eigenvector basis. Moreover, by using a partial set of components, we may still get good reconstruction. On the contrary, this is even more difficult to achieve in case of Kernel PCA. Only if we have some approximate reconstruction, then we may remodel the initial patterns in F from its components. But still, there is no surety that we will be able to construct the same pre-image of the reconstruction.

2.3 Comparison of Kernel PCA with other methods

Hebbian Networks

Firstly, I will discuss the pioneering work of E. Oja [3], where several algorithms have been proposed based on neural networks to calculate PCs. They have advantages in cases where the data are dynamic in nature, as there is no need of using the normal method of diagonalizing the covariance matrix. By using the nonlinear neurons, we may get the nonlinear version of these algorithms.
The algorithms then calculate the feature components, referred to nonlinear PCs. However, these methods do not have similar geometrical meaning as of Kernel PCA. Hence, it is therefore very tough to realize what have actually been calculated from these methods.

Auto associative Multi-Layer Perceptron

Auto associative Multi-Layer Perceptron is a linear perceptron with only one hidden layer, which must be smaller than the given input i.e., the dimension of the given dataset. Now, the model is trained to get the output equals to the input, then the hidden unit activation function generates a lower-dimensional depiction of the given data and this is closely linked with the linear PCA.

To generalize this to the nonlinear version, we may use nonlinear neurons and some more hidden layers. Though this method can have resemblance with the nonlinear PCA, but we need to do the optimization job which is prone to errors due to its inability to differentiate between local and global minima. Moreover, there are high chances of overfitting the training data. Also, one more downside of these proposed approaches is that we need to mention the number of components to be calculated prior to modeling.

Principal Curves

The approach of principal curves gives idea about geometric representation in input space, by iteratively estimating a curve that apprehends the shape of the given data. This approach works by projecting the data on a curve which is to be calculated from the algorithm, and have the characteristic that each data point on the curve is the mean of all data points projecting on it. Only straight lines which follow the above mentioned property form the PCs and hence, this method can be considered as the notion behind linear PCA. We need to solve a nonlinear optimization problem to calculate the principal curves. Also, we need to specify the number of features to be generated prior to modeling.

Multidimensional Scaling (MDS)

The underlying idea behind Multidimensional Scaling is to project the data points and keep the distances among the observations. For instance, let there are k data points given such that for each pair of data point (i, j), we have dissimilarity measure between each pair of data points. Our focus is to place these k data points into a Euclidean space so that there exist some relationship between distances of data points and their associated dissimilarity. In case of classical scaling, inter point distances is equals to the associated dissimilarities. While in metric MDS, the distances between each pair of data points are related as some nonlinear functions of their respective dissimilarity. If the kernel used in the modeling of kernel PCA is isotropic in nature, then we can understand kernel PCA as doing metric MDS.

This can be done by doing classical scaling in the higher dimensional space. Hence, we need to minimize the stress function over the sum of all dissimilarities and for that we have to perform optimization.

Locally Linear Embedding (LLE)

Locally Linear Embedding is another dimensionality reduction method that preserves the local structure within the given data. Compared to MDS, this method do not require to calculate the distances among each of observations and form global structure from local linear fittings. Though this method determines embedding which retain the local structure, there is always chances of its diversion which may give false results. So, if we have taken a smooth surface (given the neighborhood taken is small), the proposed method may perform better. This method heavily depends on selection of local neighborhood. Also, if the kernel matrix involved in kernel PCA algorithm is calculated through LLE kernel, we may get its solution.

Orthogonal Series Density Estimation

Kernel PCA is the method of calculating the number of nonlinear PCs from multidimensional dataset. The eigenvalue decomposition of gram matrix in kernel PCA algorithm gives the coefficients for a nonparametric orthogonal series density estimator. This approach forms the relation between the Eigen function of the integral operator T_k connected with the kernel function and the eigenvectors calculated from the Kernel matrix.

3. PROCEDURE TO APPLY KERNEL PCA

Suppose we have a dataset containing M samples and N variables, so the dataset can be thought as an M x N matrix. While approaching the solution of kernel PCA, we assume that the dataset consists of M vectors of N dimensions. For example: if we have 100 sample points with 3 variables, we may think of it as 100x3 matrix or, 100 3d points in space.

i. Computation of the kernel (similarity) matrix:

Firstly, we will evaluate $K(x_i, x_j)$ for every pair of observations.

Note: K will be a matrix of M x M dimension and each point represents the (1 x N) vector.

Some common types of Kernel functions which are used to calculate Kernel matrix are Polynomial, Gaussian kernel, sigmoid kernel and Gaussian radial basis function (RBF).

Now, all further steps are similar to the standard PCA in which principal components are determined by utilizing the eigenvalues and eigenvectors of the covariance matrix.

ii. Centering of data in Kernel matrix:

Since it is not necessary that the kernel matrix is centered beforehand, we first need to apply the following equation to do so:

$$K_{ij} = (K - 1_M K - K1_M + 1_M K 1_M)_{ij}$$

, where, 1_M is (like the kernel matrix) an M x M size matrix with all values equal to 1/M.

Now, we need to calculate the eigenvectors of the centered kernel matrix and consider only those vectors which correspond to the highest variance or, eigenvalues.

Note: Whenever K is used below, it is representing the centered kernel matrix K_{ij}.

iii. Eigen decomposition of the kernel matrix:

$$M\lambda\alpha = K\alpha$$

This equation will provide eigenvalues and eigenvectors of the given Kernel matrix.

iv. Normalization of eigenvectors coefficients:

$$1 = \lambda_k(\alpha^k . \alpha^k)$$

Applying each eigenvector to the above equation to get the normalized values. This step needs to be done when some of the eigenvalues are zero. Now, taking the 'k' largest eigenvalues which correspond to 'k' eigenvectors such that 'k' eigenvectors can show the desired amount of variance.

Note: Each eigenvector is of M dimensions and there are total M eigenvectors, out of which 'k' are selected, so the final result will be the matrix of size (k x M) for the given dataset. However, if we want to project the new data points on the existing eigenvectors or principal components, we can do so by following the Step 5.

v. Projecting new data (x):

$$V^k . \phi(x) = \sum_{i=1}^{M} \alpha_i^k . K(x, x_i)$$

Now, above equation is applied to get $\phi(x)$.

Note: Here, V^k represents the matrix of size (k x M), containing all the 'k' selected eigenvectors and α_i^k represents each eigenvector.

In this way, we can easily project the new data without even knowing about the kernel function ϕ.

4. DRAWBACKS OF KERNEL PCA

i. Kernel PCA is the nonlinear variant of standard PCA which uses kernel trick to extract the components in Feature space F. However, the size of kernel matrix is of concern due to the limited computation power of today's generation machines. The size of kernel matrix increases at double rate than the rate at which input data grows.

ii. Industrial processes and the associated variables are always complex in nature and also do not completely follow Gaussian or non-Gaussian distribution, that's why any single technique is never adequate to take out the hidden info with good accuracy. And, KPCA do not perform well for non-Gaussian data (Kernel Independent Component Analysis is generally used for non-Gaussian data).

iii. Kernel PCA can't capture the time-varying relationship between variables (dynamic systems), which results into higher false alarms frequency and missing detection rates in fault detection.

iv. Kernel PCA does not consider any inner relationship among neighborhood observations. That's why, Kernel PCA can be thought as a techique to analyze the global data structure. Hence, Kernel PCA do not able to preserve the local structure during transformation into feature space. So, to caluculate the components which can retain maximum variance of the original data and also maintain the local data structure among the neighboring observations, Local KPCA is used.

v. If some of the variables present in the given dataset are loosely dependent or independent to each other, then Kernel PCA alone can't perform the analysis successfully. Hence, we first need to separate the loosely dependent or independent variables from dependent ones and then apply different analytical techniques on both categories of variables.

5. MODIFICATIONS & APPLICATIONS OF KERNEL PCA IN CHEMICAL ENGINEERING

i. "Nonlinear process monitoring using kernel principal component analysis" [9] (2004):

In this paper, two statistics are introduced, that is, T^2 and SPE (Q statistic) and applied to the KPCA algorithm. In order to calculate the variation in the kernel PCA, T^2 statistic is used which is actually the sum of normalized scores. While SPE/Q statistic is used to calculate the goodness of fit to PCA. This method is used on a simple multivariate process and also on the biological wastewater treatment process. The processes showed that the approach given in the paper is able to determine the nonlinear relationships among input variables. Also, the proposed method gave better accuracy than the standard PCA when used for process monitoring. To calculate the number of principal components from both methods, different approaches are used: for linear PCA, they used a cross-validation method based on PRESS (prediction residual sum of squares) and for Kernel PCA, they employed the cut-off method that use the average eigenvalue. Cutoff method is chosen because of its simplicity and robustness. In this paper, RBF (radial basis kernel function) is used and the parameter is chosen after testing the monitoring performance for various values of c.

ii. "Fault identification for process monitoring using kernel principal component analysis" [1] (2005):

In this paper, two more statistics are introduced, that is, C- T^2 and C- SPE (Q statistic) and applied on the KPCA algorithm for Fault Identification. This proposed approach is also applied on two processes, that is, on a simple nonlinear process and also on a non-isothermal CSTR process. These statistics gives idea about the contribution of each dimension (or, variable) to the previously introduced statistics T^2 and SPE, which facilitates the identification of faulty variable. The proposed method is applied on two type of faults, that is, ramp and bias type, and both type of faults gave adequate results. Also, it can successfully distinguish a faulty variable even from a complex fault which they showed for the first type of fault on the CSTR process. They have used radial basis function and determined the kernel parameter empirically. They have calculated the number of principal components by using DimFS. 90% of the variance was retained in the transformed data.

iii. "Enhanced statistical analysis of nonlinear processes using KPCA, KICA, and SVM" [15] (2009):

Industrial processes and the associated variables are always complex in nature and also do not completely follow Gaussian or non-Gaussian distribution, that's why any single

technique is never adequate to take out the hidden info with good accuracy. Hence, both KICA (for non-Gaussian part) and KPCA (for Gaussian part) are used to detect the faults. The proposed approach is applied to the Tennessee Eastman (TE) process to detect and diagnose the faults. The support vector machine classifier not only used for modeling but also helped in distinguishing the flow regimes which assists in forecasting the transition region. The performances of each method (Fisher Discriminant Analysis, Support Vector Machine and proximal SVM) are compared with each other for fault diagnosis. In this paper too, they have chosen the kernel parameters by trial and error method. Initially, some value is empirically decided, and then the variations around the initial value are observed to select the value which gave better results. Here also, T^2 and SPE (Q statistic) are used to detect the fault and to diagnose it.

iv. "Improved kernel PCA-based monitoring approach for nonlinear processes" [5] (2009):

In this paper, a new variable is proposed, that is, improved residual in the statistical local approach. The key feature of this proposed variable is that it will only follow Gaussian distribution, hence its distribution would not depend on the distribution of the given dataset. Moreover, two statistics are also proposed to monitor the process by evaluating their extreme values by $\chi 2$ distribution. This joint local approach KPCA method showed better accuracy on fault detection sensitivity than classic Kernel PCA. Two examples were used to test the new approach, i.e., a numerical study and Tennessee Eastman benchmark process.
For Kernel PCA , they used the RBF kernel and kernel parameter is determined empirically. For the first numerical example, only two kernel principal components are selected which showed 95.34% of the variance of the original data. However, they selected 20 PCs to explain 97.41% of the original variance present for the case of Tennessee Eastman process. According to the cutoff value 0.001 for the eigenvalue, the dimension of the mapped space was selected as 26.

v. "Dynamic processes monitoring using recursive kernel principal component analysis" [14] (2012):

The key features of the proposed approach are:

a. New SVD technique is suggested in this paper.
b. Also, new model is generated by using the data from the previous model and also by utilizing the recursive eigenvectors and their corresponding eigenvalues in the feature space.

The results are beneficial as the dynamic nature of the input variables was captured in the proposed method. It is tried on two processes (continuous annealing and penicillin fermentation) to check how efficiently it could monitor the variables and changes its

monitoring strategy according to the variations in the data. Recursive Kernel PCA could successfully capture the dynamic as well as the nonlinear relationship in process variables while Kernel PCA failed to do so. Hence, false alarms frequency and missing detection rates are reduced.

vi. "The optimization of the kind and parameters of the kernel function in KPCA for process monitoring" [7] (2012):

To select the kernel function and its parametrs for kernel PCA, an optimization problem is designed by using genetic algorithm. While appying the algorithm, it is required to select the decision variables and objectives/constraints. So, kernel function and its parameters are selected as variables and three objectives are defined:

a. Maximum value of monitoring rate
b. Minimum number of principal compnents, and
c. Minimum value of squared prediction error

The above proposed techinque is implemented on a numerical example and also on penicillin fermentation process. For penicillin fermentation process, a study [10] indicated that polynomial function would be more efficient that any other kernel function to understand the intrinsic nonlinear nature of the process and its variables.
The idea of genetic algorithm is taken from the theory of natural evolution. Genetic algorithm implements the idea of natural selection by taking the fittest individuals for reproduction to generate offspring. If the parents have good fitness, then their offsprings will be better than parents and hence will have more chance of surviving. This process keep on repeating itself and finally, a generation will be found which have the fittest individuals. This approach is applied on a optimization problem. We try to find several solutions for a specified problem and then choose the best out of all. For the penicillin fermentation process, a modular simulator PenSim v2.0 is used. This simulator is developed on the basis of the model formed by Bajpai and Reuss.

vii. "Modified kernel principal component analysis based on local structure analysis and its application to nonlinear process fault diagnosis" [2] (2013):

Kernel PCA does not consider any inner relationship among neighborhood observations. That's why, Kernel PCA can be thought as a techique to analyze the global data structure. Hence, Kernel PCA do not able to preserve the local structure during transformation into feature space. That's why, a modified form of kernel PCA, called local KPCA is introduced in this paper. An optimization problem needs to be solved whose objectives are to maximize $\alpha TKK\alpha$ (for global variance extraction) and to minimize $\alpha TKLK\alpha$ (for local structure-

preserving) to preserve the local structure within data and also retain maximum data variance. They presented a LKPCA contribution plot method to recognize the variables which are faulty in nature. This method determines the changes in the output variables caused due to the distubances in parameters, similar to the sensitivity approach. The Gaussian kernel function is used for the Tennessee Eastman and its parameter is calculated through cross-validation method. The local Kernel PCA gave better results than normal kernel PCA when applied on the TE process for the detection of faults.

viii. "Related and independent variable fault detection based on KPCA and SVDD" [6] (2016):

In this paper, independent and related variables are separated first based on Mutual Information (MI). Then, SVDD (Support Vector Data Description) is used to examine the space formed by independent variables and Kernel PCA is used for correlated variables. On the basis of the results from both of these methods, a statistic is formed. MI approach is used to evaluate the interdependence among variable and is also able to determine linear as well as nonlinear relations. Hence, this method is able to measure 2^{nd} or even higher order dependencies. To check the performance, the proposed method is applied on a numerical example and also on the TE process. The underlying approach behind SVDD is to calculate a hypersphere that would entail the whole input data using as low volume as possible. And, this problem of calculating the hypersphere having minimum volume is solved by converting it to an optimization problem, which is futhur converted to its dual form. To do the transformation of input data to the feature space, the Gaussian kernel function is used. The monitoring performance showed that the proposed method performed better than standard PCA, Kernel PCA, and SVDD methods. However, this method is only applied to the normally distributed data and also in a single-mode process.

ix. "Online reduced kernel principal component analysis for process monitoring" [4] (2018):

This paper worked on two limitations of Kernel PCA:

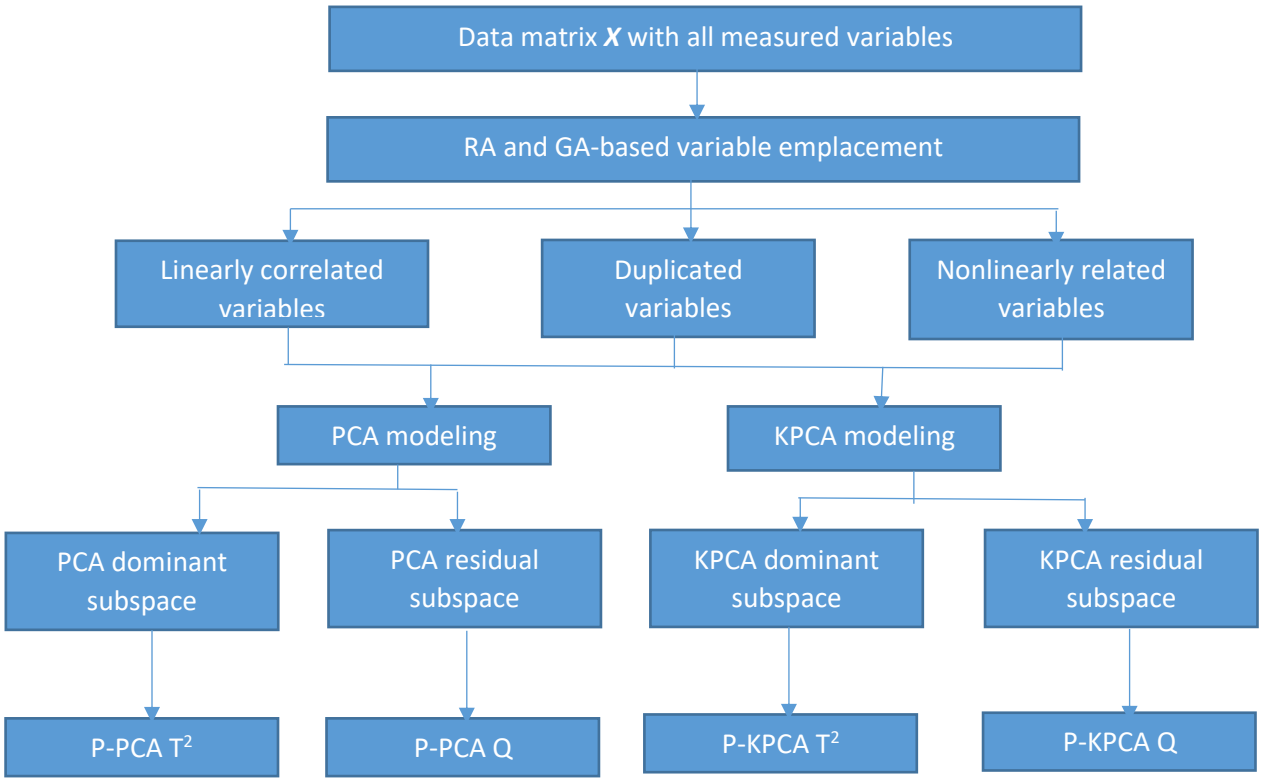
a. It can't capture the dynamic nature of process variables and hence, do not give promising results.
b. It starts giving vague response when the training dataset is large.

The proposed method is conducted in the two phases. Firstly, the offline phase generates the starting kernel PCA model and also calculate the square prediction error (SPE) index. Then, online phase only changes this SPE index according to the conditions. As the number of observations increases, the number of kernel functions in the proposed method also increases

with same rate. However, higher increase in kernel functions makes it inappropriate for online use. The major limitations of the algorithm are that it challenges the computational power of the machine which may be very time consuming for the amending the model and there are also the chances of overfitting the training data. Now, to overcome these limitations, a dictionary is generated which carry only the suitable kernel functions. So, the parameters involved in the method are changed when there are any changes in the dictionary. In this way, a new online reduced kernel PCA algorithm is generated and applied on a numerical problem and also on TE process.

x. "Parallel PCA–KPCA for nonlinear process monitoring" [8] (2018):

This papers tries to solve the problem of process monitoring having dataset which involves both, linearly as well as non-linearly related variables. Randomized algorithm integrated with genetic algorithm is used to find which variables are used in the both of the models. The flowsheet given below explains the modelling of Parallel PCA-KPCA:



(The above flowchart is directly taken from [8].)

The Parallel PCA-Kernel PCA proposed approach follow 2 steps, i.e., emplacement of variables and obtaining the monitoring statistic. For the first step, Randomized algorithm and Genetic Algorithm designate the variables for both the models by determining the highest

Fault Detection Rate on the given dataset. While in the second step, appropriate statistics are generated for both of these models. This method is applied on a numerical example and on the CSTR. This study only discusses fault detection issues, nothing mentioned about fault diagnosis. And, if the number of variables are less, then only KPCA model can give satisfying results.

xi. "Sensor fault detection and isolation of an industrial gas turbine using partial adaptive KPCA" [11] (2018):

In this paper, adaptive T^2 statistic is used with Kernel PCA to apply on dynamic systems having nonlinear nature. The proposed statistic checks the connection among the input dimensions and detects the variations. A threshold value is chosen for the statistic by keeping in mind the maximum acceptable false detections. If the statistic value for any new record is higher than the threshold value, then there are high chances that some fault has happened in the process. One advantage of adaptive T^2 statistic over normal T^2 statistic is that the former is affected by varying magnitude as well as directions of the dynamic shift, unlike normal T^2 statistic which only take magnitude into consideration. Hence, while dealing with the dynamic processes, adaptive T^2 statistic gave better results. However, there is one main drawback of these type of approaches that we need to have clean data which takes all the working conditions (without any fault) of the system into consideration, which is generally not the case for industry level data.

6. CONCLUSION

In this report, I have discussed kernel PCA in detail ranging from its theory to application of it in Chemical Engineering. I have also shown the step by step approach to apply kernel PCA and some of the cases in which kernel PCA fails to perform well.

As discussed while comparing the kernel PCA with other non-linear methods, it has precedence over others as it only needs the answer of an eigenvalue problem, so there is no need to perform nonlinear optimization. Moreover, it also provides solution to many categories of non-linearities by the virtue of different kernel functions that are available.

Today, standard PCA have several scientific and technical implementations, including noise reduction and fault detection. Kernel PCA can be used in all of the fields where standard PCA is being used to extract the key features plus where nonlinear behavior is shown by the input data.

Hence, I would say before applying any technique on the given dataset, firstly do some exploratory data analysis to understand the data and then apply the technique that is suitable with the data as shown in the Modifications & Applications of kernel PCA in Chemical Engineering section of the report that different type of datasets require different modifications of Kernel PCA.

7. REFERENCES

1. Cho, J., Lee, J., Choi, S., Lee, D., Lee, I., 2005. Fault identification for process monitoring using kernel principal component analysis. Chemical Engineering Science 60 (2005) 279 – 288.
2. Deng, X., Tian, X., Chen, S., 2013. Modified kernel principal component analysis based on local structure analysis and its application to nonlinear process fault diagnosis. Chemometrics and Intelligent Laboratory Systems 127 (2013) 195–209.
3. E. Oja. A, 1982. Simplified neuron model as a principal component analyzer. Journal of Mathematical Biology, 15:267-273.
4. Fezai, R., Mansouri, M., Taouali, O., Harkat, M. F., Bouguil N., 2018. Online reduced kernel principal component analysis for process monitoring. Journal of Process Control 61 (2018) 1–11.
5. Ge, Z., Yang, C., Song, Z., 2009. Improved kernel PCA-based monitoring approach for nonlinear processes. Chemical Engineering Science 64 (2009) 2245-2255.
6. Huang, J., Yan, X., 2016. Related and independent variable fault detection based on KPCA and SVDD. Journal of Process Control 39 (2016) 88–99.
7. Jia, M., Xu, H., Liu, X., Wang, N., 2012. The optimization of the kind and parameters of kernel function in KPCA for process monitoring. Computers and Chemical Engineering 46 (2012) 94-104.
8. Jiang, Q., Yan, X., 2018. Parallel PCA–KPCA for nonlinear process monitoring. Control Engineering Practice 80 (2018) 17–25.
9. Lee, J., Yoo, C., Choi, S.W., Vanrolleghem, P.A., Lee, I., 2004. Nonlinear process monitoring using kernel principal component analysis. Chemical Engineering Science 59 (2004) 223 – 234.
10. Lee, J. M., Yoo, C. K., & Lee, I. B., 2004. Fault detection of batch processes using multiway kernel principal component analysis. Computers and Chemical Engineering, 28, 1837–1847.
11. Navi, M., Meskin, N., Davoodi M., 2018. Sensor fault detection and isolation of an industrial gas turbine using partial adaptive KPCA. Journal of Process Control 64 (2018) 37–48.
12. Scholkopf, B., Smola, A.J., Muller, K., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10 (5), 1299–1399.
13. Scholkopf, B., Smola, A.J., 2002. Learning with Kernels (Support Vector Machines, Regularization, Optimization, and Beyond).
14. Zhang, Y., Li, S., Teng, Y., 2012. Dynamic processes monitoring using recursive kernel principal component analysis. Chemical Engineering Science 72 (2012) 78–86.
15. Zhang, Y., 2009. Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM. Chemical Engineering Science 64 (2009) 801-811.