On Robustness of Kernel Principal Component Analysis using Fast HCS

Lekaa Ali Muhamed¹, Hayder Yahya mohammed²

¹ Asst. Prof. Department of Statistics, College of Management and Economics
Baghdad University, Baghdad, Iraq

² PhD student. Department of Statistics, College of Management and Economics
Baghdad University, Baghdad, Iraq

Corresponding Author Email: haiderstatistic@yahoo.com.

Abstract

When dealing with multivariate data with higher dimensions, we often use principal component analysis (PCA) to lessen the dimensions, but in the case of nonlinear data it is not possible to deal with classic estimated because of obtaining misleading results and therefore using kernel methods , when data contain outliers the results of the kernel pca (KPCA) for correlation matrix or variance covariance matrix are inaccurate. The aim of this research is to employ Robust KPCA (RKPCA) to solve nonlinear data using kernel function and outlier observation using Robust method termed as FastHCS (High-dimensional Congruent Subsets) that stands for a robust PCA algorithm appropriate for high-dimensional appliances to know the most effect variables on the phenomenon.

Keywords: High-dimensional data, outlier detection, kernel principal component analysis(KPCA) , FastHCS (High-dimensional Congruent Subsets).

I. Introduction

Principal component analysis (PCA) can be employed to discover and interpret the apparent reliability between variables and to examine the relationship that may exist between observations [VII].

It is able to compress huge dimensional data with diminutive lost information through projecting the data onto a minor dimensional subspace demarcated by a new-fangled group of derived variables called component that are not related to each other, by an eigen decomposition of the centered covariance matrix of the data set [IX][III].

KPCA by way of a nonlinear feature extractor has demonstrated as influential preprocessing step for procedures of classification. Nevertheless, it can be also
considered as a natural generality of linear PCA. The results of this extrication were affected by the kernel type and the size of bandwidth used, the observed data are transformed as a result of a nonlinear function high dimensional feature space using kernel trick [I][VI][VIII].

KPCA stands for a tremendous nonlinear algorithm, but it doesn’t reflect in the input data the state of outliers, so the it does not disregard outliers, so we use Robust Kernel Pca Using FastHCS to solve this problem.

Materials and methods

Kernel principal component analysis

As a general rule, PCA has the feasibility to be successfully accomplished on a group of observation that are linear, assumed a group of M centered observations \( x_k \), \( k=1,\ldots,M \), \( x_k \in \mathbb{R}^N \), \( \sum_{k=1}^{M} x_k = 0 \), PCA diagonalizes the covariance matrix:

\[
C = \frac{1}{m} \sum_{j=1}^{m} x_j x_j^T
\]  

(1)

For doing this, the eigenvalue equation can be solved by:

\[
\lambda V = CV
\]  

(2)

For eigenvalue \( \lambda \geq 0 \) and eigenvector \( v \in \mathbb{R}^N \setminus \{0\} \). As \( CV = \lambda \sum_{j=1}^{m} (x_j . v) x_j \), all solution \( v \) must lie in the span of \( x_1 \ldots x_M \), hence (2) is equivalent to

\[
\lambda(x_k . v) = (x_k . Cv) \text{ for all } k = 1 \ldots M
\]  

(3)

when the variations are nonlinear [VII], The use of kernel functions that describe this computation in another dot product space \( F \), that is associated with input space by a possible nonlinear map data \( X \) into \( \Phi(x) \) (nonlinear transformation), \( F \) stands for the feature space. The covariance matrix in \( F \) is:

\[
C^F = \frac{1}{m} \sum_{j=1}^{m} \Phi(x_j) \Phi(x_j)^T
\]  

(4)

We now have to fined eigenvalue \( \lambda \geq 0 \) and eigenvector \( v \in \mathbb{R}^N \setminus \{0\} \) satisfying

\[
\lambda V = C^F V
\]  

(5)

\[
C^F V = \left( \frac{1}{m} \sum_{j=1}^{m} \Phi(x_j) \Phi(x_j)^T \right) V
\]  

(6)

\[
\lambda \Phi(Xk, V) = \left( \Phi(Xk) C^F V \right)
\]  

(7)

Then we find the coefficient of characteristic root coefficient \( \alpha_i \) \( (i=1\ldots m) \) as follows

\[
V = \sum_{i=1}^{m} \alpha_i \Phi(x_j)
\]  

(8)
Combining (7) and (8), we get
\[ \lambda \sum_{j=1}^{m} \alpha_i(\emptyset(Xk), \emptyset(xi)) = \frac{1}{m} \sum_{i=1}^{m} \alpha_i(\emptyset(Xk), \sum_{j=1}^{m} \emptyset(xj)) \cdot (\emptyset(xj), \emptyset(xj)) \] (9)

For all \( k = 1 \ldots m \)

Define an \( M \times M \) matrix \( K \) by
\[ K = [K_{ij}] = (\emptyset(xi), \emptyset(xj)) \] (10)

This leads to
\[ \lambda m K \alpha = K^2 \alpha \] (11)

where \( \alpha = [\alpha_1 \ldots \alpha_M]^T \). To find a solution of equation (11), the eigenvalue problem can be solved by:
\[ \lambda m \alpha = K \alpha \] (12)

for nonzero eigenvalues. An explanation of this process is specified in Schölkopf et al. 1998. Currently, performance PCA in \( F \) is corresponding to resolve the eigen-problem of the equation (12). This yields eigenvector \( \alpha_1 \ldots \alpha_M \) and eigenvalues \( \lambda_1 \geq \cdots \geq \lambda_M \). The dimensionality of the problem can be decreased as a result of holding only the 1st \( p \) eigenvector. \( \alpha_1 \ldots \alpha_p \) are normalized in the case of the equivalent vectors in \( F \) are normalized as exampled by:
\[ (v_k, v_k) = 1 \quad \text{for all} \quad k = 1 \ldots p \] (13)

By asset of (8) and (12) leads to
\[ 1 = \sum_{i,j=1}^{M} \alpha_i \alpha_j K_{ij} = \sum_{i,j=1}^{M} \alpha_i \alpha_j K_{ij} = \lambda_k (\alpha^k \alpha^k) \] (14)

The \( PC_t \) of attest vector \( x \) is then extracted by projecting \( \emptyset(x) \) onto eigenvector \( v_k \) in \( F \), where \( k = 1 \ldots p \).
\[ t_k = (v_k, \emptyset(x)) = \sum_{i=1}^{M} \alpha_i \emptyset(xi), \emptyset(x) \] (15)

May be called its nonlinear pc analogous with \( \emptyset \). (8)(9)

To evade the categorical nonlinear mapping, the kernel trick can be employed. Different kernel functions that satisfied Mercer’s formula of practical investigation. There is a mapping into a space as a kernel function stands for a dot product in the case of kernel function has been continuous for positive integral operator. Therefore, the condition on a kernel function is that it agrees with Mercer’s formula for representative kernel function is as following:

Laplacian Kernel:
\[ K(X, Y) = \exp (-\|X - Y\|/c) \]
Where \( c \) stands for quantified a priori by the user [VII].

**The Robust Kernel PCA Using FastHCS**

KPCA stands for highly tremendous nonlinear PCA process. Nevertheless, it doesn't take into account the circumstances of the outliers in the input data. Consequently, it doesn't remove outliers as used in PCA in feature space. It is reasonably straightforward to exclude outliers in input space. Numerous researchers have projected numerous procedures. Nonetheless, eliminating outliers in feature space are highly problematic for the reason that the explicit cannot obtained form non-linear mapping function \( \Phi \). With the intention of eliminating the consequence of outliers to the algorithm. We need to study two complications. Initially, whether or not the outliers in input space persist outliers in feature space. Secondly, how to exclude outliers in feature space? By way of the 1st problem has fixed in the case of the nonlinear mapping has been smooth and incessant. At that point, the topographic organization of the data in input space can be well-kept in feature space [V]. For resolving the 2nd problem, the Robust Kernel PCA by means of FastHCSv can be used.

**Eigenspace**

An eigenspace is the collection of eigenvectors associated with each eigenvalue for the linear transformation applied to the eigenvector. The linear transformation is often a square matrix (a matrix that has the same number of columns as it does rows).

Eigenspace has the amount of data points, their mean, group of support vectors over the data, and a measured spread of the data per support vector. They are calculated by means of either eigenvalue decomposition (EVD) of the covariance matrix of the data, or singular value decomposition (SVD) of the same data with identical set of support vectors. The spread magnitudes in EVD have been related to data variance, however in SVD, spread magnitudes are relational to the standard deviancy of the data.

In advance for testifying the difficulties that are of our issue, we must clarify in larger detail what the term eigenspace implies based on SVD and EVD as follows:

1. **eigenvalue decomposition**

Let \( X = [x_1, x_2, ..., x_N] \) be a group of \( N \) data points. The EVD of the covariance of the data has expressed by \((X - M1)(X - M1)^T = (1/N)ULU^T\) where \( M \) stands for the data mean, \( 1 \) is a row \( N \) 1’s, \( U \) stands for a \( nxn \) matrix of eigenvectors (support vectors), and \( L \) stands for a \( nxn \) diagonal matrix of eigenvalues (spread values).
It is frequently presumed that just those eigenvectors that stand for huge spread values are relevant, the others are eliminated through removing columns from the matrix U. Characteristically, the amount of non-zero eigenvalues is \( P \leq \min(n,N) \). This stands for covariance matrix rank of \((X - M)^T\). Hypothetically, \( P \) is selected to comprise small magnitudes along with zero values. Deflation leaves \( p \) eigenvectors in a nxp matrix \( U_{np} \) and \( p \) eigenvalues in a diagonal matrix \( L_{pp} \). \( P \) stands for dimension of the eigenspace,

\[
(X - M)(X - M)^T \approx U_{np}L_{pp}U_{np}^T
\]

For the reason that of deflation (here \( L \) is a diagonal matrix. We correspondingly have \( U_{np}^TU_{np} = I \), but \( U_{np}^TU_{np} \neq I \), eigenvectors support a subspace of dimension \( p \) set in a space of dimension \( n \).

II. Singular value decomposition

The SVD of the equivalent data, \( X \) stands for \( X - M1 = U\Sigma V^T \) that \( U \) represents a matrix of left singular vectors (support vectors), \( \Sigma \) stands for a nxN matrix which is nonzero only on its leading diagonal. These are the singular magnitudes based on spread values, while \( V \) stands for a matrix of right singular vectors that have details about the data projected into eigenspace. SVD as well as EVD are associated with the left singular vectors and eigenvectors have been alike [II].

FastHCS

Assumed an \( n \times p \) data matrix \( Y = \{y_i\}_{i=1}^n \), FastHCS calculates the mean-centered data matrix \( \bar{X} = Y - \sum_{i=1}^n (\text{ave}_{i=1}^n y_i) \), and achieves the Linear Kernel with eigenvalue decomposition (EVD) of \( \bar{X}\bar{X}^T = ULU^T \), where \( U \) stands for \( n \times r \) matrix of eigen vectors and \( L \) stands for \( r \times r \) diagonal matrix of eigen values, \( r=\text{rank}(X) \) then FastHCS will be with the \( n \times r \) matrix \( X = \bar{X}\bar{X}^T U(L)^{1/2} \). The transformation from \( Y \) to \( X \) produces without any loss of information or strength as we hold the whole constituents for non-zero eigenvalues.

However, this transformation decreases the computational rate of the consequent steps of the algorithm. In the last part of the algorithm, FastHCS undoes these transformations with the intention of the reverted parameter approximations are dependable with conformist PCA [XII] [XIV].
I-index h-subset

It stands for a subset selection standard firstly familiarized by Vakili and Schmitt (2014). It is employed to recognize an outlier free subset to act as the base of the robust PCS position with scatter estimator. The index has been considered to be unresponsive to the outliers arrangement. Consequently, the fit established by FastPCS is approximately unchanged by the existence of data outliers as quantitative robustness. Based on (Schmitt et al., 2014), PCS estimates as well have the extreme potential breakdown point as qualitative robustness. Robust position and scatter estimation are as well significant for PCA. In the PCS setting, the I-index can be realistic in the interpretations in their original dimensionality, and one attitude for realizing robust PCA can be robust PCS covariance estimate as a starting point aimed at PCA. This subsection designates how the I-index can be lengthy to the PCA context by smearing it to data projections on to subspaces.

FastHCS illustrates \( M \) random subsets of size \((q + 1)\) from \( X \) devoid of replacement, where \( M \) has specified by:

\[
M = \left[ \log \frac{0.01}{\log(1-(\frac{h}{n})^{q+1})} \right]
\] (16)

where \( h < e < n \) stands for an integer identifying the amount of uncontaminated observations. Generally, we adjust \( e = h \). However if the user has been guaranteed that the contamination rate of the tester has been less than \( (n - h)/n \), we offer this information to reduce the computational cost of running FastHCS. Symbolize these \((q + 1)\)-subsets as \( \{H_0^m\}_{m=1}^M \). SVD decomposition of the observations indicated by \( H_0^m \) can be:

\[
SVD_{i \in H_0^m}(x_i - t_0^m)/\sqrt{q} = U_q^m (L_q^m)^{1/2} (P_q^m)
\]

Where \( t_0^m \) stands for the projected center, \( L_q^m \) stands for a diagonal matrix for non-zero components \( (L_0^m)_{j=1, \ldots, q} \) stand for the descendant eigenvalues of the PCA model fitted to \( \{x_i, i \in H_0^m\} \) and the eigenvectors \( P_q^m \) stand for the first \( q \) loadings of this model. Afterward, we calculate the score matrix \( S_0^m \) with \( n \) rows:

\[
S_{0,i}^m = (x_i - t_0^m) P_{0,q}^m \quad 1 \leq i \leq n
\]

For measuring the outlyingness of an \( S_{0,i}^m \) to the members of \( \{S_{0,i}^m : i \in H_0^m\} \), we can employ its squared orthogonal distance to \( a_k^m \). The direction has been normal to the hyperplane throughout \( q \) members of \( \{S_{0,i}^m : i \in H_0^m\} \) drawn at random:

\[
d_k^2(a_k^m, S_0^m) = ((S_{0,i}^m)^T a_k^m - 1)^2 / \|a_k^m\|^2
\]

Copyright reserved © J. Mech. Cont.& Math. Sci.
Lekaa Ali Muhamed et al.
To eliminate the dependency of this measure on the direction $a_k^m$, we average it over $K$ such direction $\{a_k^m\}_{k=1}^K$:

$$D_i(H_0^m) = \frac{K}{AVE} \frac{\sum_{k=1}^K d_i^2(a_k^m, s_0^m)}{\sum_{i \in H_0^m} d_i^2(a_k^m, s_0^m)} 1 \leq i \leq n$$  \hspace{1cm} (17)

The denominator in Equation (17) regularizes these distances across the directions $a_k^m$.

We can currently refer to the computing of 1st step of FastHCS. For a specified $a$ (q + 1) subset $H_0^m$ of $\{1:n\}$ along with its equivalent matrix $S_0^m$, Algorithm 1 returns and $h$-subset $H_w^m$ of indexes of $\{1:n\}$ by means of an iterative process is based on emergent steps. In every step, $w, H_w^m$ are updated and have indexes of $W_w$ observations with minimum magnitudes of $D_i(H_{w-1}^m)$. The value of $W_w$ itself increases incrementally from $\left(\frac{(n-q-1)}{(2w)}\right) + q + 1$ to $h$ in $W$ steps. These steps don’t consume a convergence principle, so the amount of iterations $W$ need to be adjusted previously. Through experiments [XIII], raising $K$ higher than 25 or $W$ higher than 5 doesn’t markedly enhance the routine of the algorithm, so we adjust these limitations to those magnitudes.

Algorithm 1: growingStep($H_0^m, X, q$)

1. for $w$ to $W$

2. $D_i(H_{w-1}^m) = \frac{K}{AVE} \frac{\sum_{k=1}^K d_i^2(a_k^m, s_0^m)}{\sum_{i \in H_{w-1}^m} d_i^2(a_k^m, s_0^m)} 1 \leq i \leq n$

3. set $W_w = \left[\frac{(n-q-1)}{(2w)}\right] + q + 1$

4. set $H_w^m = \{i, D_i(H_{w-1}^m) \leq D_{W_w}(H_{w-1}^m)\}$

5. end for

6. $H_w^m = H^m$

After upward $M$ candidate $H^m$ 's, FastHCS estimates for each by means of a principle. We call the I-index, and fits a robust PCA exemplary to the $H^m$ with minimum magnitude of the I-index. For a specified $h$-subset $H^m$ and direction $a_k^m$, we describe a subset $H_k^m$ which is optimum concerning $a_k^m$ in the sense that it
indexes the h observations with the minimum magnitude of $d_i^2(a_k^m, S_0^m)$. More specifically, signifying $d(h)$ with the $h^{th}$ order statistic of a vector $d$, we have:

$$H_k^m = \{d_i^2(a_k^m, S_0^m) \leq d^2_{(h)}(a_k^m, S_0^m)\}$$

At that point, we describe the I-index of an $H^m$ along $a_k^m$ as:

$$I(H^m, S_0^m, a_k^m) = \log \left( \frac{\text{ave}_{i \in H^m} d_i^2(a_k^m, S_0^m)}{\text{ave}_{i \in H_k^m} d_i^2(a_k^m, S_0^m)} \right)$$

(18)

with the agreement that $\log(0/0) := 0$. The $I(H^m, S_0^m, a_k^m)$ measure has been continually positive and upturns the less members $H^m$ shares with $H_k^m$ along the direction $a_k^m$. This is for the reason that, for a specified direction $a_k^m$, the members of $H_k^m$ not in $H^m$ will drop the denominator in Equation (18) with no effect on the numerator, intensifying the inclusive ratio. As in the emergent steps, we eliminate the dependency of Equation (18) on the directions $a_k^m$ in view of the average over K directions:

$$I(H^m, S_0^m) = \frac{1}{K} \sum_{k=1}^{K} I(H^m, S_0^m, a_k^m)$$

Lastly, FastHCS chooses as $H^l$ for candidate h-subset $H^m$ with lowermost I-index.

For specified $H^l$, we signify the PCA parameters for $H^l$ as $(t^l, L^l q^l, P^l q^l)$ and get them as follows:

$$U^l (L^l)^T$$

Where $t^l = \text{ave}_{i \in H^l} y_i$. FastHCS calculates these considerations on the complete space of the data set, $Y$, relatively as compared with the space of $S_0^l$, to upturn their accurateness. Algorithm 2 gives details about I-index step of FastHCS.[10].

Algorithm 2: I Step (x,q)

1. for m=1 to M
2. $H_0^m = \{\text{random} (q + 1) - \text{subset from 1:n}\}$
3. $H^m = \text{growing steps}(H_0^m, X, q)$
4. $I(H^m, S_0^m) = \frac{1}{K} \sum_{k=1}^{K} I(H^m, S_0^m, a_k^m)$
5. end for
6. $H^l = \arg \min_{H^1 \ldots H^m}(H^m, S_0^m)$
7. return $(t^l, L^l q^l, P^l q^l)$

Copyright reserved © J. Mech. Cont. & Math. Sci.
Lekaa Ali Muhamed et al.
III. Results and Discussion

Simulations were performed using the R program according to the following steps:

1. To generate the study data from the natural distribution polluted by using Box-Muller formula with a pollution rate (0.25) to include three experiments that differ in terms of number of variables and sample sizes. These experiments are:

   The first experiment (N = 15, p = 10), the second experiment (n = 22, p = 15) and finally the third experiment (n = 30, p = 20).

2. Finding the estimation of non-linear methods by the kernel matrix using Laplacian. The calculation of the bandwidth was based on the Scott formula in calculating the smoothing parameter (h) (h = 2).

3. For the robust method, the estimates were obtained by the best subsets from the simple smoothing of data and based on the partial sampling of samples (0.75) in the three trials.

4. The application of the formulas for the Eigen Values corresponding to the correlation matrix, the finding of the principal components and the percentages of discrepancies explained according to the methods and formulas detail with in the theoretical aspect of this research.

Where : KPCA l= Laplacian Kernel , RKPCA= FastHCS

Table 1. Eigen Value and variances when n=15 p=10

| PC  | Eigen Value | Proportion Variance | Cumulative Variance |
|-----|-------------|---------------------|---------------------|
|     | PCA         | KPCA 1              | RKPCA              |
|     |             | PCA                 | KPCA 1             | RKPCA              | PCA                 | KPCA 1              | RKPCA              |
| Pc1 | 3.67        | 0.40                | 0.45               | 0.40               | 0.37                | 0.45               |
|     | 1.39        | 0.22                | 0.09               | 0.71               | 0.77                | 0.76               |
| Pc2 | 1.82        | 0.08                | 0.05               | 0.08               |                      |                    |
| Pc3 | 1.16        |                     |                    |                    |                      |                    |

Copyright reserved © J. Mech. Cont. & Math. Sci.
Lekaa Ali Muhammed et al.
Table 2. Eigen Value and variances when n=22 p=15

| PC   | Eigen Value | Proportion Variance | Cumulative Variance |
|------|-------------|---------------------|---------------------|
|      | PC          | KPPCA               | RKPCA               | PCA     | KPCA   | RKPCA   | PCA     | KPCA   | RKPCA   |
| Pe1  | 5.4         | 7.1                 | 14.3                | 0.36    | 0.48   | 0.39    | 0.36    | 0.48   | 0.39    |
| Pe2  | 2.6         | 4.3                 | 6.94                | 0.18    | 0.29   | 0.19    | 0.54    | 0.77   | 0.58    |
| Pe3  | 1.6         | 1.2                 | 3.61                | 0.11    | 0.09   | 0.10    | 0.65    | 0.85   | 0.68    |
| Pe4  | 1.1         | 7.1                 | 2.71                | 0.08    | 0.07   | 0.73    | 0.75    |        |         |
| Pe5  | 1.79        |                     |                     | 0.05    |        |         |         | 0.80    |         |
| Pe6  | 1.77        |                     |                     | 0.05    |        |         |         | 0.85    |         |
| Pe7  | 1.49        |                     |                     | 0.04    |        |         |         | 0.89    |         |
| Pe8  | 1.04        |                     |                     | 0.04    |        |         |         | 0.93    |         |

Table 3. Eigen Value and variances when n=30 p=20

| PC   | Eigen Value | Proportion Variance | Cumulative Variance |
|------|-------------|---------------------|---------------------|
|      | PC          | KPPCA               | RKPCA               | PCA     | KPCA   | RKPCA   | PCA     | KPCA   | RKPCA   |
| Pc1  | 5.7         | 8.1                 | 12.6                | 0.29    | 0.41   | 0.31    | 0.29    | 0.41   | 0.31    |
| Pc2  | 2.8         | 5.3                 | 6.17                | 0.14    | 0.27   | 0.15    | 0.43    | 0.68   | 0.46    |
| Pc3  | 1.9         | 1.7                 | 3.67                | 0.10    | 0.09   | 0.09    | 0.53    | 0.76   | 0.55    |
| Pc4  | 1.4         | 1.2                 | 3.50                | 0.07    | 0.06   | 0.09    | 0.60    | 0.83   | 0.64    |
| Pc5  | 1.3         | 1.0                 | 2.57                | 0.07    | 0.05   | 0.06    | 0.67    | 0.88   | 0.70    |
| Pc6  | 1.2         | 9.1                 | 2.31                | 0.06    | 0.06   | 0.73    | 0.76    |        |         |
| Pc7  | 1.0         | 9.1                 | 1.78                | 0.05    | 0.04   | 0.78    | 0.80    |        |         |
| Pc8  | 1.63        |                     |                     | 0.04    |        |         |         | 0.84    |         |
| Pc9  | 1.28        |                     |                     | 0.03    |        |         |         | 0.87    |         |
| Pc10 | 1.13        |                     |                     | 0.03    |        |         |         | 0.90    |         |

Copyright reserved © J. Mech. Cont.& Math. Sci.  
Lekaa Ali Muhamed et al.
IV. Conclusion:

A multivariate data with higher dimensions and outliers the using of PCA will leading to miss understand results, In our simulations we found that the performance of FastHCS is better than PCA and KPCA Of the total variance Explanation in all three sample of simulation. Because Fast HCS using first Linear Kernel to solve nonlinear data then use robust procedure to eliminate outlier.

References:

I. Fan J, Chow TW. Exactly Robust Kernel Principal Component Analysis. IEEE transactions on neural networks and learning systems. 2019 Apr 29.

II. Hall P, Marshall D, Martin R. Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition. Image and Vision Computing. (2002):1009-1016.

III. Huang HH, Yeh YR. An iterative algorithm for robust kernel principal component analysis. Neurocomputing. 2011 Nov 1;74(18):3921-30.

IV. Lee JM, Yoo C, Choi SW, Vanrolleghem PA, Lee IB. Nonlinear process monitoring using kernel principal component analysis. Chemical engineering science. 2004 Jan 1;59(1):223-34.

V. Lu C, Zhang T, Zhang R, Zhang C. Adaptive robust kernel PCA algorithm. In 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). 2003 Apr 6 (Vol. 6, pp. VI-621). IEEE.

VI. Mika S, Schölkopf B, Smola AJ, Müller KR, Scholz M, Rätsch G. Kernel PCA and de-noising in feature spaces. In Advances in neural information processing systems 1999 (pp. 536-542).

VII. Neil HT. Applied multivariate analysis. Board.(NY, USA),Springer, (2002).
VIII. PITOTY C. EMPIRICAL STUDY ON BANDWIDTH OPTIMIZATION FOR KERNEL PCA IN THE K-MEANS CLUSTERING OF NON-LINEARLY SEPARABLE DATA. Journal of Theoretical & Applied Information Technology. 2017 Oct 31;95(20).

IX. Samuel RT, Cao Y. Nonlinear process fault detection and identification using kernel PCA and kernel density estimation. Systems Science & Control Engineering. 2016 Jan 1;4(1):165-74.

X. Schölkopf B, Smola A, Müller KR. Nonlinear component analysis as a kernel eigenvalue problem. Neural computation. 1998 Jul 1;10(5):1299-319.

XI. Shams MA. Fault Identification using Kernel Principle Component Analysis. IFAC Proceedings Volumes. 2011 Jan 1;44(1):4320-5.

XII. Schmitt E, Vakili K. The FastHCS algorithm for robust PCA. Statistics and Computing. 2016 Nov 1;26(6):1229-42.

XIII. Schmitt E, Vakili K. Finding multivariate outliers with FastPCS. Computational Statistics & Data Analysis. 2014 Jan 1;69:54-66.

XIV. Wu W, Massart DL, De Jong S. The kernel PCA algorithms for wide data. Part I: theory and algorithms. Chemometrics and Intelligent Laboratory Systems. 1997 Apr 1;36(2):165-72.