Department of Social Systems and Management

Discussion Paper Series

**No. 1195**

# The New Approach for Kernel Principal Component Regression

by

Antoni Wibowo and Yoshitsugu Yamamoto

February 2008

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

# The New Approach for Kernel Principal Component Regression [*]

Antoni Wibowo [†]and Yoshitsugu Yamamoto[‡]

*Graduate School of Systems and Information Engineering,*
*University of Tsukuba,1-1-1 Tennodai, Tsukuba 305-8573, Japan.*

February 13, 2008

**Abstract**

In regression analysis, existence of multicollinearity (collinearity) on given data, say $\mathbf{X}$, can seriously deteriorate the result by the linear regression model. To avoid the effect of multicollinearity, we can use the principal component regression (PCR) [17]. However, the PCR has difficulties on its applications. To overcome such a drawback, Hoegaerts *et al.* [4], Jade *et al.* [5], and Rosipal *et al.* [11, 12, 13] used a technique, called the kernel principal component regression (KPCR). However, their KPCR [4, 5, 11, 12, 13] still have some drawbacks, i.e., the procedure to derive their KPCR and the choice rule of the retained number of PC's to avoid the effect of multicollinearity.

To overcome the above drawbacks, we propose a new approach for the KPCR. Firstly, we generalized the linear regression model in [3, 6, 8, 10, 16, 17] by relaxing the linear independence assumption. Secondly, we show that the PCR can also be used to reduce the effect collinearity on $\mathbf{X}$. Finally, we propose a new approach for KPCR by using the above relaxing assumption. In the new approach for KPCR, we propose an algorithm that it can automatically obtain the retained number of nonlinear PC's to avoid the effect of multicollinearity (collinearity).

In our case studies, we compared the capabilities of the linear regression, the nonlinear regressions using the Gompertz function, the previous KPCR and the new KPCR. For the real data, the stock of cars in the Netherlands (in period 1965-1989) and the weight of a certain kind of female chickens [7], the results of the new KPCR are better than the linear regression and the nonlinear regressions using the Gompertz function.

## 1 Introduction

*Regression analysis* is a model of the relationship between a single random variable $Y$, called the *response variable*, and independent variables $x_1, x_2, \cdots, x_p$. The independent variables are called the *regressor variables.* The regression analysis is one of the important techniques in multivariate data analysis. The multiple linear regression has been extensively applied in almost every field, including engineering, the physical and chemical sciences, economics, management, life and biological sciences, and the social sciences [10].

[*]The authors thank Maiko Shigeno and Hideo Suzuki for comments and suggestions.

[†]Email address: wibowo@sk.tsukuba.ac.jp.

[‡]Email address: yamamoto@sk.tsukuba.ac.jp.

The *multiple linear regression model* with $p$ regressors is given by

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \epsilon. \tag{1.1}$$

The parameters $\beta_j, (j = 0, 1, \cdots, p)$ are called the *regression coefficients* and $\epsilon$ is a random variable and called the *random error.* It is assumed that the values of $x_1, x_2, \cdots, x_p$ are chosen by an experimenter and $\beta_j$ are unknown. The term linear is used since Eq. (1.1) is a linear function of the regression coefficients $\beta_j$.

Assume that $Y_i$ is the $i$th response variable on the $i$th observation $(i = 1, 2, \cdots, N)$, $x_{ij} \in \mathbb{R}$ is the $i$th observation of regressor $x_j$ and $\epsilon_i$ is the $i$th random error on the $i$th observation, where $\mathbb{R}$ is the set of real number. We denote $\mathbf{x}_i^T = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{ip} \end{pmatrix}$, $\mathbf{Y} = (Y_1 \quad Y_2 \quad \cdots \quad Y_N)^T$, $\tilde{\mathbf{X}} = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_N)^T$, $\mathbf{X} = \begin{pmatrix} \mathbf{1}_N & \tilde{\mathbf{X}} \end{pmatrix}$, $\boldsymbol{\beta} = (\beta_0 \quad \beta_1 \quad \cdots \quad \beta_p)^T$, and $\boldsymbol{\epsilon} = (\epsilon_1 \quad \epsilon_2 \quad \cdots \quad \epsilon_N)^T$; where sizes of $\mathbf{x}_i$, $\mathbf{Y}$, $\tilde{\mathbf{X}}$, $\mathbf{X}$, $\boldsymbol{\beta}$ and $\boldsymbol{\epsilon}$ are $p \times 1$, $N \times 1$, $N \times p$, $N \times (p+1)$, $(p+1) \times 1$ and $N \times 1$, respectively, and $\mathbf{1}_N = (1 \quad 1 \quad \cdots \quad 1)_{N\times 1}^T$. The vector $\mathbf{x}_i^T$ denotes the transpose of the vector $\mathbf{x}_i$. The *ordinary multiple linear regression model* corressponding to Eq. (1.1) is written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{1.2}$$

It is assumed that the expected value of $\boldsymbol{\epsilon}$, denoted by $E(\boldsymbol{\epsilon})$, is equal $\mathbf{0}$ and the variance of $\boldsymbol{\epsilon}$, denoted by $E(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T)$, is equal $\sigma^2 I_N$. The matrix $I_N$ denotes the $N \times N$ identity matrix.

The problem of regression analysis is to find the estimator of $\boldsymbol{\beta}$, say $\hat{\boldsymbol{\beta}} = \begin{pmatrix} \hat{\beta}_0 & \hat{\beta}_1 & \cdots & \hat{\beta}_p \end{pmatrix}^T$, such that $\|\boldsymbol{\epsilon}\|^2$ is minimized. The solution can be found by solving the following equation

$$\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{Y}. \tag{1.3}$$

The Eq. (1.3) is called the *normal system of* system $\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{Y}$. Note that, $\mathbf{X}^T\mathbf{X}$ is a symmetric and positive semidefinite matrix. Hence, the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are real and non-negative [1]. We say that *multicollinearity* exists on given data, say $\mathbf{X}$, if $\mathbf{X}^T\mathbf{X}$ is a near singular matrix, i.e, if all eigenvalues of $\mathbf{X}^T\mathbf{X}$ are positive numbers and some eigenvalues of $\mathbf{X}^T\mathbf{X}$ are near zero. We say that *collinearity* exists on $\mathbf{X}$ if $\mathbf{X}^T\mathbf{X}$ is a singular matrix, i.e., if some eigenvalues of $\mathbf{X}^T\mathbf{X}$ are zero [10, 16]. If collinearity exists on $\mathbf{X}$ then there are infinitely many solutions of Eq. (1.3), which makes it difficult to choose the best linear multiple regression model in this case. This implication is known as the *effect of collinearity.*

In [3, 6, 8, 10, 16, 17], they restricted the ordinary multiple linear regression model to the case where the column vectors of $\mathbf{X}$ are linearly independent. In this case, the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are positive and real numbers [1]. It implies the collinearity never exists on $\mathbf{X}$. The variance of $\hat{\beta}_j$ for $j = 0, 1, \cdots, p$, denoted by $Var(\hat{\beta}_j)$, is given by

$$Var(\hat{\beta}_j) = \sigma^2((\mathbf{X}^T\mathbf{X})^{-1})_{j+1,j+1}, \qquad j = 0, 1, \cdots, p. \tag{1.4}$$

where $(\mathbf{X}^T\mathbf{X})^{-1}$ is the inverse of the matrix $\mathbf{X}^T\mathbf{X}$. If multicollinearity exists on $\mathbf{X}$ then $Var(\hat{\beta}_j)$ will be a large number [17]. If multicollinearity exists on $\mathbf{X}$ and under the assumption $\varepsilon_i$ is normally distribution then tests for inferences $\beta_j$ $(j =$

$0, 1, \cdots, p$), have low power and confidence interval will be large. It will be difficult to decide if a variable $x_j$ makes a significant contribution to the regression [17]. These implications are known as the *effect of multicollinearity.*

Suppose that $\mathbf{y} = (y_1 \quad y_2 \quad \cdots \quad y_N)^T \in \mathbb{R}^N$ is the observation corresponding to $\mathbf{Y}$. Under the assumption the column vectors of $\mathbf{X}$ are linearly independent, the estimator of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}, \tag{1.5}$$

see for example [3, 6, 8, 10, 16, 17]. The *prediction multiple linear regression model* corresponding to the regressors variable $x_1, x_2, \cdots, x_p$; say $\hat{y}$, is given by

$$\hat{y} := \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j. \tag{1.6}$$

The *vector of prediction values* $\hat{y}_i$ corresponding to the observed values $y_i$, say $\hat{\mathbf{y}}$, is given by

$$\hat{\mathbf{y}} := \mathbf{X}\hat{\boldsymbol{\beta}}. \tag{1.7}$$

The error between $\mathbf{y}$ and $\hat{\mathbf{y}}$, say $\mathbf{e}$, is given by

$$\mathbf{e} := \mathbf{y} - \hat{\mathbf{y}}, \tag{1.8}$$

the *root mean square error* (RSME) between $\mathbf{y}$ and $\hat{\mathbf{y}}$ is given by

$$RMSE := \frac{1}{\sqrt{N}}\|\mathbf{e}\|, \qquad \text{where } \|\mathbf{e}\| = (\mathbf{e}^T\mathbf{e})^{\frac{1}{2}}. \tag{1.9}$$

To avoid the effect of multicollinearity, we transform the regression model (1.2) to another fashion by using an orthogonal matrix. The technique is called the *principal component regression (PCR)* [10, 17]. However, the PCR model still has a linear fashion. Since the most real problems are nonlinear, the PCR has difficulties on its applications.

To overcome such a drawback, Hoegaerts *et al.* [4], Jade *et al.* [5], and Rosipal *et al.* [11, 12, 13] used a technique, called the *kernel principal component regression (KPCR).* They transformed $\mathbf{x}_i$ ($i = 1, 2, \cdots, N$), by using a function $\psi : \mathbb{R}^p \to \mathbb{F}$, where $\mathbb{F} \subseteq \mathbb{R}^{p_F}$ and $p_F \leq \infty$. The set $\mathbb{F}$ is called the *feature space* which is a higher dimensional Euclidean space. Hence, the image of $\mathbf{x}_i$ in $\mathbb{F}$ is given by $\psi(\mathbf{x}_i)$. Note that, the function $\psi$ is not *explicitly defined*, i.e., the function $\psi$ is not written explicitly in terms of its independent variables. We define that a function $\varphi$ is said to be a *symmetric function* if $\varphi(\mathbf{w}_i, \mathbf{w}_j) = \varphi(\mathbf{w}_j, \mathbf{w}_i)$ for every $\mathbf{w}_i, \mathbf{w}_j \in \mathbb{R}^p$ and is said to be a *positive semidefinite function* if for every $m \in \mathbb{N}$, where $\mathbb{N}$ is set of natural number, such that $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_m \in \mathbb{R}^p$ gives rise to a positive semidefinite matrix $\mathbf{W} = (\varphi(\mathbf{w}_i, \mathbf{w}_j))_{i,j=1,2,\cdots,m}$, see [15] for the detailed discussion. The function $\psi$ is derived from another function, say $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, where $\kappa$ is a symmetric, continuous and positive semidefinite function [2, 9, 14]. The function $\kappa$ is called the *kernel* function.

Define a matrix

$$\boldsymbol{\Psi} := (\psi(\mathbf{x}_1) \quad \psi(\mathbf{x}_2) \quad \cdots \quad \psi(\mathbf{x}_N))^T,$$

$$\mathbf{K} := \boldsymbol{\Psi}\boldsymbol{\Psi}^T,$$

where size of $\mathbf{\Psi}$ and $\mathbf{K}$ are $N \times p_F$ and $N \times N$, respectively. Hoegaerts *et al.* [4], Jade *et al.* [5], and Roman *et al.* [11, 12, 13] defined the *standard multiple linear regression model in the feature space* as the following model

$$\mathbf{Y} = \mathbf{\Psi}\boldsymbol{\nu} + \boldsymbol{\epsilon^*}, \tag{1.10}$$

where $\boldsymbol{\nu} = \begin{pmatrix}\nu_1 & \nu_2 & \cdots & \nu_{p_F}\end{pmatrix}^T$ is a vector of regression coefficients in the feature space and $\boldsymbol{\epsilon^*}$ is a vector of random error in the feature space. They [4, 5, 11, 12, 13] assumed that $\sum_{i=1}^{N} \psi(\mathbf{x}_i) = \mathbf{0}$, $E(\boldsymbol{\epsilon^*}) = \mathbf{0}$, $E(\boldsymbol{\epsilon^* \epsilon^{*T}}) = \delta^2 \mathbf{I}_N$ and $N \ll p_F$. Suppose that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_{\hat{r}} \geq \mu_{\hat{r}+1} \geq \cdots \geq \mu_{p_F}$ are eigenvalues of $\mathbf{\Psi}^T\mathbf{\Psi}$. Let $\mathbf{V} = (\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_{p_F})$ be the matrix of eigenvectors $\mathbf{v}_i$ of $\mathbf{\Psi}^T\mathbf{\Psi}$ and $\mathbf{V} = \mathbf{V}^{-1}$ where $\mathbf{v}_i$ correponding to $\mu_i$ $(i = 1, \cdots, p_F)$. They rewrite the model (1.10) as

$$\mathbf{Y} = \mathbf{B}\boldsymbol{w} + \boldsymbol{\epsilon^*}, \tag{1.11}$$

where $\mathbf{B} = \mathbf{\Psi V}$ and $\boldsymbol{w} = (w_1 \quad w_2 \quad \cdots \quad w_{p_{p_F}})^T = \mathbf{V}^T\boldsymbol{\nu}$. They stated the estimator of $\boldsymbol{w}$, say $\hat{\boldsymbol{w}} := (\hat{w}_1 \quad \hat{w}_2 \quad \cdots \quad \hat{w}_{p_{p_F}})^T$, is given by

$$\hat{\boldsymbol{w}} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{y}. \tag{1.12}$$

Further, the estimator of $\boldsymbol{\nu}$, say $\hat{\boldsymbol{\nu}}$, is written as

$$\hat{\boldsymbol{\nu}} = \mathbf{V}\hat{\boldsymbol{w}} = \sum_{i=1}^{\hat{p}_F} \mu_i^{-1}\mathbf{v}_i\mathbf{v}_i^T\mathbf{\Psi}^T\mathbf{y}, \tag{1.13}$$

and its corresponding covariance matrix as

$$cov(\hat{\boldsymbol{\nu}}) = \delta^2 \sum_{i=1}^{\hat{p}_F} \mu_i^{-1}\mathbf{v}_i\mathbf{v}_i^T. \tag{1.14}$$

It is evident that from (1.14) that the influence of small eigenvalues can significantly increase to overall variance of the estimator of $\boldsymbol{\nu}$. To avoid the effect of multicollinearity, PCR deletes some eigenvectors of $\mathbf{\Psi}^T\mathbf{\Psi}$ corresponding to small values of the eigenvalues $\mu_i$.[1] Let $\boldsymbol{\varsigma}_j := (\varsigma_{j1} \quad \varsigma_{j2} \quad \cdots \quad \varsigma_{jN})^T$ be an eigenvector of $\mathbf{K}$ corresponding to $\mu_j \neq 0$ for some $j \in \{1, \cdots, p_F\}$. Using the first $\hat{r}$ of $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{p_F}$, they stated that their KPCR is written as

$$h(\mathbf{x}) = \sum_{i=1}^{N} a_j \kappa(\mathbf{x}_i, \mathbf{x}) + d, \tag{1.15}$$

where $a_i = \sum_{k=1}^{\hat{r}} \hat{w}_k \varsigma_{ik}$ for $i = 1, \cdots, N$, and $d$ is a bias term. The term $d$ will vanish when $\sum_{i=1}^{N} Y_i = 0$ [4, 5, 11, 12, 13]. The number $\hat{r}$ is called the *retained number of nonlinear PC's* in the KPCR model.

Note that, the above claims are true when column vectors of $\mathbf{\Psi}$ are linearly independent. A question arises: does the matrix $(\mathbf{B}^T\mathbf{B})^{-1}$ exist? Since they assumed $N \ll p_F$, this implies the column vectors of $\mathbf{\Psi}$ are linear dependent. It is well known that rank($\mathbf{B}^T\mathbf{B}$)=rank($\mathbf{\Psi}^T\mathbf{\Psi}$), where rank($\mathbf{B}^T\mathbf{B}$) stands for the rank of the matrix

[1] The relation of eigenvalues and eigenvectors of the matrices $\mathbf{\Psi}^T\mathbf{\Psi}$ and $\mathbf{K}$ were introduced by Scholkopf *et al.* [15] (see Appendix I)

$\mathbf{B}^T\mathbf{B}$. Since column vectors of $\mathbf{\Psi}$ are linear dependent, then rank($\mathbf{B}^T\mathbf{B}$) $< p_F$. This implies $(\mathbf{B}^T\mathbf{B})$ is not invertible. Hence, the matrix $(\mathbf{B}^T\mathbf{B})^{-1}$ does not exist [1] and we have a contradiction. Implication of the contradiction is the choice rule of the retained number of PC's to avoid the effect of multicollinearity becomes unclear. Another question will also arise. How to choose the function $\kappa$ such that the column vectors of $\mathbf{\Psi}$ are linearly independent? It is a difficult task. The next question, how to handle if the matrix $(\mathbf{B}^T\mathbf{B})^{-1}$ does not exist? Yet, their work did not consider it.

To overcome the above drawbacks, we propose a new approach for the KPCR. Firstly, we generalized the linear regression model in [3, 6, 8, 10, 16, 17] by relaxing the linear independence assumption. Stated in other words, our model can be used whether the column vectors of $\mathbf{X}$ are linearly independent or linearly dependent. Secondly, we show that the PCR can also be used to reduce the effect of collinearity. Finally, we propose a new approach for KPCR by using the above relaxing assumption. The procedure to derive the new approach for the KPCR is straightforward as the procedure to derive the PCR is. In the new approach for KPCR, we propose an algorithm that it can automatically obtain the retained number of PC's to avoid the effect of multicollinearity (collinearity). We refer the KPCR proposed by Hoegaerts *et al.* [4], Jade *et al.* [5], and Rosipal *et al.* [11, 12, 13] as the *previous KPCR* and we refer the new approach for the KPCR as the *new KPCR*.

This manuscript is organized as follows: Section 2, we review the PCR and show that the PCR can also be used to reduce the effect of collinearity. In Section 3, the detailed of the new KPCR model will be discussed. Afterwards, we construct an algorithm for the new KPCR. In Section 4, we compare the capabilities of the linear regression, the previous KPCR, the non linear regression based on Gompertz function and the new KPCR. Conclusions are given in Section 5. Finally, the proofs of the some Theorems and claims and the MATLAB code for the new KPCR algorithm are given in Appendix.

## 2 Principal Component Regression

The *standard centered multiple linear regression model* corresponding to Eq. (1.2) is given by

$$\mathbf{Y}_o = \mathbf{Z}\tilde{\boldsymbol{\beta}} + \boldsymbol{\epsilon}_o, \tag{2.1}$$

where $\mathbf{Z} = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\tilde{\mathbf{X}}$, $\boldsymbol{\epsilon}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{\epsilon}$, $\tilde{\boldsymbol{\beta}} = (\beta_1 \quad \beta_2 \quad \cdots \quad \beta_p)^T$, $\mathbf{Y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{Y}$ and $\mathbf{y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y}$ . We define $\bar{y} = \frac{1}{N}\mathbf{1}_N^T\mathbf{y}$ and $\bar{x}_l = \frac{1}{N}\mathbf{1}_N^T\mathbf{x}_l$ for $l = 1, 2, \cdots, p$ and let $\hat{p}$ be the rank of $\mathbf{Z}^T\mathbf{Z}$ where $\hat{p} \leq \min(N, p)$.

Since $\mathbf{Z}^T\mathbf{Z}$ is a symmetric and positive semidefinite matrix, then the eigenvalues of $\mathbf{Z}^T\mathbf{Z}$ are real and non-negative. Suppose that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r \geq \lambda_{r+1} \geq \cdots \geq \lambda_{\hat{p}} > \lambda_{p\hat{+}1} = \cdots = \lambda_p = 0$ are eigenvalues of $\mathbf{Z}^T\mathbf{Z}$. Let $\mathbf{A} = (\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_p)$ be the matrix of eigenvectors $\mathbf{a}_l$ of $\mathbf{Z}^T\mathbf{Z}$ and $\mathbf{A} = \mathbf{A}^{-1}$, then

$$\mathbf{A}^T\mathbf{Z}^T\mathbf{Z}\mathbf{A} = \mathbf{D}$$

where $\mathbf{a}_l$ corresponds to $\lambda_l$ for $l = 1, 2, \cdots, p$, and

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{(\hat{p})} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{pmatrix},$$

$$\mathbf{D}_{(\hat{p})} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \lambda_{\hat{p}} \end{pmatrix},$$

and $\mathbf{O}$ is a zero matrix.

Note that, $\mathbf{A}$ is orthogonal matrix [1]. This implies $\mathbf{A}\mathbf{A}^T = \mathbf{I}_p$. Further, we can rewrite the model (2.1) as

$$\mathbf{Y}_o = \mathbf{U}\boldsymbol{\omega} + \boldsymbol{\epsilon}_o, \tag{2.2}$$

where $\mathbf{U} = \mathbf{Z}\mathbf{A}$ and $\boldsymbol{\omega} = \mathbf{A}^T\tilde{\boldsymbol{\beta}}$. Rewriting

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_{(\hat{p})} & \mathbf{U}_{(p-\hat{p})} \end{pmatrix} \text{ and } \boldsymbol{\omega} = \begin{pmatrix} \boldsymbol{\omega}^T_{(\hat{p})} & \boldsymbol{\omega}^T_{(p-\hat{p})} \end{pmatrix}^T,$$

where sizes of $\mathbf{U}_{(\hat{p})}$, $\mathbf{U}_{(p-\hat{p})}$, $\boldsymbol{\omega}_{(\hat{p})}$, and $\boldsymbol{\omega}_{(p-\hat{p})}$ are $N \times \hat{p}$, $N \times (p-\hat{p})$, $\hat{p} \times 1$ and $(p-\hat{p}) \times 1$, respectively. Since $\mathbf{D} = \mathbf{A}^T\mathbf{Z}^T\mathbf{Z}\mathbf{A} = \mathbf{U}^T\mathbf{U}$, we obtain

$$\mathbf{U}^T_{(\hat{p})}\mathbf{U}_{(\hat{p})} = \mathbf{D}_{(\hat{p})},$$
$$\mathbf{U}^T_{(p-\hat{p})}\mathbf{U}_{(p-\hat{p})} = \mathbf{O},$$

and

$$\mathbf{U}^T_{(\hat{p})}\mathbf{U}_{(p-\hat{p})} = \mathbf{O}.$$

The above model (2.2) can now be written as

$$\mathbf{Y}_o = \mathbf{U}_{(\hat{p})}\boldsymbol{\omega}_{(\hat{p})} + \mathbf{U}_{(p-\hat{p})}\boldsymbol{\omega}_{(p-\hat{p})} + \boldsymbol{\epsilon}_o. \tag{2.3}$$

Since $\|(\mathbf{U}_{(p-\hat{p})}\boldsymbol{\omega}_{(p-\hat{p})})^T(\mathbf{U}_{(p-\hat{p})}\boldsymbol{\omega}_{(p-\hat{p})})\| = 0$, we obtain $\mathbf{U}_{(p-\hat{p})}\boldsymbol{\omega}_{(p-\hat{p})}$ is equal to $\mathbf{0}$. The model (2.3) reduces to

$$\mathbf{Y}_o = \mathbf{U}_{(\hat{p})}\boldsymbol{\omega}_{(\hat{p})} + \boldsymbol{\epsilon}_o. \tag{2.4}$$

This result shows that the effect of collinearity on $\mathbf{Z}$ is reduced by transforming the orthogonal matrix $\mathbf{A}$.

Further, we assume that $\lambda_{r+1} \approx 0, \lambda_{r+2} \approx 0, \cdots, \lambda_{\hat{p}} \approx 0$ and let

$$\mathbf{U}_{(\hat{p})} = \begin{pmatrix} \mathbf{U}_{(r)} & \mathbf{U}_{(\hat{p}-r)} \end{pmatrix}, \boldsymbol{\omega}_{(\hat{p})} = \begin{pmatrix} \boldsymbol{\omega}^T_{(r)} & \boldsymbol{\omega}^T_{(\hat{p}-r)} \end{pmatrix}^T$$

and

$$\mathbf{D}_{(\hat{p})} = \begin{pmatrix} \mathbf{D}_{(r)} & \mathbf{O} \\ \mathbf{O} & \mathbf{D}_{(\hat{p}-r)} \end{pmatrix},$$

where

$$\mathbf{D}_{(r)} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \lambda_r \end{pmatrix},$$

$$\mathbf{D}_{(\hat{p}-r)} = \begin{pmatrix} \lambda_{r+1} & 0 & \cdots & 0 \\ 0 & \lambda_{r+2} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \lambda_{\hat{p}} \end{pmatrix},$$

and sizes of $\mathbf{U}_{(r)}$, $\mathbf{U}_{(\hat{p}-r)}$, $\boldsymbol{\omega}_{(r)}$, and $\boldsymbol{\omega}_{(\hat{p}-r)}$ are $N \times r$, $N \times (\hat{p}-r)$, $r \times 1$ and $(\hat{p}-r) \times 1$, respectively. Since $\mathbf{D}_{(\hat{p})} = \mathbf{U}_{(\hat{p})}^T \mathbf{U}_{(\hat{p})}$, we obtain that

$$\begin{aligned} \mathbf{U}_{(r)}^T \mathbf{U}_{(r)} &= \mathbf{D}_{(r)}, \\ \mathbf{U}_{(\hat{p}-r)}^T \mathbf{U}_{(\hat{p}-r)} &= \mathbf{D}_{(\hat{p}-r)} \end{aligned}$$

and

$$\mathbf{U}_{(r)}^T \mathbf{U}_{(\hat{p}-r)} = \mathbf{O}.$$

The model (2.4) can now be written as

$$\mathbf{Y}_o = \mathbf{U}_{(r)}\boldsymbol{\omega}_{(r)} + \mathbf{U}_{(\hat{p}-r)}\boldsymbol{\omega}_{(\hat{p}-r)} + \boldsymbol{\epsilon}_o. \tag{2.5}$$

To avoid effects of multicollinearity on $\mathbf{Z}$, we drop the term $\mathbf{U}_{(\hat{p}-r)}\boldsymbol{\omega}_{(\hat{p}-r)}$ in the model (2.5) [17] and obtain

$$\mathbf{Y}_o = \mathbf{U}_{(r)}\boldsymbol{\omega}_{(r)} + \boldsymbol{\epsilon}_o^*, \tag{2.6}$$

where $\boldsymbol{\epsilon}_o^*$ is a random vector influenced by dropping $\mathbf{U}_{(\hat{p}-r)}\boldsymbol{\omega}_{(\hat{p}-r)}$ in the model (2.5). The model (2.6) shows that the effects of collinearity and multicollinearity on $\mathbf{Z}$ are reduced by transforming the orthogonal matrix $\mathbf{A}$.[2]

Note that, $\mathbf{U}_{(r)}^T \mathbf{U}_{(r)} = \mathbf{D}_{(r)}$. This implies $\text{rank}(\mathbf{U}_{(r)}^T \mathbf{U}_{(r)}) = r$. The last statement implies $\mathbf{U}_{(r)}^T \mathbf{U}_{(r)}$ is invertible. Hence, the estimator of $\boldsymbol{\omega}_{(r)}$, say $\hat{\boldsymbol{\omega}}_{(r)}$, is

$$\hat{\boldsymbol{\omega}}_{(r)} = (\mathbf{U}_{(r)}^T \mathbf{U}_{(r)})^{-1} \mathbf{U}_{(r)}^T \mathbf{y}_o. \tag{2.7}$$

Since $\mathbf{U}_{(r)}^T \mathbf{y} = \mathbf{U}_{(r)}^T \mathbf{y}_o$ (see Appendix III), we obtain

$$\hat{\boldsymbol{\omega}}_{(r)} = (\mathbf{U}_{(r)}^T \mathbf{U}_{(r)})^{-1} \mathbf{U}_{(r)}^T \mathbf{y}. \tag{2.8}$$

The prediction value of $\mathbf{y}$, say $\tilde{\mathbf{y}}$, is given by

$$\tilde{\mathbf{y}} = \bar{y}\mathbf{1}_N + \mathbf{U}_{(r)}\hat{\boldsymbol{\omega}}_{(r)}. \tag{2.9}$$

Since

$$\begin{aligned} \mathbf{U} &= \begin{pmatrix} \mathbf{U}_{(r)} & \mathbf{U}_{(\hat{p}-r)} & \mathbf{U}_{(p-\hat{p})} \end{pmatrix} \\ &= \mathbf{Z}\mathbf{A} \\ &= \mathbf{Z}\begin{pmatrix} \mathbf{A}_{(r)} & \mathbf{A}_{(\hat{p}-r)} & \mathbf{A}_{(p-\hat{p})} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{Z}\mathbf{A}_{(r)} & \mathbf{Z}\mathbf{A}_{(\hat{p}-r)} & \mathbf{Z}\mathbf{A}_{(p-\hat{p})} \end{pmatrix}, \end{aligned}$$

we obtain $\mathbf{U}_{(r)} = \mathbf{Z}\mathbf{A}_{(r)}$. The Eq. (2.9) can be now written as

$$\tilde{\mathbf{y}} = \bar{y}\mathbf{1}_N + \mathbf{Z}\mathbf{A}_{(r)}\hat{\boldsymbol{\omega}}_{(r)}. \tag{2.10}$$

The *prediction PCR model* is given by

$$\mathbf{f}(\mathbf{z}) := \bar{y} + \mathbf{z}^T \mathbf{A}_{(r)}\hat{\boldsymbol{\omega}}_{(r)}, \tag{2.11}$$

where $\mathbf{z}^T = \begin{pmatrix} x - \bar{x}_1 & x - \bar{x}_2 & \cdots & x - \bar{x}_p \end{pmatrix}$.

[2] To detect multicollinearity (collinearity) on $\mathbf{Z}$, we use the comparison of $\frac{\lambda_l}{\lambda_1}$ for $l = 1, 2, \cdots, p$. If $\frac{\lambda_l}{\lambda_1} < \frac{1}{1000}$ then we consider that multicollinearity (collinearity) exists on $\mathbf{Z}$ [10].

## 3 Kernel Principal Component Regression

### 3.1 The new approach for KPCR

Assume we have a function $\psi : \mathbb{R}^p \rightarrow \mathbb{F}$, where $\mathbb{F}$ is the feature space which is a higher dimensional Euclidean space. By using this function, we transform $\mathbf{x}_i$ ($i = 1, 2, \cdots, N$), into the feature space $\mathbb{F}$. The image of $\mathbf{x}_i$ in $\mathbb{F}$ is given by $\psi(\mathbf{x}_i)$. As in Section 1, we define $\boldsymbol{\Psi} = \begin{pmatrix}\psi(\mathbf{x}_1) & \psi(\mathbf{x}_2) & \cdots & \psi(\mathbf{x}_N)\end{pmatrix}^T$, $\mathbf{K} = \boldsymbol{\Psi}\boldsymbol{\Psi}^T$ and $\tilde{\mathbf{C}} := \frac{1}{N}\boldsymbol{\Psi}^T\boldsymbol{\Psi}$, where size of $\boldsymbol{\Psi}$, $\tilde{\mathbf{C}}$ and $\mathbf{K}$ are $N \times p_F$, $p_F \times p_F$ and $N \times N$, respectively. We assume that $\sum_{i=1}^{N} \psi(\mathbf{x}_i) = \mathbf{0}$. The relation of eigenvalues and eigenvectors of the matrices $\tilde{\mathbf{C}}$ and $\mathbf{K}$ were firstly introduced by Scholkopf *et al.* [15]. We formalize the relation of the matrices $\tilde{\mathbf{C}}$ and $\mathbf{K}$ by the following theorem:

**Theorem 3.1.** *Suppose $\hat{\lambda} \neq 0$ and $\hat{\mathbf{a}} \in \mathbb{F} \setminus \{\mathbf{0}\}$. The following statements are equivalent:*

1. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\mathbf{a} = \tilde{\mathbf{C}}\mathbf{a}$.*
2. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\mathbf{a} = \sum_{i=1}^{N} b_i\psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix}b_1 & b_2 & \cdots & b_N\end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.*
3. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\tilde{\boldsymbol{b}} = \mathbf{K}\tilde{\boldsymbol{b}}$ and $\mathbf{a} = \sum_{i=1}^{N} \tilde{b}_i\psi(\mathbf{x}_i)$, for some $\tilde{\mathbf{b}} = \begin{pmatrix}\tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_N\end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.*

*Proof.* See Appendix I. □

The matrices $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ and $\tilde{\mathbf{C}}$ are related by the following theorem:

**Theorem 3.2.** *$\hat{\lambda}$ is an eigenvalue of $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ and $\hat{\mathbf{a}}$ is an eigenvector of $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ corresponding to $\hat{\lambda}$ if only if $\frac{1}{N}\hat{\lambda}$ is an eigenvalue of $\tilde{\mathbf{C}}$ and $\hat{\mathbf{a}}$ is an eigenvector of $\tilde{\mathbf{C}}$ corresponding to $\frac{1}{N}\hat{\lambda}$.*

*Proof.* See Appendix II. □

The *standard centered multiple linear regression model in the feature space* is given by

$$\mathbf{Y}_o = \boldsymbol{\Psi}\boldsymbol{\gamma} + \boldsymbol{\epsilon}^*, \tag{3.1}$$

where $\boldsymbol{\gamma} = \begin{pmatrix}\gamma_1 & \gamma_2 & \cdots & \gamma_{p_F}\end{pmatrix}^T$ is a vector of regression coefficients in the feature space, $\boldsymbol{\epsilon}^*$ is a vector of random error in the feature space, $\mathbf{Y}_o$ and $\mathbf{y}_o$ are defined as Section 2. Let $\hat{p}_F$ be the rank of $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ where $\hat{p}_F \leq \min(N, p_F)$.

Since the matrix $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ is a symmetric and positive semidefinite matrix, then eigenvalues of $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ are real numbers and nonnegative. Suppose that $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_{\tilde{r}} \geq \tilde{\lambda}_{\tilde{r}+1} \geq \cdots \geq \tilde{\lambda}_{\hat{p}_F} > \tilde{\lambda}_{\hat{p}_F+1} = \cdots = \tilde{\lambda}_{p_F} = 0$ are eigenvalues of $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$. Let $\tilde{\mathbf{A}} = (\tilde{\mathbf{a}}_1 \quad \tilde{\mathbf{a}}_2 \quad \cdots \quad \tilde{\mathbf{a}}_{p_F})$ be the matrix of eigenvectors $\tilde{\mathbf{a}}_l$ of $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ and $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}^{-1}$, then

$$\tilde{\mathbf{A}}^T\boldsymbol{\Psi}^T\boldsymbol{\Psi}\tilde{\mathbf{A}} = \tilde{\mathbf{D}},$$

where $\tilde{\mathbf{a}}_l$ corresponds to $\tilde{\lambda}_l$ for $l = 1, 2, \cdots, p_F$, and

$$\tilde{\mathbf{D}} = \begin{pmatrix} \tilde{\mathbf{D}}_{(\hat{p}_F)} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{pmatrix},$$

$$\tilde{\mathbf{D}}_{(\hat{p_F})} = \begin{pmatrix} \tilde{\lambda}_1 & 0 & \cdots & 0 \\ 0 & \tilde{\lambda}_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \tilde{\lambda}_{\hat{p}_F} \end{pmatrix}.$$

Since $\tilde{\mathbf{A}}$ is orthogonal matrix, we have $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T = \mathbf{I}_{p_F}$. Further, we can rewrite the model (3.1) as

$$\mathbf{Y}_o = \tilde{\mathbf{U}}\tilde{\boldsymbol{\omega}} + \boldsymbol{\epsilon}^*, \tag{3.2}$$

where $\tilde{\mathbf{U}} = \boldsymbol{\Psi}\tilde{\mathbf{A}}$ and $\tilde{\boldsymbol{\omega}} = \tilde{\mathbf{A}}^T\boldsymbol{\gamma}$. Rewriting

$$\tilde{\mathbf{U}} = \begin{pmatrix} \tilde{\mathbf{U}}_{(\hat{p}_F)} & \tilde{\mathbf{U}}_{(p_F-\hat{p}_F)} \end{pmatrix} \text{ and } \tilde{\boldsymbol{\omega}} = \begin{pmatrix} \tilde{\boldsymbol{\omega}}^T_{(\hat{p}_F)} & \tilde{\boldsymbol{\omega}}^T_{(p_F-\hat{p}_F)} \end{pmatrix}^T,$$

where sizes of $\tilde{\mathbf{U}}_{(\hat{p}_F)}$, $\tilde{\mathbf{U}}_{(p_F-\hat{p}_F)}$, $\tilde{\boldsymbol{\omega}}_{(\hat{p}_F)}$, and $\tilde{\boldsymbol{\omega}}_{(p_F-\hat{p}_F)}$ are $N \times \hat{p}_F$, $N \times (p_F - \hat{p}_F)$, $\hat{p_F} \times 1$ and $(p_F - \hat{p}_F) \times 1$, respectively. Since $\tilde{\mathbf{D}} = \tilde{\mathbf{A}}^T\boldsymbol{\Psi}^T\boldsymbol{\Psi}\tilde{\mathbf{A}} = \tilde{\mathbf{U}}^T\tilde{\mathbf{U}}$, we obtain

$$\tilde{\mathbf{U}}^T_{(\hat{p}_F)}\tilde{\mathbf{U}}_{(\hat{p}_F)} = \tilde{\mathbf{D}}_{(\hat{p}_F)},$$
$$\tilde{\mathbf{U}}^T_{(p_F-\hat{p}_F)}\tilde{\mathbf{U}}_{(p_F-\hat{p}_F)} = \mathbf{O},$$

and

$$\tilde{\mathbf{U}}^T_{(\hat{p}_F)}\tilde{\mathbf{U}}_{(p_F-\hat{p}_F)} = \mathbf{O}.$$

The model (3.2) can be written as

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\hat{p}_F)}\tilde{\boldsymbol{\omega}}_{(\hat{p}_F)} + \tilde{\mathbf{U}}_{(p_F-\hat{p}_F)}\tilde{\boldsymbol{\omega}}_{(p_F-\hat{p}_F)} + \boldsymbol{\epsilon}^*. \tag{3.3}$$

Since $\|(\tilde{\mathbf{U}}_{(p_F-\hat{p}_F)}\tilde{\boldsymbol{\omega}}_{(p_F-\hat{p}_F)})^T(\tilde{\mathbf{U}}_{(p_F-\hat{p}_F)}\tilde{\boldsymbol{\omega}}_{(p_F-\hat{p}_F)})\| = 0$, we obtain $\tilde{\mathbf{U}}_{(p_F-\hat{p}_F)}\tilde{\boldsymbol{\omega}}_{(p_F-\hat{p}_F)}$ is equal to $\mathbf{0}$. The model (3.3) reduces to

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\hat{p}_F)}\tilde{\boldsymbol{\omega}}_{(\hat{p}_F)} + \boldsymbol{\epsilon}^*. \tag{3.4}$$

Further, we assume that $\tilde{\lambda}_{\tilde{r}+1} \approx 0, \tilde{\lambda}_{\tilde{r}+2} \approx 0, \cdots, \tilde{\lambda}_{\hat{p}_F} \approx 0$ and let

$$\tilde{\mathbf{U}}_{(\hat{p}_F)} = \begin{pmatrix} \tilde{\mathbf{U}}_{(\tilde{r})} & \tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})} \end{pmatrix}, \tilde{\boldsymbol{\omega}}_{(\hat{p}_F)} = \begin{pmatrix} \tilde{\boldsymbol{\omega}}^T_{(\tilde{r})} & \tilde{\boldsymbol{\omega}}^T_{(\hat{p}_F-\tilde{r})} \end{pmatrix}^T$$

and

$$\tilde{\mathbf{D}}_{(\hat{p})} = \begin{pmatrix} \tilde{\mathbf{D}}_{(\tilde{r})} & \mathbf{O} \\ \mathbf{O} & \tilde{\mathbf{D}}_{(\hat{p}_F-\tilde{r})} \end{pmatrix},$$

where

$$\tilde{\mathbf{D}}_{(\tilde{r})} = \begin{pmatrix} \tilde{\lambda}_1 & 0 & \cdots & 0 \\ 0 & \tilde{\lambda}_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \tilde{\lambda}_{\tilde{r}} \end{pmatrix},$$

$$\tilde{\mathbf{D}}_{(\hat{p}_F-\tilde{r})} = \begin{pmatrix} \tilde{\lambda}_{\tilde{r}+1} & 0 & \cdots & 0 \\ 0 & \tilde{\lambda}_{\tilde{r}+2} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \tilde{\lambda}_{\hat{p}_F} \end{pmatrix},$$

and sizes of $\tilde{\mathbf{U}}_{(\tilde{r})}$, $\tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})}$, $\tilde{\boldsymbol{\omega}}_{(\tilde{r})}$, and $\tilde{\boldsymbol{\omega}}_{(\hat{p}_F-\tilde{r})}$ are $N \times \tilde{r}$, $N \times (\hat{p}_F - \tilde{r})$, $\tilde{r} \times 1$ and $(\hat{p}_F - \tilde{r}) \times 1$, respectively. Since $\tilde{\mathbf{D}}_{(\hat{p}_F)} = \tilde{\mathbf{U}}^T_{(\hat{p}_F)}\tilde{\mathbf{U}}_{(\hat{p}_F)}$, we obtain

$$\begin{aligned} \tilde{\mathbf{U}}^T_{(\tilde{r})}\tilde{\mathbf{U}}_{(\tilde{r})} &= \tilde{\mathbf{D}}_{(\tilde{r})}, \\ \tilde{\mathbf{U}}^T_{(\hat{p}_F-\tilde{r})}\tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})} &= \tilde{\mathbf{D}}_{(\hat{p}_F-\tilde{r})} \end{aligned}$$

and

$$\tilde{\mathbf{U}}^T_{(\tilde{r})}\tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})} = \mathbf{O}.$$

The above model (3.4) can now be written as

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\tilde{r})}\tilde{\boldsymbol{\omega}}_{(\tilde{r})} + \tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})}\tilde{\boldsymbol{\omega}}_{(\hat{p}_F-\tilde{r})} + \boldsymbol{\epsilon}^*. \tag{3.5}$$

To avoid effects of multicollinearity on $\boldsymbol{\Psi}$, we drop the term $\tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})}\tilde{\boldsymbol{\omega}}_{(\hat{p}_F-\tilde{r})}$ in the model (3.5) and obtain

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\tilde{r})}\tilde{\boldsymbol{\omega}}_{(\tilde{r})} + \boldsymbol{\epsilon}^{**}, \tag{3.6}$$

where $\boldsymbol{\epsilon}^{**}$ is a random vector influenced by dropping $\tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})}\tilde{\boldsymbol{\omega}}_{(\hat{p}_F-\tilde{r})}$ in the model (3.5). The model (3.6) shows that the effects of collinearity and multicollinearity on $\boldsymbol{\Psi}$ are reduced by transforming the orthogonal matrix $\tilde{\mathbf{A}}$.

Note that, $\tilde{\mathbf{U}}^T_{(\tilde{r})}\tilde{\mathbf{U}}_{(\tilde{r})} = \tilde{\mathbf{D}}_{(\tilde{r})}$. This implies $\text{rank}(\tilde{\mathbf{U}}^T_{(\tilde{r})}\tilde{\mathbf{U}}_{(\tilde{r})}) = \tilde{r}$. The last statement implies $\tilde{\mathbf{U}}^T_{(\tilde{r})}\tilde{\mathbf{U}}_{(\tilde{r})}$ is invertible. Hence, the estimator of $\tilde{\boldsymbol{\omega}}_{(\tilde{r})}$, say $\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})}$, is given by

$$\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})} = (\tilde{\mathbf{U}}^T_{(\tilde{r})}\tilde{\mathbf{U}}_{(\tilde{r})})^{-1}\tilde{\mathbf{U}}^T_{(\tilde{r})}\mathbf{y}_o. \tag{3.7}$$

Since $\tilde{\mathbf{U}^T}_{(\tilde{r})}\mathbf{y} = \tilde{\mathbf{U}}^T_{(\tilde{r})}\mathbf{y}_o$ (see Appendix IV), we obtain

$$\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})} = (\tilde{\mathbf{U}}^T_{(\tilde{r})}\tilde{\mathbf{U}}_{(\tilde{r})})^{-1}\tilde{\mathbf{U}}^T_{(\tilde{r})}\mathbf{y}. \tag{3.8}$$

The prediction value of $\mathbf{y}$, say $\ddot{\mathbf{y}}$, is given by

$$\ddot{\mathbf{y}} = \bar{y}\mathbf{1}_N + \tilde{\mathbf{U}}_{(\tilde{r})}\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})}. \tag{3.9}$$

Since

$$\begin{aligned} \tilde{\mathbf{U}} &= \begin{pmatrix} \tilde{\mathbf{U}}_{(\tilde{r})} & \tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})} & \tilde{\mathbf{U}}_{(p_F-\hat{p}_F)} \end{pmatrix} \\ &= \boldsymbol{\Psi}\tilde{\mathbf{A}} \\ &= \boldsymbol{\Psi}\begin{pmatrix} \tilde{\mathbf{A}}_{(\tilde{r})} & \tilde{\mathbf{A}}_{(\hat{p}_F-\tilde{r})} & \tilde{\mathbf{A}}_{(p_F-\hat{p}_F)} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})} & \boldsymbol{\Psi}\tilde{\mathbf{A}}_{(\hat{p}_F-\tilde{r})} & \boldsymbol{\Psi}\tilde{\mathbf{A}}_{(p_F-\hat{p}_F)} \end{pmatrix}, \end{aligned}$$

we obtain $\tilde{\mathbf{U}}_{(\tilde{r})} = \boldsymbol{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})}$. The Eq. (3.9) can be now written as

$$\ddot{\mathbf{y}} = \bar{y}\mathbf{1}_N + \boldsymbol{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})}\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})}. \tag{3.10}$$

The *prediction of the new KPCR* model is given by

$$\mathbf{g}(\mathbf{x}) := \bar{y} + \psi(\mathbf{x})^T\tilde{\mathbf{A}}_{(\tilde{r})}\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})}. \tag{3.11}$$

The elements of the vector $\psi(\mathbf{x})^T\tilde{\mathbf{A}}_{(\tilde{r})} = \begin{pmatrix} \psi(\mathbf{x})^T\tilde{\mathbf{a}}_1 & \psi(\mathbf{x})^T\tilde{\mathbf{a}}_2 & \cdots & \psi(\mathbf{x})^T\tilde{\mathbf{a}}_{\tilde{r}} \end{pmatrix}$ are called the *1st*, *2nd*, $\cdots$, $\tilde{r}$th *nonlinear principal component (PC) corresponding* $\psi$, respectively [14].

Until now, yet we do not know $\tilde{\mathbf{U}}_{(\tilde{r})}$ explicitly. To obtain $\tilde{\mathbf{U}}_{(\tilde{r})}$ explicitly, we consider the following theorem:

**Theorem 3.3.** *(**Mercer** [2, 9, 14]) For any symmetric, continuous and positive semidefinite kernel $\xi : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, there exist a function $\phi : \mathbb{R}^p \to \mathbb{F}$ such that*

$$\xi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y}).$$

By using the Mercer Theorem, if we choose a continuous, symmetric and positive semidefinite kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ then there exist $\phi : \mathbb{R}^p \to \mathbb{F}$ such that $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. Suppose $\psi = \phi$ and define $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. The matrix $\mathbf{K}$ can now be written as

$$\mathbf{K} = \begin{pmatrix} K_{11} & K_{12} & \cdots & K_{1N} \\ K_{21} & K_{22} & \cdots & K_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ K_{N1} & K_{N2} & \cdots & K_{NN} \end{pmatrix}.$$

We have assumed that $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_{\tilde{r}} > 0$ are eigenvalues of $\mathbf{\Psi}^T \mathbf{\Psi}$ and $\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, \cdots, \tilde{\mathbf{a}}_{\tilde{r}}$ are eigenvectors of $\mathbf{\Psi}^T \mathbf{\Psi}$ corresponding to $\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_{\tilde{r}}$, where

$$\tilde{\mathbf{a}}_i^T \tilde{\mathbf{a}}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

By using theorem 3.1 and 3.2, we can obtain

$$\begin{aligned} \tilde{\lambda}_l \boldsymbol{\tau}_l &= \mathbf{K} \boldsymbol{\tau}_l \qquad \text{for } l = 1, 2, \cdots, \tilde{r}, \\ \tilde{\mathbf{a}}_l &= \sum_{i=1}^{N} \tau_{li} \psi(\mathbf{x}_i), \end{aligned}$$

for some $\boldsymbol{\tau}_l = \begin{pmatrix} \tau_{l1} & \tau_{l2} & \cdots & \tau_{lN} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

By definition, the vector $\boldsymbol{\tau}_l$ is an eigenvector of $\mathbf{K}$. Since $\mathbf{K}$ is symmetric matrix, there exists $\hat{\boldsymbol{\tau}}_1, \hat{\boldsymbol{\tau}}_2, \cdots, \hat{\boldsymbol{\tau}}_{\tilde{r}} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$ and

$$\hat{\boldsymbol{\tau}}_i^T \hat{\boldsymbol{\tau}}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise;} \end{cases}$$

such that $\hat{\boldsymbol{\tau}}_1, \hat{\boldsymbol{\tau}}_2, \cdots, \hat{\boldsymbol{\tau}}_{\tilde{r}}$ are eigenvectors of $\mathbf{K}$ corresponding to $\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_{\tilde{r}}$. This implies that $\frac{\hat{\boldsymbol{\tau}}_1}{\sqrt{\tilde{\lambda}_1}}, \frac{\hat{\boldsymbol{\tau}}_2}{\sqrt{\tilde{\lambda}_2}}, \cdots, \frac{\hat{\boldsymbol{\tau}}_{\tilde{r}}}{\sqrt{\tilde{\lambda}_{\tilde{r}}}}$ are also eigenvectors of $\mathbf{K}$ corresponding to $\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_{\tilde{r}}$.

We denote $\hat{\boldsymbol{\tau}}_l^T = \begin{pmatrix} \hat{\tau}_{l1} & \hat{\tau}_{l2} & \cdots & \hat{\tau}_{lN} \end{pmatrix}$ for $l = 1, 2, \cdots, \tilde{r}$. Further, we can obtain

$$\begin{aligned} \tilde{\mathbf{a}}_l &= \sum_{i=1}^{N} \frac{\hat{\tau}_{li}}{\sqrt{\tilde{\lambda}_l}} \psi(\mathbf{x}_i) \qquad \text{for } l = 1, 2, \cdots, \tilde{r} \\ &= \mathbf{\Psi}^T \frac{\hat{\boldsymbol{\tau}}_l}{\sqrt{\tilde{\lambda}_l}}. \end{aligned} \tag{3.12}$$

Denoting

$$\boldsymbol{\alpha}_l := \frac{\hat{\boldsymbol{\tau}}_l}{\sqrt{\tilde{\lambda}_l}} \qquad l = 1, 2, \cdots, \tilde{r}, \tag{3.13}$$

and defining

$$\mathbf{\Gamma}_{(\tilde{r})} := \begin{pmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \cdots & \boldsymbol{\alpha}_{\tilde{r}} \end{pmatrix}, \tag{3.14}$$

imply

$$\begin{aligned}\tilde{\mathbf{A}}_{(\tilde{r})} &= (\tilde{\mathbf{a}}_1 \quad \tilde{\mathbf{a}}_2 \quad \cdots \quad \tilde{\mathbf{a}}_{\tilde{r}}) \\ &= \mathbf{\Psi}^T \mathbf{\Gamma}_{(\tilde{r})}.\end{aligned} \tag{3.15}$$

Since $\tilde{\mathbf{U}}_{(\tilde{r})} = \mathbf{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})}$ and by using Eq. (3.15), we obtain

$$\begin{aligned}\tilde{\mathbf{U}}_{(\tilde{r})} &= \mathbf{\Psi}\mathbf{\Psi}^T \mathbf{\Gamma}_{(\tilde{r})} \\ &= \mathbf{K}\mathbf{\Gamma}_{(\tilde{r})}.\end{aligned} \tag{3.16}$$

Hence,

$$\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})} = ((\mathbf{K}\mathbf{\Gamma}_{(\tilde{r})})^T (\mathbf{K}\mathbf{\Gamma}_{(\tilde{r})}))^{-1} (\mathbf{K}\mathbf{\Gamma}_{(\tilde{r})})^T \mathbf{y}, \tag{3.17}$$

and the prediction value $\ddot{\mathbf{y}}$ can now be written

$$\ddot{\mathbf{y}} = \bar{y}\mathbf{1}_N + \mathbf{K}\mathbf{\Gamma}_{(\tilde{r})}\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})}, \tag{3.18}$$

and the prediction of the new KPCR model can now be written as

$$\mathbf{g}(\mathbf{x}) = \bar{y} + \sum_{i=1}^{N} c_i \kappa(\mathbf{x}, \mathbf{x}_i), \tag{3.19}$$

where $\begin{pmatrix} c_1 & c_2 & \cdots & c_N \end{pmatrix}^T = \mathbf{\Gamma}_{(\tilde{r})}\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})}$.

### 3.2 The new KPCR's algorithm

In this Section, we summarize the procedures in Section 2 to obtain the prediction by the new KPCR.

**Algorithm**:

1. Given: $(y_i, x_{i1}, x_{i2}, \cdots, x_{ip})$, $i = 1, 2, \cdots, N$.
2. Construct:
   $\mathbf{y} = \begin{pmatrix} y_1 & y_2 & \cdots & y_2 \end{pmatrix}^T$, $\mathbf{x}_i = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{ip} \end{pmatrix}^T$ and $\bar{y} = \frac{1}{N}\mathbf{1}_N^T\mathbf{y}$.
3. Choose a kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$.
4. Construct: $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{K} = (K_{ij})$.
5. Diagonalize $\mathbf{K}$.
   Let $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_{\hat{r}} \geq \tilde{\lambda}_{\hat{r}+1} \geq \cdots \geq \tilde{\lambda}_N \geq 0$ be eigenvalues of $\mathbf{K}$ and $\hat{\boldsymbol{\tau}}_1, \hat{\boldsymbol{\tau}}_2 \cdots, \hat{\boldsymbol{\tau}}_{\hat{r}}, \hat{\boldsymbol{\tau}}_{\hat{r}+1}, \cdots, \hat{\boldsymbol{\tau}}_N$ be eigenvector of $\mathbf{K}$ corresponding to $\tilde{\lambda}_1, \tilde{\lambda}_2 \cdots, \tilde{\lambda}_{\hat{r}}, \tilde{\lambda}_{\hat{r}+1}, \cdots, \tilde{\lambda}_N$, respectively; where

$$\hat{\boldsymbol{\tau}}_i^T \hat{\boldsymbol{\tau}}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

6. Detect collinearity and multicollinearity on $\mathbf{K}$.
   Let $\tilde{r}$ be the retained number of nonlinear PC's such that $\frac{\tilde{\lambda}_{\tilde{r}+1}}{\tilde{\lambda}_1}, \frac{\tilde{\lambda}_{\tilde{r}+2}}{\tilde{\lambda}_1}, \cdots, \frac{\tilde{\lambda}_{\tilde{N}}}{\tilde{\lambda}_1} < \frac{1}{1000}$.

7. Construct:
$\boldsymbol{\alpha}_l = \frac{\hat{\boldsymbol{\tau}}_l^T}{\sqrt{\tilde{\lambda}_l}}$ for $l = 1, 2, \cdots, \tilde{r}$ and $\boldsymbol{\Gamma}_{(\tilde{r})} = \begin{pmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \cdots & \boldsymbol{\alpha}_{\tilde{r}} \end{pmatrix}$.

8. Calculate:
$\mathbf{U}_{(\tilde{r})} = \mathbf{K}\boldsymbol{\Gamma}_{(\tilde{r})}$,
$\hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})} = (\mathbf{U}_{(\tilde{r})}^T \mathbf{U}_{(\tilde{r})})^{-1} \mathbf{U}_{(\tilde{r})}^T \mathbf{y}$,
$\mathbf{c} := \begin{pmatrix} c_1 & c_2 & \cdots & c_N \end{pmatrix}^T = \boldsymbol{\Gamma}_{(\tilde{r})} \hat{\tilde{\boldsymbol{\omega}}}_{(\tilde{r})}$.

9. Given a vector $\mathbf{x} \in \mathbb{R}^p$, the prediction by the new KPCR is given by

$$g(\mathbf{x}) = \bar{y} + \textstyle\sum_{j=1}^N c_i \kappa(\mathbf{x}, \mathbf{x}_j).$$

Note that, the above algorithm works under the assumption $\sum_{i=1}^N \psi(\mathbf{x}_i) = \mathbf{0}$. When $\sum_{i=1}^N \psi(\mathbf{x}_i) \neq \mathbf{0}$, we construct $\mathbf{K}_N := \mathbf{K} - \mathbf{E}\mathbf{K} - \mathbf{K}\mathbf{E} + \mathbf{E}\mathbf{K}\mathbf{E}$ instead of $\mathbf{K}$ in Step 4, where $\mathbf{E}$ is a matrix $N \times N$ and all elements of $\mathbf{E}$ are $\frac{1}{N}$. Further, we diagonalize $\mathbf{K}_N$ in Step 5 and work based on $\mathbf{K}_N$ in the subsequent steps.

# 4 Case Studies

In these case studies, we used the Gaussian kernel $K(\mathbf{x}, \mathbf{y}) = \exp(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{\varrho})$, the Polynomial kernel $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y})^d$ and the Sigmoid kernel $K(\mathbf{x}, \mathbf{y}) = \tanh(r_2(\mathbf{x}^T\mathbf{y})^{r_1} + \theta)$ where $\varrho, d, r_1, r_2$ and $\theta$ are parameters of the kernel functions.

To test the capabilities of the linear regression, the previous KPCR and the new KPCR, we used toy data and some real data. We divided the toy data into two data sets, i.e., training data and testing data. A *training data* set is defined as given data that are going to be used to obtain the estimators of the linear regression, the previous KPCR and the new KPCR, respectively. A *testing data* set is defined as given data with noise that are going to be used to test the capabilities of the linear regression, the previous KPCR and the new KPCR, respectively. We assume that the noise is normally distributed.

As real data, we have used the stock of cars in the Netherlands (in period 1965-1989) and the weight of a certain kind of female chickens observed once a week [7] by using the Gaussian kernel ($\varrho = 0.5$). We compared the capabilities of the linear regression, the previous KPCR, a nonlinear regression based on Gompertz function and the new KPCR. The Gompertz function is given by

$$f(x, a, b, c) = \exp^{a - b\exp^{-cx}}, \qquad b, c > 0, a \in \mathbb{R}. \tag{4.1}$$

## 4.1 The toy data

For this case study, we use toy data generated by the function $y = \frac{|\sin(x)|}{|x|}$ for $x \neq 0$. We set $y = 1$ for $x = 0$. The training data are generated by $x = -10 + 0.2 \times (i - 1)$ for $i = 1, 2, \cdots, 101$. The testing data are generated by $x = -10 + 0.25 \times (i - 1)$ for $i = 1, 2, \cdots, 81$, and the standard deviation of noise of the training data is 0.05. The results of this study by using the Gaussian kernel ($\varrho = 0.5$), the Polynomial kernel ($d = 4$) and the Sigmoid kernel ($r_1 = 4, r_1 = 2, \theta = 0.1$) are shown in Figures 1 - 6, Figures 7-8 and Figures 9-12, respectively.

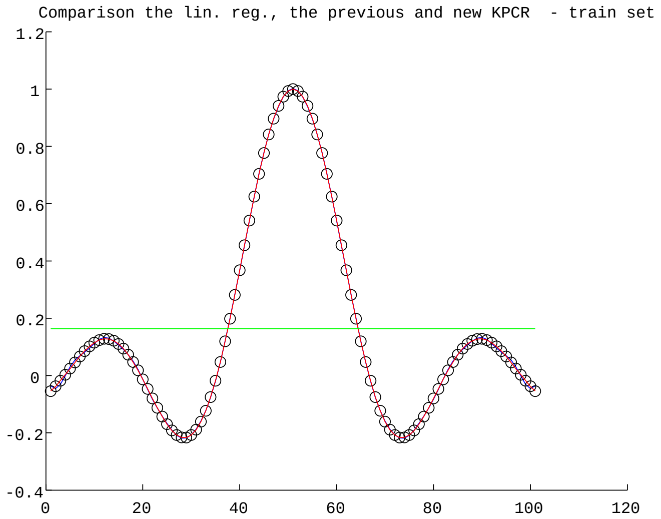

Figure 1: The linear regression, the new KPCR and the previous KPCR (Using the Gaussian kernel, $\varrho = 0.5$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given training data, the GREEN CURVE is the linear regression (RMSE=0.3516), the RED CURVE is the new KPCR (RMSE=4.0033e-004 and the number of nonlinear PC's retained is 48), and the BLUE CURVE is the previous KPCR (RMSE=0.0042 and the retained number of nonlinear PC's is 30), respectively.

According to this study, the RMSE of the new KPCR by using the Gaussian kernel is better than the others. The RMSE of the new KPCR by using the Gaussian kernel is smaller than that of the Polynomial kernel and the Sigmoid kernel.

### 4.2 The real data

As real data, we have used the stock of cars in the Netherlands (in period 1965-1989) and the weight of a certain kind of female chickens observed once a week [7] by using the Gaussian kernel ($\varrho = 0.5$). The stock of cars in the Netherlands and the weight of female chickens are given in the Table 1 and Table 2, respectively.

Jukic *et al.* [7] used the Gompertz function (4.1) to obtain the nonlinear regressions of the data. The estimators they obtained for $a, b$ and $c$ are 8.69571, 1.53597 and 0.105687, respectively, for the stock of cars in the Netherlands. The RMSE of their nonlinear regression for the stock of cars in the Netherlands is 63.2097 [7]. In our observation, the RMSE using the linear regression is 2.0587E+02. By using the retained number of PC's are 10, 20, and 24, the RMSE of the previous KPCR are 218.0894, 38.4962 and 2.1932e-012, respectively. The RMSE of the new KPCR is 2.1932e-012. The results of the case studies are shown in Figures 12-14.

For the weight of female chickens, the estimators they obtained for $a, b$ and $c$ are 1.55467, 4.13773 and 0.238587, respectively. The RMSE of their nonlinear regression for the weight of female chickens is 1.405E-02 [7]. The RMSE using the linear regression is 1.0230E-01. By using the retained number of PC's are 5, 10, 12 and 13, the RMSE of the previous KPCR are 0.2068, 0.0472, 1.3259e-015 and

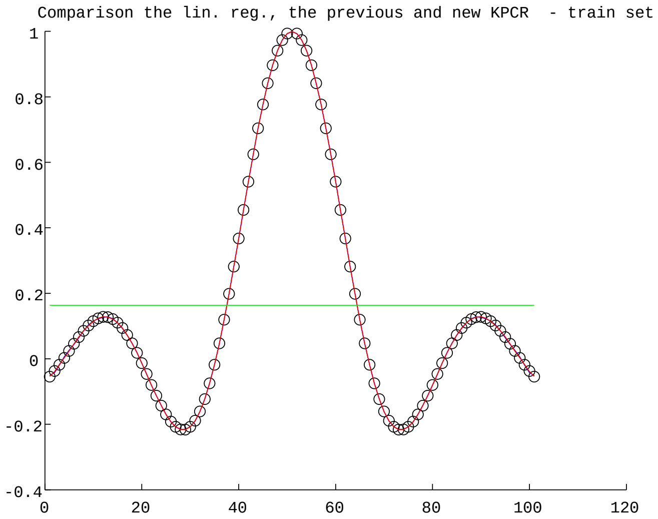

Figure 2: The linear regression, the new KPCR and the previous KPCR (Using the Gaussian kernel, $\varrho = 0.5$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given training data, the GREEN CURVE is the linear regression (RMSE=0.3516), the RED CURVE is the new KPCR (RMSE=4.0033e-004 and the number of nonlinear PC's retained is 48), and the BLUE CURVE is the previous KPCR (RMSE=0.0013 and the retained number of nonlinear PC's is 40), respectively.

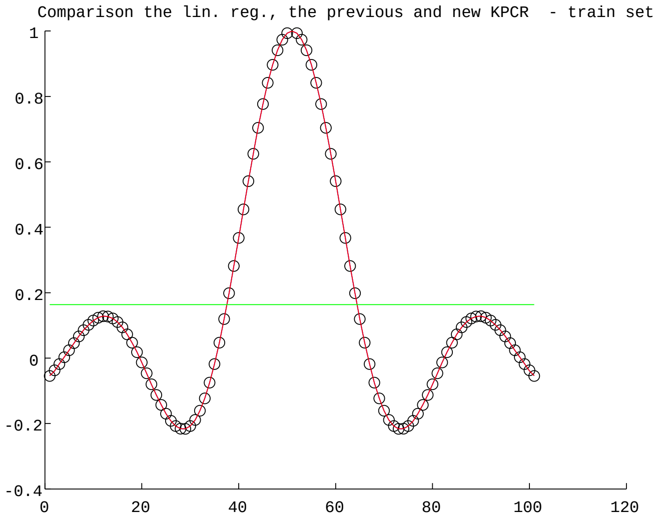

Figure 3: The linear regression, the new KPCR and the previous KPCR (Using the Gaussian kernel, $\varrho = 0.5$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given training data, the GREEN CURVE is the linear regression (RMSE=0.3516), the RED CURVE is the new KPCR (RMSE=4.0033e-004 and the number of nonlinear PC's retained is 48), and the BLUE CURVE is the previous KPCR (RMSE=4.0033e-004 and the retained number of nonlinear PC's is 48), respectively.

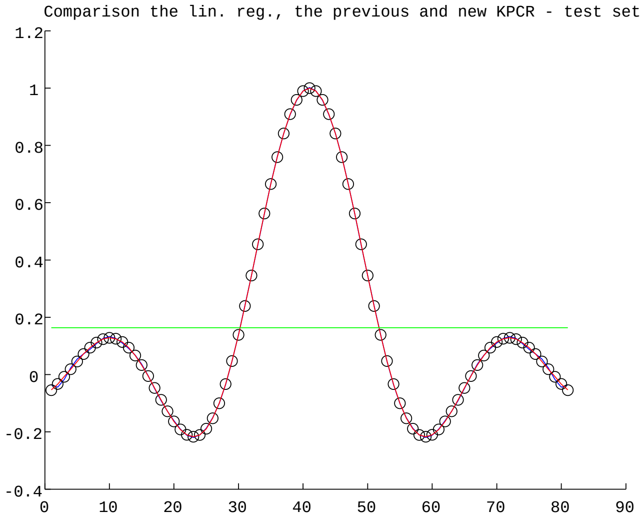

Figure 4: The linear regression, the new KPCR and the previous KPCR (Using the Gaussian kernel, $\varrho = 0.5$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given testing data, the GREEN CURVE is the linear regression (RMSE=0.3513), the RED CURVE is the new KPCR (RMSE=3.7733e-004 and the number of nonlinear PC's retained is 48), and the BLUE CURVE is the previous KPCR (RMSE=0.0044 and the retained number of nonlinear PC's is 30), respectively.

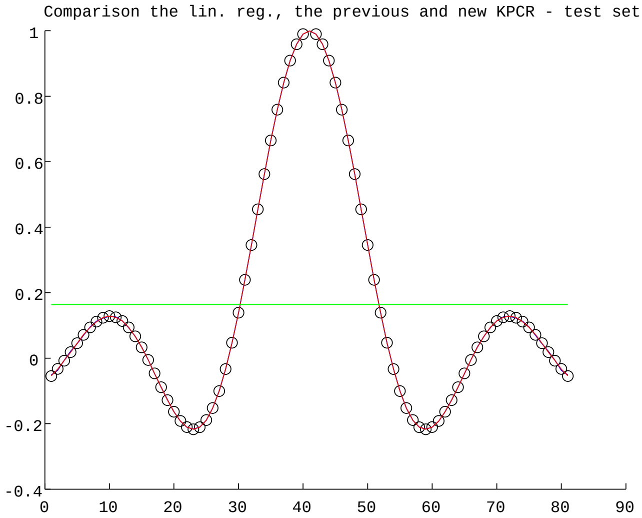

Figure 5: The linear regression, the new KPCR and the previous KPCR (Using the Gaussian kernel, $\varrho = 0.5$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given testing data, the GREEN CURVE is the linear regression (RMSE= 0.3513), the RED CURVE is the new KPCR (RMSE=3.7733e-004 and the number of nonlinear PC's retained is 48), and the BLUE CURVE is the previous KPCR (RMSE=0.0013 and the retained number of nonlinear PC's is 40), respectively.

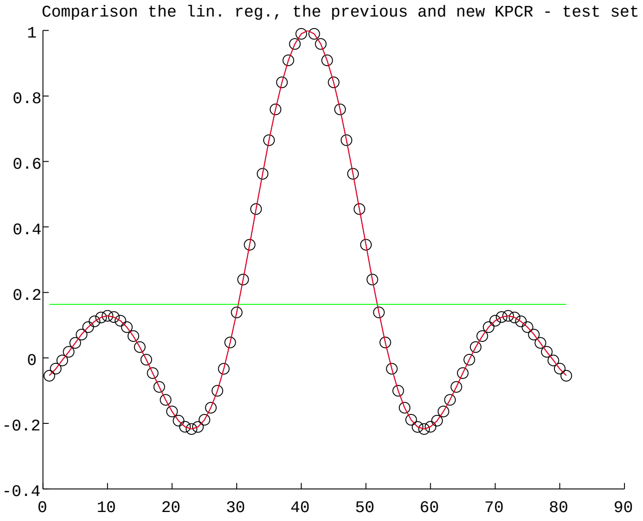

Figure 6: The linear regression, the new KPCR and the previous KPCR (Using the Gaussian kernel, $\varrho = 0.5$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given testing data, the GREEN CURVE is the linear regression (RMSE=0.3513), the RED CURVE is the new KPCR (RMSE=3.7733e-004 and the number of nonlinear PC's retained is 48), and the BLUE CURVE is the previous KPCR (RMSE=3.7733e-004 and the retained number of nonlinear PC's is 48), respectively.

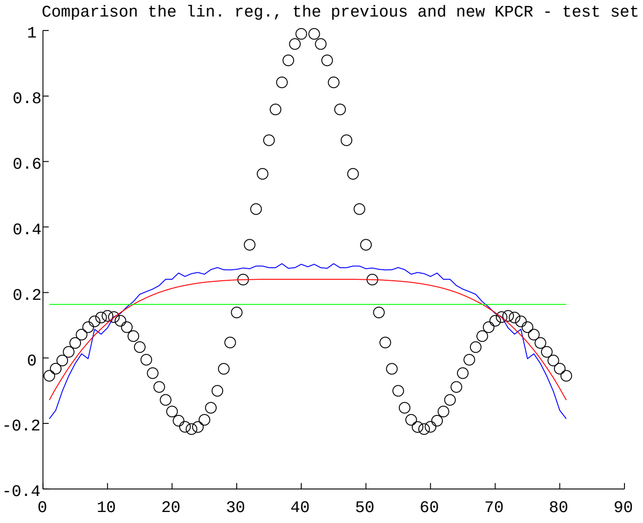

Figure 7: The linear regression, the new KPCR and the previous KPCR (Using the Polynomial kernel, $d = 4$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the testing data, the GREEN CURVE is the linear regression (RMSE=0.3516), the RED CURVE is the new KPCR (RMSE= 0.3360 and the number of nonlinear PC's retained is 1), and the BLUE CURVE is the previous KPCR (RMSE=0.3365 and the retained number of nonlinear PC's is 4), respectively.

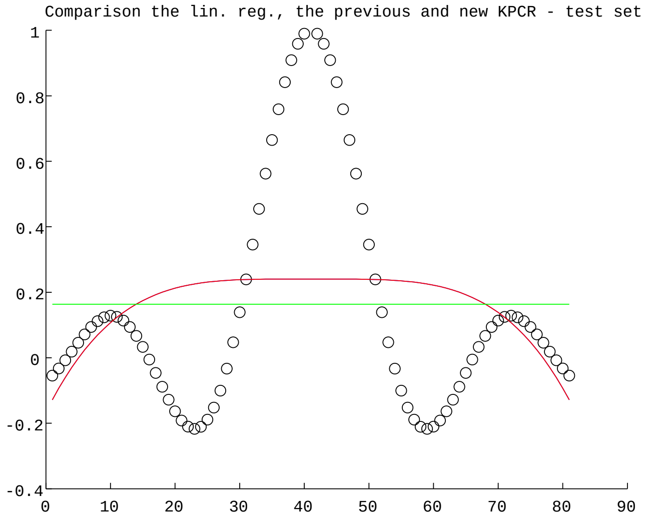

Figure 8: The linear regression, the new KPCR and the previous KPCR (Using the Polynomial kernel, $d = 4$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the testing data, the GREEN CURVE is the linear regression (RMSE=0.3516), the RED CURVE is the new KPCR (RMSE=0.3360 and the number of nonlinear PC's retained is 1), and the BLUE CURVE is the previous KPCR (RMSE=0.3360 and the retained number of nonlinear PC's is 1), respectively.

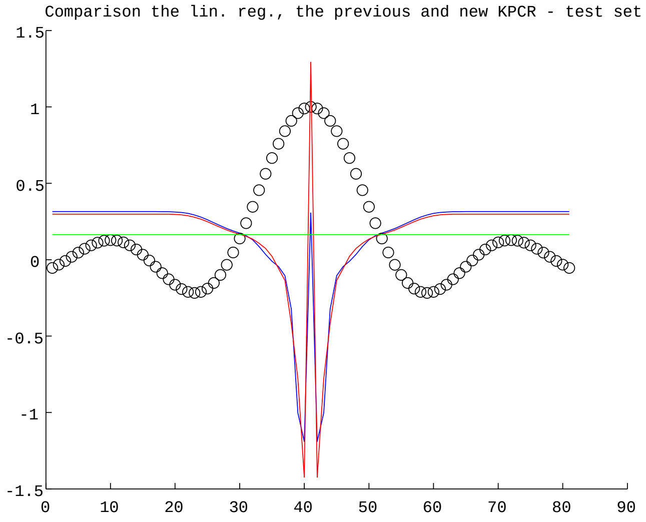

Figure 9: The linear regression, the new KPCR and the previous KPCR (Using the Sigmoid kernel, $r_1 = 4, r_1 = 2, \theta = 0.1$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given testing data, the GREEN CURVE is the linear regression (RMSE=0.3516), the RED CURVE is the new KPCR (RMSE=0.6325 and the retained number of nonlinear PC's is 9), and the BLUE CURVE is the previous KPCR (RMSE=0.6324 and the number of nonlinear PC's retained is 5), respectively.

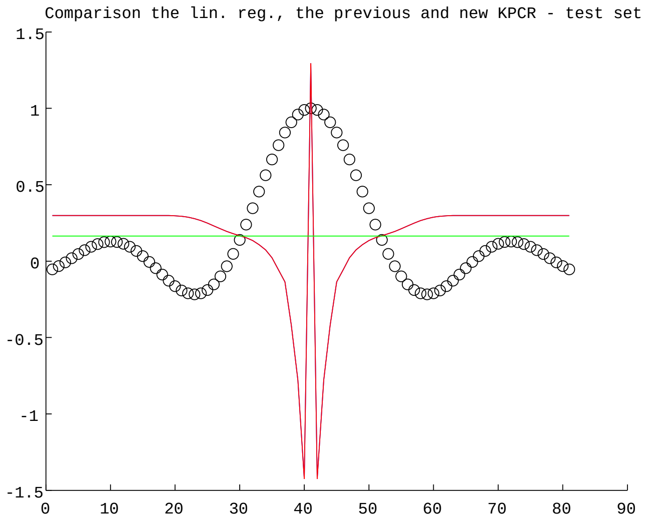

Figure 10: The linear regression, the new KPCR and the previous KPCR (Using the Sigmoid kernel, $r_1 = 4, r_1 = 2, \theta = 0.1$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given testing data, the GREEN CURVE is the linear regression (RMSE=0.3516), the RED CURVE is the new KPCR (RMSE=0.6325 and the number of nonlinear PC's retained is 9), and the BLUE CURVE is the previous KPCR (RMSE=0.6325 and the retained number of nonlinear PC's is 9), respectively.

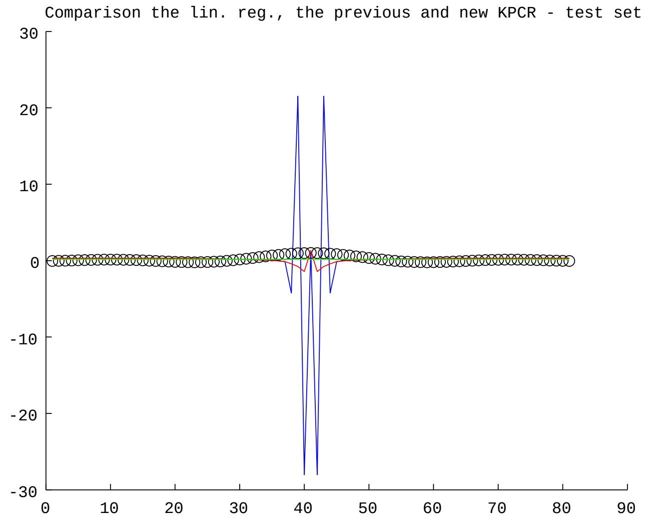

Figure 11: The linear regression, the new KPCR and the previous KPCR (Using the Sigmoid kernel, $r_1 = 4, r_1 = 2, \theta = 0.1$) for toy data generated by the function $y = \frac{|\sin(x)|}{|x|}, x \neq 0$. The BLACK DOTS are the given testing data, the GREEN CURVE is the linear regression (RMSE=0.3516), the RED CURVE is the new KPCR (RMSE=0.6325 and the number of nonlinear PC's retained is 9), and the BLUE CURVE is the previous KPCR (RMSE=5.6581 and the retained number of nonlinear PC's is 10), respectively.

1.3788e-015, respectively. The RMSE of the new KPCR is 1.3259e-015. The results of the case studies are shown in Figures 15-18.

Table 1:
The stock of cars (expressed in Thousands) in the Netherlands (in period 1965-1989).

| $x_i$(year-1965) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $y_i$ | 1273 | 1502 | 1696 | 1952 | 2212 | 2465 | 2702 | 2903 | 3080 | 3214 | 3399 | 3629 | 3851 | 4056 | 4312 | 4515 | 4594 | 4630 | 4728 | 4818 | 4901 | 4950 | 5118 | 5251 | 5371 |

Table 2:
The observing once a week the weight of a certain kind of female chickens.

| $x_i(week)$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $y_i(kg)$ | 0.147 | 0.357 | 0.641 | 0.980 | 1.358 | 1.758 | 2.159 | 2.549 | 2.915 | 3.251 | 3.510 | 3.740 | 3.925 |

We conclude this Section by two conclusions. Firstly, for the stock of cars in the Netherlands and for the weight of female chickens, the new KPCR is better than the linear regression and the nonlinear regressions based on the Gompertz function. Secondly, when the retained number of PC's for the previous KPCR and the new KPCR is the same number then they will give the same RMSE.

# 5 Conclusion

The regression analysis is one of the important techniques in multivariate data analysis. However, the linear regression model has a drawback. Existence of multicollinearity (collinearity) on $\mathbf{X}$ can seriously deteriorate the result by the linear regression model. In [3, 6, 8, 10, 16, 17], they restricted the linear regression model to the case where the column vectors of $\mathbf{X}$ are linearly independent. In this case, collinearity never exists on $\mathbf{X}$.

To avoid the effect of multicollinearity, we can use the principal component regression (PCR) [17]. However, the PCR model still has a linear fashion. Since the most real problems are nonlinear, the PCR has difficulties on its applications. To overcome such a drawback, Hoegaerts *et al.* [4], Jade *et al.* [5], and Rosipal *et al.* [11, 12, 13] used a technique, called the kernel principal component regression (KPCR). However, their KPCR [4, 5, 11, 12, 13] still have some drawbacks, i.e., the procedure to derive their KPCR and the choice rule of the retained number of PC's to avoid the effect of multicollinearity.

To overcome the above drawbacks, we propose a new approach for the KPCR. Firstly, we generalized the linear regression model in [3, 6, 8, 10, 16, 17] by relaxing the linear independence assumption. Stated in other words, our model can be used whether the column vectors of $\mathbf{X}$ are linearly independent or linearly dependent. Secondly, we showed that the PCR can also be used to reduce the effect of collinearity on $\mathbf{X}$. Finally, we propose a new approach for KPCR by using the above relaxing

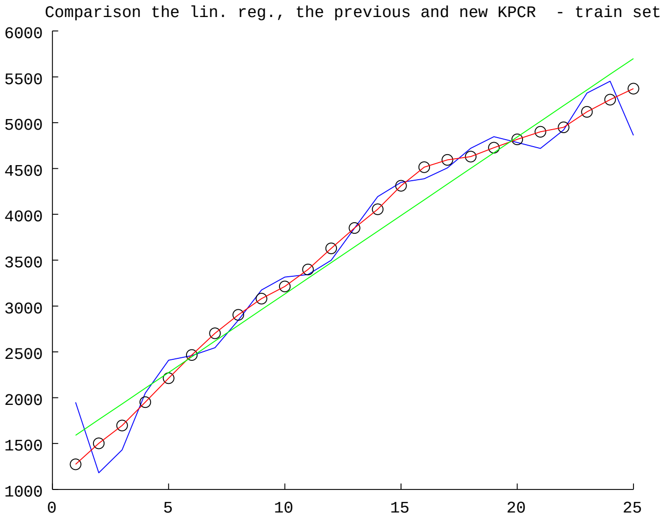

Figure 12: The linear regression, the new KPCR and the previous KPCR of the stock of cars (expressed in Thousands, using the Gaussian kernel ($\varrho = 0.5$)) in the Netherlands (in period 1965-1989). The BLACK DOTS are the given data, the GREEN CURVE is the linear regression (RMSE=205.8677), the RED CURVE is the new KPCR (RMSE=2.1932e-012 and the number of nonlinear PC's retained is 24) and BLUE CURVE is the previous KPCR (RMSE=218.0894 and the retained number of nonlinear PC's is 10), respectively.

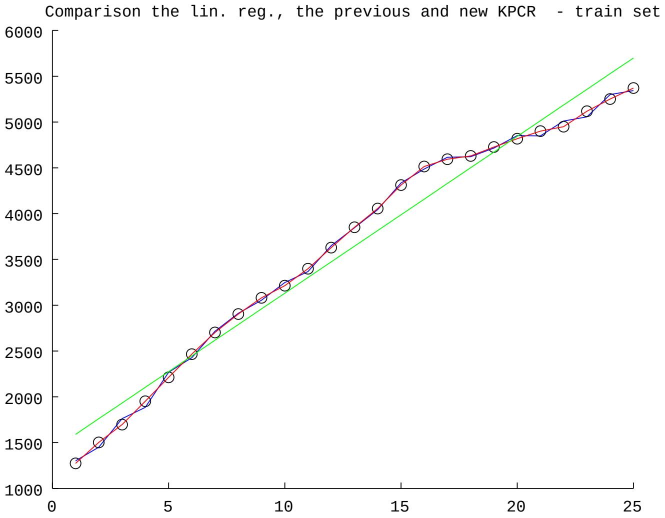

Figure 13: The linear regression, the new KPCR and the previous KPCR of the stock of cars (expressed in Thousands, using the Gaussian kernel ($\varrho = 0.5$)) in the Netherlands (in period 1965-1989). The BLACK DOTS are the given data, the GREEN CURVE is the linear regression (RMSE=205.8677), the RED CURVE is the new KPCR (RMSE=2.1932e-012 and the number of nonlinear PC's retained is 24) and BLUE CURVE is the previous KPCR (RMSE=38.4962 and the retained number of nonlinear PC's is 20), respectively.

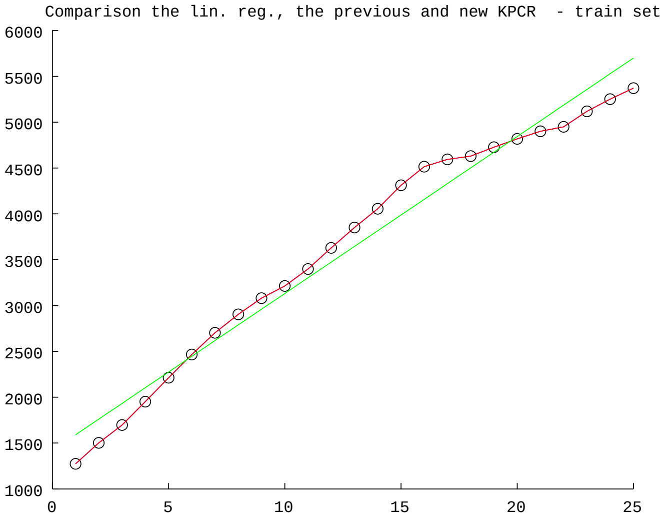

Figure 14: The linear regression, the new KPCR and the previous KPCR of the stock of cars (expressed in Thousands, using the Gaussian kernel ($\varrho = 0.5$)) in the Netherlands (in period 1965-1989). The BLACK DOTS are the given data, the GREEN CURVE is the linear regression (RMSE=205.8677), the RED CURVE is the new KPCR (RMSE=2.1932e-012 and the number of nonlinear PC's retained is 24) and BLUE CURVE is the previous KPCR (RMSE=2.1932e-012 and the retained number of nonlinear PC's is 24), respectively.

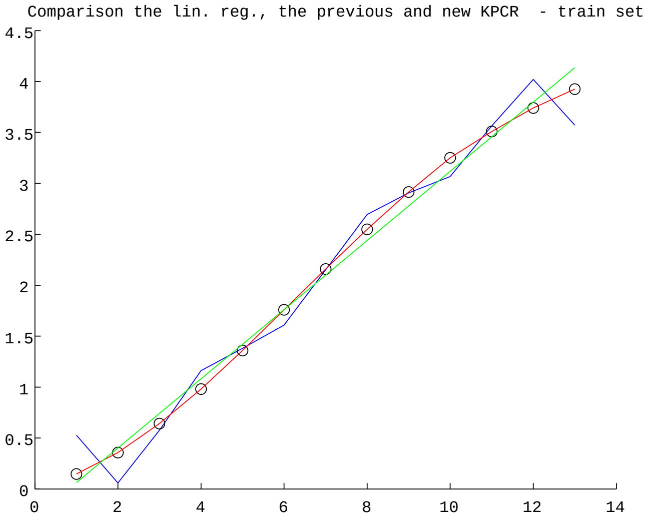

Figure 15: The linear regression, the new KPCR and the previous KPCR of the weight of female chickens (Using the Gaussian kernel, $\varrho = 0.5$). The BLACK DOTS are the given data, the GREEN CURVE is the linear regression (RMSE=0.1023), the RED CURVE is the new KPCR (RMSE=1.3259e-015 and the number of nonlinear PC's retained is 12) and BLUE CURVE is the previous KPCR (RMSE=0.2068 and the retained number of nonlinear PC's is 5), respectively.

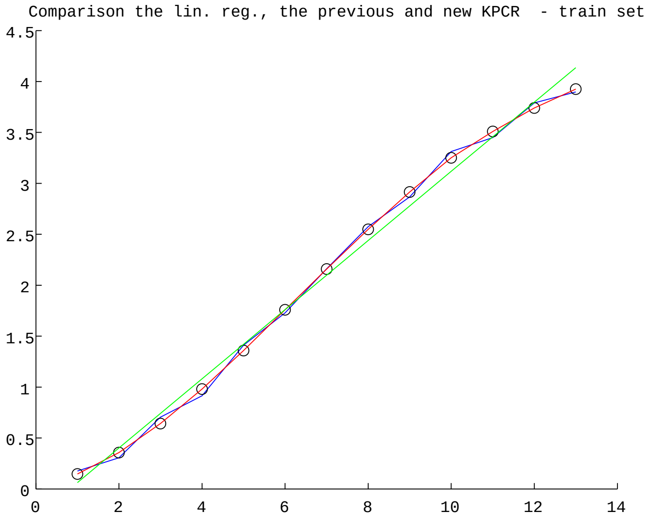

Figure 16: The linear regression, the new KPCR and the previous KPCR of the weight of female chickens (Using the Gaussian kernel, $\varrho = 0.5$). The BLACK DOTS are the given data, the GREEN CURVE is the linear regression (RMSE=0.1023), the RED CURVE is the new KPCR (RMSE=1.3259e-015 and the number of nonlinear PC's retained is 12) and BLUE CURVE is the previous KPCR (RMSE=0.0472 and the retained number of nonlinear PC's is 10), respectively.

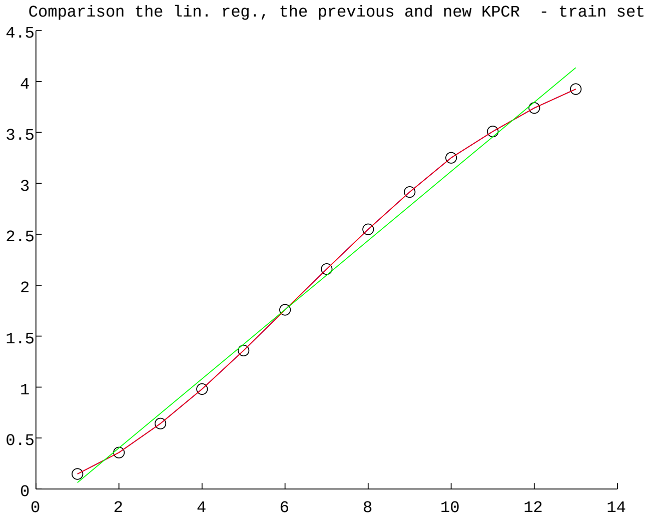

Figure 17: The linear regression, the new KPCR and the previous KPCR of the weight of female chickens (Using the Gaussian kernel, $\varrho = 0.5$). The BLACK DOTS are the given data, the GREEN CURVE is the linear regression (RMSE=0.1023), the RED CURVE is the new KPCR (RMSE=1.3259e-015 and the number of nonlinear PC's retained is 12) and BLUE CURVE is the previous KPCR (RMSE=1.3788e-015 and the retained number of nonlinear PC's is 13), respectively.

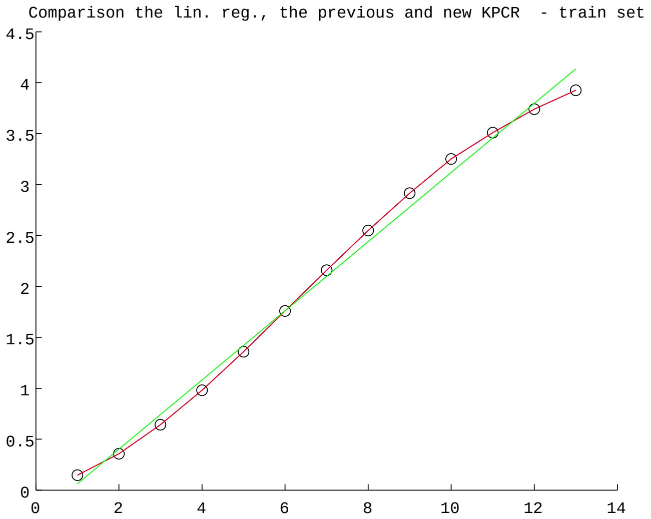

Figure 18: The linear regression, the new KPCR and the previous KPCR of the weight of female chickens (Using the Gaussian kernel, $\varrho = 0.5$). The BLACK DOTS are the given data, the GREEN CURVE is the linear regression (RMSE=0.1023), the RED CURVE is the new KPCR (The number of nonlinear PC's retained is 12, RMSE=1.3259e-015) and BLUE CURVE is the previous KPCR (RMSE=1.3259e-015 the retained number of nonlinear PC's is 12), respectively.

assumption. The procedure to derive the new approach for the KPCR is straightforward as the procedure to derive the PCR is. In the new approach for KPCR, we propose an algorithm that it can automatically obtain the retained number of nonlinear PC's to avoid the effect of multicollinearity (collinearity).

Further, we have done several case studies by using several kernel functions. Our case studies showed that the new KPCR by using the Gaussian kernel is better than the others in the RMSE sense. The RMSE of the new KPCR by using the Gaussian kernel is smallest. In our case studies, we compared the capabilities of the linear regression, the previous KPCR and the new KPCR. We note that the retained numbers of PC's for the previous KPCR are chosen by an experimenter. When the retained number of PC's for the previous KPCR and the new KPCR is the same number then they will give the same RMSE. For the real data, the stock of cars in the Netherlands (in period 1965-1989) and the weight of a certain kind of female chickens [7], the results of the new KPCR are better than the linear regression and the nonlinear regressions using the Gompertz function.

## References

[1] Howard Anton. *Elementary Linear Algebra.* John Wiley and Sons, Inc., 2000.

[2] K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks: Theory and Applications.* John Wiley and Sons, Inc., 1996.

[3] Norman R. Draper and Harry Smith. *Applied Regression Analysis.* John Wiley and Sons, 1998.

[4] L. Hoegaerts, J.A.K. Suykens, J. Vandewalle, and B. De Moor. Subset based least squares subspace in reproducing kernel hilbert space. *Neurocomputing*, pages 293–323, 2005.

[5] A.M. Jade, B. Srikanth, B.D Kulkari, J.P Jog, and L. Priya. Feature extraction and denoising using kernel pca. *Chemical Engineering Sciences*, 58:4441–4448, 2003.

[6] I.T. Jolliffe. *Principal Component Analysis.* Springer, 2002.

[7] Dragan Jukic, Gordana Kralik, and Rudolf Scitovski. Least-squares fitting gompertz curve. *Journal of Computation and Applied Mathematics*, 169:359–375, 2004.

[8] William Mendenhall, Dennis D. Wackerly, and Richard L. Sheaffer. *Mathematical Statistics with Applications.* PWS-Kent Publishing Company, 1990.

[9] Ha Quang Minh, Partha Niyogi, and Yuan Yao. Mercer's theorem, feature maps, and smoothing. *Lecture Notes in Computer Science, Springer Berling*, 4005/2006, 20009.

[10] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. *Introduction to Linear Regression.* Wiley-Interscience, 2006.

[11] Roman Rosipal, Mark Girolami, Leonard J. Trejo, and Andrzej Cichoki. Kernel pca for feature extraction and de-noising in nonlinear regression. *Neural Computing and Applications*, pages 231–243, 2001.

[12] Roman Rosipal and Leonard J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research 2*, pages 97–123, 2002.

[13] Roman Rosipal, Leonard J. Trejo, and Andrzej Cichoki. Kernel principal component regression with em approach to nonlinear principal component extraction. *Technical Report, University of Paisley, UK*, 2001.

[14] B. Scholkopf, A. Smola, and K.R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[15] Bernhard Scholkopf and Alexander J. Smola. *Learning with kernels.* The MIT Press., 2002.

[16] George A.F. Seber and Alan J. Lee. *Linear Regression Analysis.* John Wiley and Sons, Inc., 2003.

[17] M.S. Srivastava. *Methods of Multivariate Statistics.* John Wiley and Sons, Inc., 2002.

# 6 Appendix

## 6.1 Appendix I

**Lemma 6.1.** *Let* $\mathbf{x}_k \in \mathbb{R}^p, (k = 1, 2 \cdots, N)$*, be a set of a data.* $\tilde{\mathbf{X}} := \begin{pmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_N^T \end{pmatrix}^T$ *and* $\mathbf{C} := \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i\mathbf{x}_i^T$*. Suppose* $\hat{\lambda} \neq 0$ *and* $\hat{\mathbf{v}} \in \mathbb{R}^p \setminus \{\mathbf{0}\}$*. The following statements are equivalent:*
*(1)* $\hat{\lambda}$ *and* $\hat{\mathbf{v}}$ *satisfy* $\lambda\mathbf{v} = \mathbf{C}\mathbf{v}$*.*
*(2)* $\hat{\lambda}$ *and* $\hat{\mathbf{v}}$ *satisfy* $\lambda\mathbf{x}_k^T\mathbf{v} = \mathbf{x}_k^T\mathbf{C}\mathbf{v}$*, for* $k = 1, \cdots, N$*, and*
$\mathbf{v} \in$ *span* $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$*.*
*(3)* $\hat{\lambda}$ *and* $\hat{\mathbf{v}}$ *satisfy* $\lambda\mathbf{C}\mathbf{v} = \mathbf{C}^2\mathbf{v}$ *and*
$\mathbf{v} \in$ *span* $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$*.*

*Proof.* We prove (1) ⇒ (2), (2) ⇒ (3) and (3) ⇒ (1).
(1) ⇒ (2):
Suppose $\hat{\lambda}$ and $\hat{\mathbf{v}}$ satisfy $\lambda\mathbf{v} = \mathbf{C}\mathbf{v}$,
$\Rightarrow \hat{\lambda}\hat{\mathbf{v}} = \mathbf{C}\hat{\mathbf{v}}$
$\Rightarrow$ (a) $\hat{\lambda}\mathbf{x}_k^T\hat{\mathbf{v}} = \mathbf{x}_k^T\mathbf{C}\hat{\mathbf{v}}, \quad k = 1, \cdots, N.$
(b) $\hat{\mathbf{v}} = \frac{1}{\hat{\lambda}}\mathbf{C}\hat{\mathbf{v}}, \quad$ since $\hat{\lambda} \neq 0$.
$\Rightarrow \hat{\mathbf{v}} = \frac{1}{\hat{\lambda}}\frac{1}{N}\sum_{i=1}^N \mathbf{x}_i\mathbf{x}_i^T\hat{\mathbf{v}} = \sum_{i=1}^N \frac{1}{N\hat{\lambda}} < \mathbf{x}_i, \hat{\mathbf{v}} > \mathbf{x}_i.$
Letting $\alpha_i = \frac{1}{N\hat{\lambda}} < \mathbf{x}_i, \hat{\mathbf{v}} >$,
$\Rightarrow \hat{\mathbf{v}} = \sum_{i=1}^N \alpha_i\mathbf{x}_i.$
$\Rightarrow \hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{v}}$ satisfy $\lambda\mathbf{x}_k^T\mathbf{v} = \mathbf{x}_k^T\mathbf{C}\mathbf{v}$, for $k = 1, \cdots, N$, and
$\mathbf{v} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.

(2) ⇒ (3):
Suppose $\hat{\lambda}$ and $\hat{\mathbf{v}}$ satisfy $\lambda\mathbf{x}_k^T\mathbf{v} = \mathbf{x}_k^T\mathbf{C}\mathbf{v}$, for $k = 1, \cdots, N$,
and $\mathbf{v} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow$ (a) $\hat{\lambda}\mathbf{x}_k^T\hat{\mathbf{v}} = \mathbf{x}_k^T\mathbf{C}\hat{\mathbf{v}}, \quad k = 1, \cdots, N.$
(b) $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow$ (a) $\hat{\lambda}\mathbf{x}_k\mathbf{x}_k^T\hat{\mathbf{v}} = \mathbf{x}_k\mathbf{x}_k^T\mathbf{C}\hat{\mathbf{v}}, \quad k = 1, \cdots, N.$
(b) $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow$ (a) $\hat{\lambda}\sum_{k=1}^N \mathbf{x}_k\mathbf{x}_k^T\hat{\mathbf{v}} = \sum_{k=1}^N \mathbf{x}_k\mathbf{x}_k^T\mathbf{C}\hat{\mathbf{v}}.$
(b) $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow$ (a) $\hat{\lambda}(\sum_{k=1}^N \mathbf{x}_k\mathbf{x}_k^T)\hat{\mathbf{v}} = (\sum_{k=1}^N \mathbf{x}_k\mathbf{x}_k^T)\mathbf{C}\hat{\mathbf{v}}.$
(b) $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow$ (a) $\lambda N\mathbf{C}\hat{\mathbf{v}} = N\mathbf{C}\mathbf{C}\hat{\mathbf{v}}$
(b) $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow$ (a) $\hat{\lambda}\mathbf{C}\hat{\mathbf{v}} = \mathbf{C}^2\hat{\mathbf{v}}.$
(b) $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{v}}$ satisfy $\lambda\mathbf{C}\mathbf{v} = \mathbf{C}^2\mathbf{v}$ and
$\mathbf{v} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.

(3) ⇒ (1):
Suppose $\hat{\lambda}$ and $\hat{\mathbf{v}}$ satisfy $\lambda\mathbf{C}\mathbf{v} = \mathbf{C}^2\mathbf{v}$ and $\mathbf{v} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
$\Rightarrow \hat{\lambda}\mathbf{C}\hat{\mathbf{v}} = \mathbf{C}^2\hat{\mathbf{v}}$ and $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.
Since $\mathbf{C}$ is symmetric,

$\Rightarrow \exists_{\mathbf{p}_1,\mathbf{p}_2,\cdots,\mathbf{p}_p \in \{\mathbf{p}|\mathbf{p} \text{ is an eigenvector of } \mathbf{C}\}}$ $\{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_p\}$ is an orthonormal basis for $\mathbb{R}^p$.

Let $\lambda_i$ be eigenvalue of $\mathbf{C}$ belonging to $\mathbf{p}_i, (i = 1, \cdots, p)$.

$\Leftrightarrow \lambda_i \mathbf{p}_i = \mathbf{C}\mathbf{p}_i, (i = 1, \cdots, p)$

Since $\hat{\mathbf{v}} \in \mathbb{R}^p \setminus \{\mathbf{0}\}$,

$\Rightarrow \exists_{\alpha_1,\alpha_2,\cdots,\alpha_p \in \mathbb{R}}$ $\hat{\mathbf{v}} = \sum_{i=1}^{p} \alpha_i \mathbf{p}_i$.

Case 1: $\lambda_i > 0$ for $i = 1, \cdots, p$.

$\Rightarrow \hat{\lambda}\mathbf{C}\sum_{i=1}^{p} \alpha_i \mathbf{p}_i = \mathbf{C}^2 \sum_{i=1}^{p} \alpha_i \mathbf{p}_i$.

$\Rightarrow \hat{\lambda}\sum_{i=1}^{p} \alpha_i \mathbf{C}\mathbf{p}_i = \sum_{i=1}^{p} \alpha_i \mathbf{C}^2 \mathbf{p}_i$.

$\Rightarrow \hat{\lambda}\sum_{i=1}^{p} \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^{p} \alpha_i \lambda_i^2 \mathbf{p}_i$.

$\Rightarrow \sum_{i=1}^{p} (\hat{\lambda}\alpha_i\lambda_i - \alpha_i\lambda_i^2)\mathbf{p}_i = \mathbf{0}$.

Since $\{\mathbf{p}_1, \cdots, \mathbf{p}_p\}$ is linearly independent,

$\Rightarrow (\hat{\lambda}\alpha_i\lambda_i - \alpha_i\lambda_i^2) = 0$, for $i = 1, \cdots, p$.

$\Rightarrow \lambda_i(\hat{\lambda}\alpha_i - \alpha_i\lambda_i) = 0$, for $i = 1, \cdots, p$.

Since $\lambda_i > 0$ for $i = 1, \cdots, p$,

$\Rightarrow (\hat{\lambda}\alpha_i - \alpha_i\lambda_i) = 0$, for $i = 1, \cdots, p$.

$\Rightarrow \hat{\lambda}\alpha_i = \alpha_i\lambda_i$, for $i = 1, \cdots, p$.

$\Rightarrow \hat{\lambda}\alpha_i\mathbf{p}_i = \alpha_i\lambda_i\mathbf{p}_i$, for $i = 1, \cdots, p$.

$\Rightarrow \hat{\lambda}\sum_{i=1}^{p} \alpha_i \mathbf{p}_i = \sum_{i=1}^{p} \alpha_i\lambda_i\mathbf{p}_i = \sum_{i=1}^{p} \alpha_i \mathbf{C}\mathbf{p}_i$.

Since $\hat{\mathbf{v}} = \sum_{i=1}^{p} \alpha_i \mathbf{p}_i$,

$\Rightarrow \hat{\lambda}\hat{\mathbf{v}} = \mathbf{C}\hat{\mathbf{v}}$.

Case 2: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_p = 0$.

$\Rightarrow \hat{\lambda}\mathbf{C}\sum_{i=1}^{p} \alpha_i \mathbf{p}_i = \mathbf{C}^2 \sum_{i=1}^{p} \alpha_i \mathbf{p}_i$.

$\Rightarrow \hat{\lambda}\mathbf{C}(\sum_{i=1}^{r} \alpha_i \mathbf{p}_i + \sum_{i=r+1}^{p} \alpha_i \mathbf{p}_i) = \mathbf{C}^2(\sum_{i=1}^{r} \alpha_i \mathbf{p}_i + \sum_{i=r+1}^{p} \alpha_i \mathbf{p}_i)$.

Let $\mathbf{v}_1 = \sum_{i=1}^{r} \alpha_i \mathbf{p}_i$ and $\mathbf{v}_2 = \sum_{i=r+1}^{p} \alpha_i \mathbf{p}_i \Rightarrow \hat{\mathbf{v}} = \mathbf{v}_1 + \mathbf{v}_2$.

$\Rightarrow \hat{\lambda}\mathbf{C}(\mathbf{v}_1 + \mathbf{v}_2) = \mathbf{C}^2(\mathbf{v}_1 + \mathbf{v}_2)$.

Since $\mathbf{C}\mathbf{v}_2 = \mathbf{0} \Rightarrow \mathbf{C}^2\mathbf{v}_2 = \mathbf{0}$.

$\Rightarrow \hat{\lambda}\mathbf{C}\mathbf{v}_1 = \mathbf{C}^2\mathbf{v}_1$.

$\Rightarrow \hat{\lambda}\mathbf{C}\sum_{i=1}^{r} \alpha_i \mathbf{p}_i = \mathbf{C}^2 \sum_{i=1}^{r} \alpha_i \mathbf{p}_i$.

$\Rightarrow \hat{\lambda}\sum_{i=1}^{r} \alpha_i \mathbf{C}\mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \mathbf{C}^2 \mathbf{p}_i$.

$\Rightarrow \hat{\lambda}\sum_{i=1}^{r} \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \lambda_i^2 \mathbf{p}_i$.

$\Rightarrow \sum_{i=1}^{r} (\hat{\lambda}\alpha_i\lambda_i - \alpha_i\lambda_i^2)\mathbf{p}_i = \mathbf{0}$.

Since $\{\mathbf{p}_1, \cdots, \mathbf{p}_r\}$ is linearly independent,

$\Rightarrow (\hat{\lambda}\alpha_i\lambda_i - \alpha_i\lambda_i^2) = 0$, for $i = 1, \cdots, r$.

$\Rightarrow \lambda_i(\hat{\lambda}\alpha_i - \alpha_i\lambda_i) = 0$, for $i = 1, \cdots, r$.

Since $\lambda_i > 0$ for $i = 1, \cdots, r$,

$\Rightarrow (\hat{\lambda}\alpha_i - \alpha_i\lambda_i) = 0$, for $i = 1, \cdots, r$.

$\Rightarrow \hat{\lambda}\alpha_i = \alpha_i\lambda_i$, for $i = 1, \cdots, r$.

$\Rightarrow \hat{\lambda}\alpha_i\mathbf{p}_i = \alpha_i\lambda_i\mathbf{p}_i$, for $i = 1, \cdots, r$.

$\Rightarrow \hat{\lambda}\sum_{i=1}^{r} \alpha_i \mathbf{p}_i = \sum_{i=1}^{r} \alpha_i\lambda_i\mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \mathbf{C}\mathbf{p}_i$.

Since $\mathbf{v}_1 = \sum_{i=1}^{r} \alpha_i \mathbf{p}_i$,

$\Rightarrow \hat{\lambda}\mathbf{v}_1 = \mathbf{C}\mathbf{v}_1$.

By assumption $\hat{\lambda} \neq 0$,

$\Rightarrow \mathbf{v}_1 = \frac{1}{\hat{\lambda}}\mathbf{C}\mathbf{v}_1$.

$\Rightarrow \mathbf{v}_1 = \frac{1}{\hat{\lambda}}\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^T\mathbf{v}_1$.

$\Rightarrow \mathbf{v}_1 = \frac{1}{N\hat{\lambda}} \sum_{i=1}^{N} < \mathbf{x}_i, \mathbf{v}_1 > \mathbf{x}_i$.

$\Rightarrow \mathbf{v}_1 = \sum_{i=1}^{N} \frac{1}{N\hat{\lambda}} < \mathbf{x}_i, \mathbf{v}_1 > \mathbf{x}_i$.

$\Rightarrow \mathbf{v}_1 \in$ Column space of $\tilde{\mathbf{X}}^T$.

By assumption, $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$,

$\Rightarrow \hat{\mathbf{v}} \in$ Column space of $\tilde{\mathbf{X}}^T$.

$\Rightarrow \mathbf{v}_2 = \hat{\mathbf{v}} - \mathbf{v}_1 \in$ Column space of $\tilde{\mathbf{X}}^T$.

Since $\mathbf{C}\mathbf{v}_2 = \mathbf{0}$

$\Rightarrow \frac{1}{N}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{v}_2 = \mathbf{0}$. (Note: $\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^T = \frac{1}{N}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$)

$\Rightarrow \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{v}_2 = \mathbf{0}$.

$\Rightarrow \mathbf{v}_2 \in$ Nullspace of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$.

Since Nullspace of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ = Nullspace of $\tilde{\mathbf{X}}$,

$\Rightarrow \mathbf{v}_2 \in$ Nullspace of $\tilde{\mathbf{X}}$.

Since Nullspace of $\tilde{\mathbf{X}} \perp$ Column space of $\tilde{\mathbf{X}}^T$,

$\Rightarrow < \mathbf{v}_2, \mathbf{v}_2 >= 0$.

$\Rightarrow \mathbf{v}_2 = \mathbf{0}$.

Since $\hat{\mathbf{v}} = \mathbf{v}_1 + \mathbf{v}_2$,

$\Rightarrow \hat{\mathbf{v}} = \mathbf{v}_1$.

Since $\hat{\lambda}\mathbf{v}_1 = \mathbf{C}\mathbf{v}_1$.

$\Rightarrow \hat{\lambda}\hat{\mathbf{v}} = \mathbf{C}\hat{\mathbf{v}}$.

Case 3: $\lambda_i = 0$ for $i = 1, 2, \cdots, p$.

$\Rightarrow \mathbf{C}\sum_{i=1}^{p} \alpha_i\mathbf{p}_i = \sum_{i=1}^{p} \alpha_i\mathbf{C}\mathbf{p}_i = \sum_{i=1}^{p} \alpha_i\lambda_i\mathbf{p}_i = \mathbf{0}$.

$\Rightarrow \hat{\mathbf{v}} \in$ Nullspace of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$.

Since Nullspace of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$= Nullspace of $\tilde{\mathbf{X}}$.

$\Rightarrow \hat{\mathbf{v}} \in$ Nullspace of $\tilde{\mathbf{X}}$.

By assumption, $\hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$.

$\Rightarrow \hat{\mathbf{v}} \in$ Column space of $\tilde{\mathbf{X}}^T$.

Since Nullspace of $\tilde{\mathbf{X}} \perp$ Column space of $\tilde{\mathbf{X}}^T$.

$\Rightarrow < \hat{\mathbf{v}}, \hat{\mathbf{v}} >= 0$.

$\Rightarrow \hat{\mathbf{v}} = \mathbf{0}$.

(Contradiction to our assumption, i.e, $\hat{\mathbf{v}} \neq \mathbf{0}$ ). □

**Lemma 6.2.** *Let $\mathbf{x}_k \in \mathbb{R}^p, (k = 1, 2 \cdots, N)$, be a set of a data, $\psi : \mathbb{R}^p \to \mathbb{F}$ be a function from $\mathbb{R}^p$ into $\mathbb{F}$, and $\psi(\mathbf{x}_k)$ be the image of $\mathbf{x}_k$. Define $\mathbf{\Psi} := \begin{pmatrix}\psi^T(\mathbf{x}_1) & \psi^T(\mathbf{x}_2) & \cdots & \psi^T(\mathbf{x}_N)\end{pmatrix}^T$, $\tilde{\mathbf{C}} := \frac{1}{N}\mathbf{\Psi}^T\mathbf{\Psi}$ and $\mathbf{K} := \mathbf{\Psi}\mathbf{\Psi}^T$. Suppose $\hat{\lambda} \neq 0$ and $\hat{\mathbf{a}} \in \mathbb{F} \setminus \{\mathbf{0}\}$. The following statements are equivalent:*

1. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\mathbf{a} = \tilde{\mathbf{C}}\mathbf{a}$.*

2. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\mathbf{a} = \sum_{i=1}^{N} b_i\psi(\mathbf{x}_i)$,*
   *for some $\mathbf{b} = \begin{pmatrix}b_1 & b_2 & \cdots & b_N\end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.*

*Proof.* Suppose $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\mathbf{a} = \tilde{\mathbf{C}}\mathbf{a}$.

$\Leftrightarrow$ $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\psi^T(\mathbf{x}_k)\mathbf{a} = \psi^T(\mathbf{x}_k)\tilde{\mathbf{C}}\mathbf{a}$, for $k = 1, \cdots, N$,

and $\mathbf{a} \in$ span $\{\psi(\mathbf{x}_1), \psi(\mathbf{x}_2), \cdots, \psi(\mathbf{x}_N)\}$ (By Lemma (6.1)).

$\Leftrightarrow$ $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\psi^T(\mathbf{x}_k)\mathbf{a} = \psi^T(\mathbf{x}_k)\tilde{\mathbf{C}}\mathbf{a}$, for $k = 1, \cdots, N$,

and $\mathbf{a} = \sum_{i=1}^{N} b_i\psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix}b_1 & b_2 & \cdots & b_N\end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Leftrightarrow$ $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\psi^T(\mathbf{x}_k)\sum_{i=1}^{N} b_i\psi(\mathbf{x}_i) = \psi^T(\mathbf{x}_k)\tilde{\mathbf{C}}\sum_{i=1}^{N} b_i\psi(\mathbf{x}_i)$, for $k = 1, \cdots, N$, and $\mathbf{a} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Leftrightarrow$ $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda \sum_{i=1}^{N} b_i \psi^T(\mathbf{x}_k)\psi(\mathbf{x}_i) = \sum_{i=1}^{N} b_i \psi^T(\mathbf{x}_k)\tilde{\mathbf{C}}\psi(\mathbf{x}_i)$, for $k = 1, \cdots, N$,
and $\mathbf{a} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Leftrightarrow$ $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N \sum_{i=1}^{N} b_i \psi^T(\mathbf{x}_k)\psi(\mathbf{x}_i) = \sum_{i=1}^{N} b_i \psi^T(\mathbf{x}_k) \sum_{j=1}^{N} \psi(\mathbf{x}_j)\psi^T(\mathbf{x}_j)\psi(\mathbf{x}_i)$,
for $k = 1, \cdots, N$,
and $\mathbf{a} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\sum_{i=1}^{N} b_i \psi^T(\mathbf{x}_k)\psi(\mathbf{x}_i) = (\mathbf{K}\boldsymbol{b})_k$ and
$\sum_{i=1}^{N} b_i \psi^T(\mathbf{x}_k) \sum_{j=1}^{N} \psi(\mathbf{x}_j)\psi^T(\mathbf{x}_j)\psi(\mathbf{x}_i) = (\mathbf{K}^2\boldsymbol{b})_k$ for $k = 1, \cdots, N$,
where $(\mathbf{K}\boldsymbol{b})_k$ is the $k$th element of $\mathbf{K}\boldsymbol{b}$ and
$(\mathbf{K}^2\boldsymbol{b})_k$ is the $k$th element of $\mathbf{K}^2\boldsymbol{b}$, respectively.

$\Leftrightarrow$ $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N(\mathbf{K}\boldsymbol{b})_k = (\mathbf{K}^2\boldsymbol{b})_k$ for $k = 1, \cdots, N$,
and $\mathbf{a} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Leftrightarrow$ $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$
and $\mathbf{a} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

□

**Theorem 6.3.** *Let $\mathbf{x}_k \in \mathbb{R}^p, (k = 1, 2 \cdots, N)$, be a set of a data, $\psi : \mathbb{R}^p \to \mathbb{F}$ be a function from $\mathbb{R}^p$ into $\mathbb{F}$, and $\psi(\mathbf{x}_k)$ be the image of $\mathbf{x}_k$. Define $\boldsymbol{\Psi} := \begin{pmatrix} \psi^T(\mathbf{x}_1) & \psi^T(\mathbf{x}_2) & \cdots & \psi^T(\mathbf{x}_N) \end{pmatrix}^T$, $\tilde{\mathbf{C}} := \frac{1}{N}\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ and $\mathbf{K} := \boldsymbol{\Psi}\boldsymbol{\Psi}^T$. Suppose $\hat{\lambda} \neq 0$ and $\hat{\mathbf{a}} \in \mathbb{F} \setminus \{\mathbf{0}\}$. The following statements are equivalent:*

1. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\mathbf{a} = \tilde{\mathbf{C}}\mathbf{a}$.*
2. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\mathbf{a} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$,
for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.*
3. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\tilde{\boldsymbol{b}} = \mathbf{K}\tilde{\boldsymbol{b}}$ and $\mathbf{a} = \sum_{i=1}^{N} \tilde{b}_i \psi(\mathbf{x}_i)$,
for some $\tilde{\mathbf{b}} = \begin{pmatrix} \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.*

*Proof.* We prove (3) $\Leftrightarrow$ (2), since (1)$\Leftrightarrow$ (2) is proven by lemma (6.2).
(3) $\Rightarrow$ (2):
Suppose $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\tilde{\boldsymbol{b}} = \mathbf{K}\tilde{\boldsymbol{b}}$ and $\mathbf{a} = \sum_{i=1}^{N} \tilde{b}_i \psi(\mathbf{x}_i)$,
for some $\tilde{\mathbf{b}} = \begin{pmatrix} \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow$ $\hat{\lambda}N\tilde{\boldsymbol{b}} = \mathbf{K}\tilde{\boldsymbol{b}}$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} \tilde{b}_i \psi(\mathbf{x}_i)$,
for some $\tilde{\mathbf{b}} = \begin{pmatrix} \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow$ $\hat{\lambda}N\mathbf{K}\tilde{\boldsymbol{b}} = \mathbf{K}^2\tilde{\boldsymbol{b}}$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} \tilde{b}_i \psi(\mathbf{x}_i)$,
for some $\tilde{\mathbf{b}} = \begin{pmatrix} \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow$ $\exists_{\mathbf{b} = (b_1 \quad b_2 \quad \cdots \quad b_N)^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}}$ $\hat{\lambda}N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$.
$\Rightarrow$ $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\mathbf{a} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$,
for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

(2) $\Rightarrow$ (3):
Suppose $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\mathbf{a} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$,
for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow$ $\hat{\lambda}N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$,

for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \exists_{\mathbf{b}=\begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}} \hat{\lambda} N \mathbf{K} \boldsymbol{b} = \mathbf{K}^2 \boldsymbol{b}$ and $\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$.

Since $\mathbf{K}$ is symmetric,

$\Rightarrow \exists_{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N \in \{\mathbf{p} | \mathbf{p} \text{ is an eigenvector of } \mathbf{K}\}} \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\}$ is an orthonormal basis for $\mathbb{R}^N$.

Let $\lambda_i$ be eigenvalue of $\mathbf{K}$ belonging to $\mathbf{p}_i, (i = 1, \cdots, N)$.

$\Leftrightarrow \lambda_i \mathbf{p}_i = \mathbf{C} \mathbf{p}_i, (i = 1, \cdots, N)$

Since $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$,

$\Rightarrow \exists_{\alpha_1, \alpha_2, \cdots, \alpha_N \in \mathbb{R}} \mathbf{b} = \sum_{i=1}^N \alpha_i \mathbf{p}_i$.

Case 1: $\lambda_i > 0$ for $i = 1, \cdots, N$.

$\Rightarrow \hat{\lambda} N \mathbf{K} \sum_{i=1}^N \alpha_i \mathbf{p}_i = \mathbf{K}^2 \sum_{i=1}^N \alpha_i \mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^N \alpha_i \mathbf{K} \mathbf{p}_i = \sum_{i=1}^N \alpha_i \mathbf{K}^2 \mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^N \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^N \alpha_i \lambda_i^2 \mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \sum_{i=1}^N (\hat{\lambda} N \alpha_i \lambda_i - \alpha_i \lambda_i^2) \mathbf{p}_i = \mathbf{0}$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Since $\{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$ is linearly independent,

$\Rightarrow (\hat{\lambda} N \alpha_i \lambda_i - \alpha_i \lambda_i^2) = 0$, for $i = 1, \cdots, N$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \lambda_i (\hat{\lambda} N \alpha_i - \alpha_i \lambda_i) = 0$, for $i = 1, \cdots, N$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Since $\lambda_i > 0$ for $i = 1, \cdots, N$,

$\Rightarrow (\hat{\lambda} N \alpha_i - \alpha_i \lambda_i) = 0$, for $i = 1, \cdots, N$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \alpha_i = \alpha_i \lambda_i$, for $i = 1, \cdots, N$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \alpha_i \mathbf{p}_i = \alpha_i \lambda_i \mathbf{p}_i$, for $i = 1, \cdots, N$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^N \alpha_i \mathbf{p}_i = \sum_{i=1}^N \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^N \alpha_i \mathbf{K} \mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Since $\mathbf{b} = \sum_{i=1}^N \alpha_i \mathbf{p}_i$,

$\Rightarrow \hat{\lambda} N \mathbf{b} = \mathbf{K} \mathbf{b}$
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N \tilde{\boldsymbol{b}} = \mathbf{K} \tilde{\boldsymbol{b}}$ and $\mathbf{a} = \sum_{i=1}^N \tilde{b}_i \psi(\mathbf{x}_i)$,
for some $\tilde{\mathbf{b}} = \begin{pmatrix} \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Case 2: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_N = 0$.

$\Rightarrow \hat{\lambda} N \mathbf{K} \sum_{i=1}^N \alpha_i \mathbf{p}_i = \mathbf{K}^2 \sum_{i=1}^N \alpha_i \mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^N b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \mathbf{K} (\sum_{i=1}^r \alpha_i \mathbf{p}_i + \sum_{i=r+1}^N \alpha_i \mathbf{p}_i) = \mathbf{K}^2 (\sum_{i=1}^r \alpha_i \mathbf{p}_i + \sum_{i=r+1}^N \alpha_i \mathbf{p}_i)$,

$\hat{\mathbf{a}} = \sum_{i=1}^{N} b_i \psi(\mathbf{x}_i)$, for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \cdots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Let $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T = \sum_{i=1}^{r} \alpha_i \mathbf{p}_i$

and $\mathbf{v}_2 = \begin{pmatrix} v_{21} & v_{22} & \cdots & v_{2N} \end{pmatrix}^T = \sum_{i=r+1}^{N} \alpha_i \mathbf{p}_i$.

$\Rightarrow \mathbf{b} = \mathbf{v}_1 + \mathbf{v}_2 = \begin{pmatrix} v_{11} + v_{21} & v_{12} + v_{22} & \cdots & v_{1N} + v_{2N} \end{pmatrix}^T$

and $\mathbf{K}\mathbf{v}_2 = \sum_{i=r+1}^{N} \alpha_i \mathbf{K}\mathbf{p}_i = \mathbf{0}$.

$\Rightarrow \sum_{i=1}^{N} v_{2i} (\mathbf{K})_{ki} = 0$ for $k = 1, 2, \cdots, N$.

$\Rightarrow \sum_{i=1}^{N} v_{2i} \psi^T(\mathbf{x}_k) \psi(\mathbf{x}_i) = 0$ for $k = 1, 2, \cdots, N$.

$\Rightarrow \psi^T(\mathbf{x}_k) \sum_{i=1}^{N} v_{2i} \psi(\mathbf{x}_i) = 0$ for $k = 1, 2, \cdots, N$.

We claim that $\sum_{i=1}^{N} v_{2i} \psi(\mathbf{x}_i) = \mathbf{0}$ (Why?).

Suppose $\sum_{i=1}^{N} v_{2i} \psi(\mathbf{x}_i) \neq \mathbf{0}$.

$\Rightarrow (\sum_{i=1}^{N} v_{2i} \psi(\mathbf{x}_i))^T (\sum_{j=1}^{N} v_{2j} \psi(\mathbf{x}_j)) \neq 0$.

$\Rightarrow v_{21} \psi^T(\mathbf{x}_1) \sum_{j=1}^{N} v_{2j} \psi(\mathbf{x}_j) + v_{22} \psi^T(\mathbf{x}_2) \sum_{j=1}^{N} v_{2j} \psi(\mathbf{x}_j) + \cdots$
$v_{2N} \psi^T(\mathbf{x}_N) \sum_{j=1}^{N} v_{2j} \psi(\mathbf{x}_j) \neq 0$

$\Rightarrow 0 \neq 0$ (Contradiction).

$\Rightarrow \hat{\lambda} N \mathbf{K}(\mathbf{v}_1 + \mathbf{v}_2) = \mathbf{K}^2(\mathbf{v}_1 + \mathbf{v}_2)$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} (v_{1i} + v_{2i}) \psi(\mathbf{x}_i)$ for some $\mathbf{b} = \mathbf{v}_1 + \mathbf{v}_2 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\mathbf{K}\mathbf{v}_2 = \mathbf{0} \Rightarrow \mathbf{K}^2 \mathbf{v}_2 = \mathbf{0}$; and $\sum_{i=1}^{N} v_{2i} \psi(\mathbf{x}_i) = \mathbf{0}$.

$\Rightarrow \hat{\lambda} N \mathbf{K}\mathbf{v}_1 = \mathbf{K}^2 \mathbf{v}_1$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \mathbf{K} \sum_{i=1}^{r} \alpha_i \mathbf{p}_i = \mathbf{K}^2 \sum_{i=1}^{r} \alpha_i \mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^{r} \alpha_i \mathbf{K}\mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \mathbf{K}^2 \mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^{r} \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \lambda_i^2 \mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \sum_{i=1}^{r} (\hat{\lambda} N \alpha_i \lambda_i - \alpha_i \lambda_i^2) \mathbf{p}_i = \mathbf{0}$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\{\mathbf{p}_1, \cdots, \mathbf{p}_r\}$ is linearly independent.

$\Rightarrow (\hat{\lambda} N \alpha_i \lambda_i - \alpha_i \lambda_i^2) = 0$, for $i = 1, \cdots, r$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \lambda_i (\hat{\lambda} N \alpha_i - \alpha_i \lambda_i) = 0$, for $i = 1, \cdots, r$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\lambda_i > 0$ for $i = 1, \cdots, r$,

$\Rightarrow (\hat{\lambda} N \alpha_i - \alpha_i \lambda_i) = 0$, for $i = 1, \cdots, r$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \alpha_i = \alpha_i \lambda_i$, for $i = 1, \cdots, r$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \alpha_i \mathbf{p}_i = \alpha_i \lambda_i \mathbf{p}_i$, for $i = 1, \cdots, r$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^{r} \alpha_i \mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \mathbf{K}\mathbf{p}_i$,
$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i} \psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\mathbf{v}_1 = \sum_{i=1}^{r} \alpha_i \mathbf{p}_i$.

$\Rightarrow \hat{\lambda} N \mathbf{v}_1 = \mathbf{K}\mathbf{v}_1$,

$\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N \mathbf{v}_1 = \mathbf{K}\mathbf{v}_1$,

$\mathbf{a} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N \tilde{\mathbf{b}} = \mathbf{K}\tilde{\mathbf{b}}$,

$\mathbf{a} = \sum_{i=1}^{N} \tilde{b}_i\psi(\mathbf{x}_i)$ for some $\tilde{\mathbf{b}} = \begin{pmatrix} \tilde{b}_1 & \tilde{b}_{12} & \cdots & \tilde{b}_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Case 3: $\lambda_1 = \lambda_2 = \cdots = \lambda_r = \lambda_{r+1} = \cdots = \lambda_N = 0$.

$\Rightarrow \mathbf{K}\mathbf{b} = \sum_{i=1}^{N} \alpha_i \mathbf{K}\mathbf{p}_i = \mathbf{0}$.

$\Rightarrow \sum_{i=1}^{N} b_i(\mathbf{K})_{ki} = 0$ for $k = 1, 2, \cdots, N$.

$\Rightarrow \sum_{i=1}^{N} b_i\psi^T(\mathbf{x}_k)\psi(\mathbf{x}_i) = 0$ for $k = 1, 2, \cdots, N$.

$\Rightarrow \psi^T(\mathbf{x}_k)\sum_{i=1}^{N} b_i\psi(\mathbf{x}_i) = 0$ for $k = 1, 2, \cdots, N$.

We claim that $\sum_{i=1}^{N} b_i\psi(\mathbf{x}_i) = \mathbf{0}$ (Why?).

Suppose $\sum_{i=1}^{N} b_i\psi(\mathbf{x}_i) \neq \mathbf{0}$.

$\Rightarrow (\sum_{i=1}^{N} b_i\psi(\mathbf{x}_i))^T(\sum_{j=1}^{N} b_j\psi(\mathbf{x}_j)) \neq 0$.

$\Rightarrow b_1\psi^T(\mathbf{x}_1)\sum_{j=1}^{N} b_j\psi(\mathbf{x}_j) + b_2\psi^T(\mathbf{x}_2)\sum_{j=1}^{N} b_j\psi(\mathbf{x}_j) + \cdots$

$b_N\psi^T(\mathbf{x}_N)\sum_{j=1}^{N} b_j\psi(\mathbf{x}_j) \neq 0$

$\Rightarrow 0 \neq 0$.

(Contradiction)

□

## 6.2 Appendix II

**Theorem 6.4.** *$\hat{\lambda}$ is an eigenvalue of $\mathbf{\Psi}^T\mathbf{\Psi}$ and $\hat{\mathbf{a}}$ is an eigenvector of $\mathbf{\Psi}^T\mathbf{\Psi}$ corresponding to $\hat{\lambda}$ if only if $\frac{1}{N}\hat{\lambda}$ is an eigenvalue of $\tilde{\mathbf{C}} = \frac{1}{N}\mathbf{\Psi}^T\mathbf{\Psi}$ and $\hat{\mathbf{a}}$ is an eigenvector of $\tilde{\mathbf{C}}$ corresponding to $\frac{1}{N}\hat{\lambda}$.*

*Proof.* $\hat{\lambda}$ is an eigenvalue of $\mathbf{\Psi}^T\mathbf{\Psi}$ and $\hat{\mathbf{a}}$ is an eigenvector of $\mathbf{\Psi}^T\mathbf{\Psi}$ corresponding to $\hat{\lambda}$.

$\Leftrightarrow \mathbf{\Psi}^T\mathbf{\Psi}\hat{\mathbf{a}} = \hat{\lambda}\hat{\mathbf{a}}$.

$\Leftrightarrow \frac{1}{N}\mathbf{\Psi}^T\mathbf{\Psi}\hat{\mathbf{a}} = \frac{1}{N}\hat{\lambda}\hat{\mathbf{a}}$.

$\Leftrightarrow \tilde{\mathbf{C}}\hat{\mathbf{a}} = \frac{1}{N}\hat{\lambda}\hat{\mathbf{a}}$.

$\Leftrightarrow \frac{1}{N}\hat{\lambda}$ is an eigenvalue of $\tilde{\mathbf{C}}$ and $\hat{\mathbf{a}}$ is an eigenvector of $\tilde{\mathbf{C}}$ corresponding to $\frac{1}{N}\hat{\lambda}$ □

## 6.3 Appendix III

We prove that $\mathbf{U}_{(r)}^T\mathbf{y} = \mathbf{U}_{(r)}^T\mathbf{y}_o$.

*Proof.* In Section 2, we have defined $\mathbf{Z} = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\tilde{\mathbf{X}}$ and $\mathbf{y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\mathbf{y}$. Let $\mathbf{B} = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)$. The matrix $\mathbf{B}$ is a symmetric and idempotent matrix, since

$\mathbf{B} = \mathbf{B}^T$ and $\mathbf{BB} = \mathbf{B}$. Hence, we have $\mathbf{Z} = \mathbf{B}\tilde{\mathbf{X}}$ and $\mathbf{y}_o = \mathbf{By}$. This implies

$$\begin{aligned}
\mathbf{Z}^T\mathbf{y}_o &= \mathbf{Z}^T\mathbf{By} \\
&= \tilde{\mathbf{X}}^T\mathbf{B}^T\mathbf{By} \\
&= \tilde{\mathbf{X}}^T\mathbf{BBy} \qquad \text{(symmetric).} \\
&= \tilde{\mathbf{X}}^T\mathbf{By} \qquad \text{(idempotent).} \\
&= \tilde{\mathbf{X}}^T\mathbf{B}^T\mathbf{y} \qquad \text{(symmetric).} \\
&= (\mathbf{B}\tilde{\mathbf{X}})^T\mathbf{y} \\
&= \mathbf{Z}^T\mathbf{y}
\end{aligned}$$

Since

$$\begin{aligned}
\mathbf{U} &= \begin{pmatrix}\mathbf{U}_{(r)} & \mathbf{U}_{(\hat{p}-r)} & \mathbf{U}_{(p-\hat{p})}\end{pmatrix} \\
&= \mathbf{ZA} \\
&= \mathbf{Z}\begin{pmatrix}\mathbf{A}_{(r)} & \mathbf{A}_{(\hat{p}-r)} & \mathbf{A}_{(p-\hat{p})}\end{pmatrix} \\
&= \begin{pmatrix}\mathbf{ZA}_{(r)} & \mathbf{ZA}_{(\hat{p}-r)} & \mathbf{ZA}_{(p-\hat{p})}\end{pmatrix},
\end{aligned}$$

we obtain $\mathbf{U}_{(r)} = \mathbf{ZA}_{(r)}$. This implies,

$$\begin{aligned}
\mathbf{U}_{(r)}^T\mathbf{y}_o &= (\mathbf{ZA}_{(r)})^T\mathbf{y}_o. \\
&= \mathbf{A}_{(r)}^T\mathbf{Z}^T\mathbf{y}_o. \\
&= \mathbf{A}_{(r)}^T\mathbf{Z}^T\mathbf{y}. \\
&= (\mathbf{ZA}_{(r)})^T\mathbf{y}. \\
&= \mathbf{U}_{(r)}^T\mathbf{y}.
\end{aligned}$$

□

### 6.4 Appendix IV

We prove that $\widetilde{\mathbf{U}^T}_{(\tilde{r})}\mathbf{y} = \tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{y}_o$.

*Proof.* From Section 3, since $\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{\Psi} = \mathbf{O}$ we have $\mathbf{\Psi} = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\mathbf{\Psi}$ and $\mathbf{y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\mathbf{y}$. Let $\mathbf{B} = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)$. The matrix $\mathbf{B}$ is a symmetric and idempotent matrix, since $\mathbf{B} = \mathbf{B}^T$ and $\mathbf{BB} = \mathbf{B}$. Hence, we have $\mathbf{\Psi} = \mathbf{B\Psi}$ and $\mathbf{y}_o = \mathbf{By}$. This implies

$$\begin{aligned}
\mathbf{\Psi}^T\mathbf{y}_o &= \mathbf{\Psi}^T\mathbf{By} \\
&= \mathbf{\Psi}^T\mathbf{B}^T\mathbf{By} \\
&= \mathbf{\Psi}^T\mathbf{BBy} \qquad \text{(symmetric).} \\
&= \mathbf{\Psi}^T\mathbf{By} \qquad \text{(idempotent).} \\
&= \mathbf{\Psi}^T\mathbf{B}^T\mathbf{y} \qquad \text{(symmetric).} \\
&= (\mathbf{B\Psi})^T\mathbf{y} \\
&= \mathbf{\Psi}^T\mathbf{y}
\end{aligned}$$

Since

$$\begin{aligned}\tilde{\mathbf{U}} &= \left(\tilde{\mathbf{U}}_{(\tilde{r})} \quad \tilde{\mathbf{U}}_{(\hat{p_F}-\tilde{r})} \quad \tilde{\mathbf{U}}_{(p_F-\hat{p_F})}\right)\\ &= \mathbf{\Psi}\tilde{\mathbf{A}}\\ &= \mathbf{\Psi}\left(\tilde{\mathbf{A}}_{(\tilde{r})} \quad \tilde{\mathbf{A}}_{(\hat{p_F}-\tilde{r})} \quad \tilde{\mathbf{A}}_{(p_F-\hat{p_F})}\right)\\ &= \left(\mathbf{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})} \quad \mathbf{\Psi}\tilde{\mathbf{A}}_{(\hat{p_F}-\tilde{r})} \quad \mathbf{\Psi}\tilde{\mathbf{A}}_{(p_F-\hat{p_F})}\right),\end{aligned}$$

we obtain $\tilde{\mathbf{U}}_{(\tilde{r})} = \mathbf{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})}$. This implies

$$\begin{aligned}\tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{y}_o &= (\mathbf{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})})^T\mathbf{y}_o.\\ &= \tilde{\mathbf{A}}_{(\tilde{r})}^T\mathbf{\Psi}^T\mathbf{y}_o.\\ &= \tilde{\mathbf{A}}_{(\tilde{r})}^T\mathbf{\Psi}^T\mathbf{y}.\\ &= (\mathbf{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})})^T\mathbf{y}.\\ &= \tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{y}.\end{aligned}$$

□

## 6.5 Appendix V

```
function new_kpcr_main
% Antoni Wibowo, last modified 08/01/2008.
% Graduate School of System and Information Engineering,
% University of Tsukuba, Japan.
%
% INPUTS:
%    X: the training data for the regressor variable.
%    Y: the training data for the response variable.
%    Xt:the testing for the regressor variable.
%    Yt:the testing for the response variable.
%
% OUTPUTS:
%    n: the number of observation/data.
%    num_of_PC: the nonlinear PCs retained in the previous KPCR (given).
%    n_pc_new: the nonlinear PCs retained in the new KPCR.
%    Yp_hat: the prediction of the previous KPCR for training data.
%    Ypt_hat: the prediction of the previous KPCR for testing data.
%    Y_hat: the prediction of the new KPCR for training data.
%    Yt_hat: the prediction of the new KPCR for testing data.
%    Y_hat_lin: the prediction of the lin. reg. for training data.
%    Yt_hat_lin: the prediction of the lin. reg. for testing data.
%    RMSEp: RMSE by the previous KPCR for training data.
%    RMSEpt: RMSE by the previous KPCR for testing data.
%    RMSE: RMSE by the new KPCR for training data.
%    RMSEt: RMSE by the new KPCR for testing data.
%    RMSE_lin: RMSE by the lin. reg. for training data.
%    RMSEt_lin: RMSE by the lin. reg. for testing data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%clear command window
clc;
%clear all figures
close all;
%removing all variables from memory.
clear all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                           Setting parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Set need_previousKPCR=1 if need to do using the previous KPCR.
need_previousKPCR=1;

%select kernel function
%     type:  ’G’ Gaussian   Kernel  exp((|x-y|^2)/par1)
%            'P' Polynomial Kernel  (<x,y>)^par2:
%            'S' Sigmoid    Kernel  tanh(par3*<x,y>^par4+par5)
%            'L' linear     kernel  <x,y>
type='G';

%gaussian parameter.
par1 =0.5;

%polynomial parameter.
par2=4;

%sigmoid parameter.
par3=2;
par4=2;
par5=0.1;

%display kernel's parameters
if type=='G'
    disp('parameter gaussian');disp(par1);
elseif type=='P'
    disp('parameter polynomial');disp(par2);
elseif type=='S'
    disp('parameter sigmoid (par3):');disp(par3)
    disp('parameter sigmoid (par4):');disp(par4);
    disp('parameter sigmoid (par5):');disp(par5);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                         end of setting parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                         choose data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n_choose=4;
[X,Y,Xt,Yt,cY,cYt]=choosedata(n_choose);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                         end choose data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% number of PCs retained in previous KPCR
num_of_PC=12;

[n,dimX]=size(X);
[nt,dimX]=size(Xt);
disp('number of observation/data :');disp(n)

if num_of_PC>n
```

```
    disp('*****************************************');
    disp('retained PCs must less than');disp(n);
    disp('*****************************************');
    return
end

Y_temp=Y;
Yt_temp=Yt;

if need_previousKPCR==1
    %%% KPCA  for the previous KPCR;
    %%% carry out to KPCA;
    [P_X,P_Xt,W,D]=newKPCA(X,Xt,num_of_PC,type,par1,par2,par3,par4,par5);
    size(P_X);

    %%% 2)  The previous KPCR - centralized model %%%%
    Y=Y_temp;
    Yt=Yt_temp;
    mn=mean(Y);
    Y=Y-mn;
    Yt=Yt-mn;
    [Yp_hat,Ypt_hat,B]=KPCR_cent(P_X,P_Xt,D,Y,Yt);
    Yp_hat=Yp_hat+mn;
    Ypt_hat=Ypt_hat+mn;

    %%% Plotting
    % training
    figure(1)
    hold on
    plot(Yp_hat,'b');
    RMSEp=sqrt(mean((cY-Yp_hat).^2));
    % testing
    if size(Xt,1)~=1
        figure(2)
        hold on
        plot(Ypt_hat,'b');
        RMSEpt=sqrt(mean((cYt-Ypt_hat).^2));
    end

    disp('*****************************************');
    disp('retained PCs in the previous KPCR (given) :');disp(num_of_PC);
    disp('RMSE train the previous KPCR :');disp(RMSEp);
    if size(Xt,1)~=1
        disp('RMSE test  the previous KPCR :');disp(RMSEpt);
    end
    disp('*****************************************');
end
```

```
%%% KPCA  for the new KPCR
Y=Y_temp;
Yt=Yt_temp;
[U_X,U_Xt,W_new,D,n_pc_new]=newKPCA2(X,Xt,type,par1,par2,par3,par4,par5);
%%% The new KPCR
mn=mean(Y);
Y=Y-mn;
Yt=Yt-mn;
[Y_hat_new,Yt_hat_new,B]=KPCR_cent_new(U_X,U_Xt,D,Y,Yt);
Y_hat_new=Y_hat_new+mn;
Yt_hat_new=Yt_hat_new+mn;

%%% Plotting
% training
figure(1)
plot(cY,'ko');
hold on
plot(Y_hat_new,'r');
title('Comparison the lin. reg., the previous and new KPCR  - train set ')
RMSE=sqrt(mean((cY-Y_hat_new).^2));
% testing
if size(Xt,1)~=1
    figure(2)
    plot(cYt,'ko');
    hold on
    plot(Yt_hat_new,'r');
    title('Comparison the lin. reg., the previous and new KPCR - test set ')
    RMSEt=sqrt(mean((cYt-Yt_hat_new).^2));
end

%%% linear regression
Y=Y_temp;
Yt=Yt_temp;
[Y_hat_lin, Yt_hat_lin]=linregression(X,Xt,Y);
%%% Plotting
% training
figure(1)
hold on
plot(Y_hat_lin,'g');
RMSE_lin=sqrt(mean((cY-Y_hat_lin).^2));
% testing
if size(Xt,1)~=1
    figure(2)
    hold on
    plot(Yt_hat_lin,'g');
    RMSEt_lin=sqrt(mean((cYt-Yt_hat_lin).^2));
end
```

```
disp('*****************************************');
disp('retained PCs in the new KPCR :');disp(n_pc_new);
disp('RMSE train the new KPCR :');disp(RMSE);
disp('RMSE train the lin. reg. :');disp(RMSE_lin);
if size(Xt,1)~=1
    disp('RMSE test the new  KPCR :');disp(RMSEt);
    disp('RMSE test the lin. reg. :');disp(RMSEt_lin);
end
disp('*****************************************');


function [P_X,P_Xt,W_new,D,n_pc_new]=newKPCA2(X,Xt,type,par1,par2,par3,par4,par5)

    %%% Kernel Principal Component Analysis
    %%%
    %%% Inputs
    %     X  : training data points  (number of samples  x dimension)
    %     Xt : testing  data points  (number of samples  x dimension)
    %
    %     type:  'G' Gaussian   Kernel  exp((|x-y|^2)/par1)
    %            'P' Polynomial Kernel  (<x,y>)^par2:
    %            'S' Sigmoid    Kernel  tanh(par3*<x,y>^par4+par5)
    %            'L' linear     kernel  <x,y>
    %
    %%  Output
    %     P_X  : projection of training data  onto the first n_pc_new  PC's
    %     P_Xt : projection of testing  data  onto the first n_pc_new  PC's
    %
    %    n_pc_new : the retained PCs for the new KPCR.
    %    W_new,D  : eigenvectors and eigenvalues of the centralized (cen_K)
    %               training data kernel matrix
    %

    [n,dim]=size(X);
    [nt,dim]=size(Xt);

    %%% training data kernel matrix construction
    K=newKernel(X,type,par1,par2,par3,par4,par5);

    %%% centering of K
    M=eye(n)-ones(n,n)/n;
    cen_K=M*K*M;

    %%% KPCA
    [u,D,W] = svd(cen_K);
    D = diag(D);
    clear u

 %  Detect collinearity and multicollinearity.
```

```
    for i=1:size(D,1),
        test_multi=D(i)/D(1);
        if test_multi >= 1/1000
            W_new(:,i) = W(:,i);
        end
      % if D(i)~=0
      %     W_p(:,i)=W(:,i);
      % end
    end
    size(W_new);
    n_pc_new=size(W_new,2);
    %%% training data projection
    for k=1:n_pc_new
       P_X(:,k)=W_new(:,k)*sqrt(D(k));
    end

    %%% TESTING PART

    % testing data kernel matrix construction
     Kt=newKernel_Test(X,Xt,type,par1,par2,par3,par4,par5);

    %size(Kt);
    %%% centering of Kt
    Mt=ones(nt,n)/n;
    cen_Kt = (Kt - Mt*K)*M;

    for k=1:n_pc_new
       Q(:,k)=W_new(:,k)/(sqrt(D(k)));
    end
    %size(Q);

    %%% testing  data projection
    P_Xt=cen_Kt*Q;
    %size(P_Xt);


function  [Y_hat_new,Yt_hat_new,B]=KPCR_cent_new(U_X,U_Xt,D,Y,Yt)

  %%% Kernel Principal Component Regression - centralized regression model
  %%%
  %%% Inputs
    %     U_X  : projected training data points onto the first n_p_new PC's
    %     U_Xt : projected testing  data points onto the first n_p_new PC's
    %     D    : (at least) first n_p_new ordered (maximal first) eigenvalues
    %             of the centralized training data kernel matrix
    %     Y    : zero mean training outputs (number of samples  x dim)
    %     Yt   : zero mean testing  outputs (number of samples  x dim)
    %
    %     Outputs:
```

```
    %
    %     Y_hat_new  : predicted training outputs (number of samples  x dim)
    %     Yt_hat_new : predicted testing  outputs (number of samples  x dim)
    %     B : matrix (or vector) of regression coefficients (n_p_new x dim)

   [n,p]=size(U_X);
   D=D(1:p)’; %%% only the first n_p_new-eigenvalues are used

   B=diag(1./D)*U_X’*Y;
   Y_hat_new=U_X*B;
   Yt_hat_new=U_Xt*B;
```

```
function [P_X,P_Xt,W,D]=newKPCA(X,Xt,n_pc,type,par1,par2,par3,par4,par5)

    %%% Kernel Principal Component Analysis
    %%%
    %%% Inputs
    %     X  : training data points  (number of samples  x dimension)
    %     Xt : testing  data points  (number of samples  x dimension)
    %
    %     n_pc : number of principal componets onto which data are projected
    %
    %     type:  ’G’ Gaussian   Kernel  exp((|x-y|^2)/par1)
    %            ’P’ Polynomial Kernel  (<x,y>)^par2:
    %            ’S’ Sigmoid    Kernel  tanh(par3*<x,y>^par4+par5)
    %            ’L’ linear     kernel  <x,y>
    %
    %%  Output
    %     P_X  : projection of training data  onto the first n_pc PC’s
    %     P_Xt : projection of testing  data  onto the first n_pc  PC’s
    %
    %     W,D  : eigenvectors and eigenvalues of the centralized (cen_K)
    %            training data kernel matrix
    %

    [n,dim]=size(X);
    [nt,dim]=size(Xt);

    %%% training data kernel matrix construction
    K=newKernel(X,type,par1,par2,par3,par4,par5);

    %%% centering of K
    M=eye(n)-ones(n,n)/n;
    cen_K=M*K*M;

    %%% KPCA
   [u,D,W] = svd(cen_K);
    D = diag(D);
    clear u

    %%% training data projection
    for k=1:n_pc
       P_X(:,k)=W(:,k)*sqrt(D(k));
    end

    %%% TESTING PART

    % testing data kernel matrix construction
    Kt=newKernel_Test(X,Xt,type,par1,par2,par3,par4,par5);

    %%% centering of Kt
    Mt=ones(nt,n)/n;
    cen_Kt = (Kt - Mt*K)*M;

    for k=1:n_pc
       Q(:,k)=W(:,k)/(sqrt(D(k)));
    end

    %%% testing  data projection
    P_Xt=cen_Kt*Q;
    size(P_Xt);

function  [Y_hat,Yt_hat,B]=KPCR_cent(P_X,P_Xt,D,Y,Yt)
  %%% Kernel Principal Component Regression - centralized regression model
  %%%
  %%% Inputs
    %     P_X  : projected training data points onto the first p PC's
    %     P_Xt : projected testing  data points onto the first p PC's
    %     D    : (at least) first p ordered (maximal first) eigenvalues of
    %            the centralized training data kernel matrix
    %     Y    : zero mean training outputs (number of samples  x dim)
    %     Yt   : zero mean testing  outputs (number of samples  x dim)
    %
    %     Outputs:
    %
    %     Y_hat  : predicted training outputs (number of samples  x dim)
    %     Yt_hat : predicted testing  outputs (number of samples  x dim)
    %     B      : matrix (or vector) of regression coefficients (p x dim)

   [n,p]=size(P_X);
   D=D(1:p)'; %%% only the first p-eigenvalues are used

   B=diag(1./D)*P_X'*Y;
   Y_hat=P_X*B;
   Yt_hat=P_Xt*B;
```

```
function Hs=newKernel(X,type,par1,par2,par3,par4,par5)

%%% kernel (Gram) matrix computation - training data
%%%
%%% Inputs
%     X - N x dim matrix of input data (number of samples  x dimension)
%                                                        par1    par2
%     type:  'G' Gaussian   Kernel  exp((|x-y|^2)/par1)
%            'P' Polynomial Kernel  (<x,y>)^par2:
%            'S' Sigmoid    Kernel  tanh(par3*<x,y>^par4+par5)
%            'L' linear     kernel  <x,y>
%%  Output
%     Hs - N x N  kernel matrix
[N,dim]=size(X);
Hs=zeros(N,N);
if type=='G'
    for i=1:N
    Hs(i,i)=0;
     for j=i+1:N
       Hs(i,j)=norm(X(i,:)-X(j,:))^2;
       Hs(j,i)=Hs(i,j);
     end
    end
    Hs=exp(-Hs/par1);
end
if  type=='P'
   for i=1:N
     for j=i:N
      dp=(dot(X(i,:),X(j,:)))^par2;
      Hs(i,j)=dp;
      Hs(j,i)=Hs(i,j);
     end
   end
end

if  type=='S'
   for i=1:N
     for j=i:N
      dp=tanh(par3*(dot(X(i,:),X(j,:)))^par4+par5);
      Hs(i,j)=dp;
      Hs(j,i)=Hs(i,j);
     end
   end
end

if  type=='L'
   Hs=X*X';
end
```

```
function [K_tst]=Kernel_Test(X,Xt,type,par1,par2,par3,par4,par5)
%%% kernel (Gram) matrix computation - testing data
%%%
%%% Inputs
%     X -  N  x dim matrix of training input data (number of samples  x dimension)
%     Xt - Nt x dim matrix of testing  input data (number of samples  x dimension)
%                                                       par1    par2
%     type:  ’G’ Gaussian   Kernel  exp((|x-y|^2)/par1)
%            ’P’ Polynomial Kernel  (<x,y>)^par2:
%            ’S’ Sigmoid    Kernel  tanh(par3*<x,y>^par4+par5)
%            ’L’ linear     kernel  <x,y>
%%  Output
%     Hs - N x N  kernel matrix
[N,dim]=size(X);
[Nt,dim]=size(Xt);

K_tst = zeros(Nt,N);

if type==’G’
 for i=1:Nt
   for j=1:N
     K_tst(i,j) = norm(Xt(i,:)-X(j,:))^2 ;
   end
 end
 K_tst=exp(-K_tst/par1);
end

if type==’P’
 for i=1:Nt
   for j=1:N
     K_tst(i,j)=(dot(Xt(i,:),X(j,:)))^par2;
   end
 end
end

if type==’S’
 for i=1:Nt
   for j=1:N
     K_tst(i,j)=tanh(par3*dot(Xt(i,:),X(j,:))^par4+par5);
   end
 end
end

if type==’L’
    K_tst=Xt*X’;
end
```

```
function [Y_hat_lin, Yt_hat_lin]=linregression(X,Xt,Y)
    V_ones=ones(size(X,1),1);
    X=[V_ones,X];
    V_ones=ones(size(Xt,1),1);
    Xt=[V_ones Xt];
    beta=inv(X'*X)*X'*Y;
    Y_hat_lin=X*beta;
    Yt_hat_lin =Xt*beta;
```