Nonlinear Predictions in Regression Models Based on Kernel Method

July 2009

Antoni Wibowo

Nonlinear Predictions in Regression Models Based on Kernel Method

Graduate School of Systems and Information Engineering

University of Tsukuba

July 2009

Antoni Wibowo

Nonlinear Predictions in Regression Models Based on Kernel Method

Antoni Wibowo

(Doctoral Program in Policy and Planning Sciences)

Advised by Professor Yoshitsugu Yamamoto

Submitted to the Graduate School of Systems and Information Engineering
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Engineering
at the University of Tsukuba

July 2009

Abstract

Regression analysis is one of the important techniques in multivariate data analysis in which the ordinary linear regression (OLR) model has been extensively applied. However, OLR has limitations on applications. First, OLR yields only a linear prediction and is sensitive to multicollinearity (collinearity) in which multicollinearity (collinearity) can seriously deteriorate the prediction. To eliminate the effects of multicollinearity (collinearity), we can use principal component regression (PCR) or ridge regression (RR). However, those methods still yield only a linear prediction. To overcome the limitations of linearity and multicollinearity (collinearity), we can use kernel principal component regression (KPCR) or kernel ridge regression (KRR). The basic idea of the KPCR (KRR) is that the set of given data is mapped to a high dimensional space by a function, say ψ, and construct OLR model in the high dimensional space. The important point is that the function ψ is not explicitly defined. Instead of choosing ψ explicitly, we choose a kernel function κ. Then, the KPCR (KRR) is obtained by using the kernel κ. We refer to the procedures to obtain the nonlinear predictions by using the kernel κ as the *kernel method*. However, the previous works of KPCR has theoretical difficulty in the procedure to derive the prediction of KPCR. In this dissertation, we revise the previous works of KPCR to overcome its limitations. Afterwards, we compare the performance of the revised KPCR with the Nadaraya-Watson regression. Our case studies showed that the revised KPCR gives the better results than the Nadaraya-Watson regression.

Second, OLR model uses the assumption that random errors have equal values. In some cases, however, we face the regression model with random errors having unequal variances. We noticed that OLR uses *ordinary least squares* (OLS) method to obtain the regression coefficients. In this regression model, the OLS estimator and hypothesis testing based on the OLS estimator become invalid [14, 27, 30, 41, 44]. Weighted least-squares (WLS) is widely used to overcome the limitations. However, applying WLS in linear regression still yields only a linear prediction and there is no guarantee that the effects of multicollinearity (collinearity) can be avoided. Therefore, we

propose two methods, a combination of WLS and KPCR (WLS-KPCR) and a combination of WLS and KRR (WLS-KRR), to eliminate the limitations of linearity and multicollinearity (collinearity) in this regression model. The basic idea of the WlS-KPCR (WlS-KRR) is that the set of given data is also mapped to a high dimensional space and construct OLR model in the high dimensional space. Then, we apply the WLS method to the OLR model. Afterwards, WLS-KPCR (WLS-KRR) is obtained by applying the kernel method. Our case study showed that WLS-KPCR and WLS-KRR give the better results than that of the WLS method, the revised KPCR and KRR for the regression model with random errors having unequal variances.

Third, we also noticed that the main disadvantage of the OLS method is its sensitivity to outliers. If the outliers are contained in the observed data, the predictions of OLR, PCR, RR, KPCR and KRR can be inappropriate to be used. To eliminate the effects of outliers, we can use a robust regression method. The *M-estimation*, which was first introduced by Huber in 1964, is one of the most widely used methods for the robust regression, which however yields only a linear prediction. Fomengko *et al.* [13] proposed a nonlinear robust prediction based on the M-estimation. Their method, however, needs a specific nonlinear regression model in advance. In this dissertation, we propose two nonlinear robust methods without the need of specifying a nonlinear model in advance. Our proposed methods are a combination of M-estimation and KPCR (R-KPCR) and a combination of M-estimation and KRR (R-KRR). The basic idea of the R-KPCR (R-KRR) is that the set of given data is also mapped to a high dimensional space and construct OLR model in the high dimensional space. Then, we apply the kernel method and M-estimation to the OLR model. Our case study showed that R-KPCR and R-KRR give the better results than that of the robust linear regression based on M-estimation, the revised KPCR and KRR for the set of data that are contaminated by outliers.

Acknowledgements

First of all, I would like to sincerely thank my supervisor, Professor Yoshitsugu Yamamoto, for his many suggestions and constant supports during this research. He kept me steered in the right direction, care and good judgments. His insights and ideas formed the foundation of this dissertation as much as mine did, and his guidance helped me to get over the various hurdles.

I also would like to thank Professor Akiko Yoshise, Professor Maiko Shigeno, Professor Yuichi Kanazawa, Professor Hideo Suzuki and Professor Masahiro Hachimori for their constructive advices and suggestions. I also wish to thank the Ministry of Education, Culture, Sports, Science and Technology Japan for financial support during my graduate years.

Many thanks also go to the members of Yamamoto's laboratory: Yuichi Takano, Naoya Ogawa and Satoko Ryuo, for providing a stimulating environment during my laboratory life. I am also grateful to Professor Junya Gotoh, Hidetomo Hanatsuka, Ayami Suzuka, Daisuke Zenke, Hidetoshi Nagai, Masafumi Tsurutani, Ryoya Kawai, Daisuke Sato, Kazuko Ohbo and Hisaaki Nakane for their kindness and friendly cooperation.

Last but not least, I am also grateful to my parents and my parents-in-law, my wife Kartika Rachmawati and my daughter Mahadewi Antika Salsabila for their patience and *love* that always cheer me up. Without them this work would never have come into existence.

Japan — Antoni Wibowo

July, 2009

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we present ordinary linear regression (OLR) model that is widely used in regression analysis. Then, we introduce the limitations of the OLR model. Further, we present our motivations for considering of nonlinear regression models which are used to overcome the limitations of the OLR model. The purposes, results and the outline of this dissertation are also presented.

1.1 Background

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 [27]. The *ordinary linear regression* (OLR) model with p regressors is given by

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \epsilon. \tag{1.1.1}$$

The parameters β_j $(j = 0, 1, \ldots, p)$ are called the *regression coefficients* and ϵ is a random variable called the *random error*. It is assumed that the values of $x_1, x_2, \ldots, x_p$ are chosen by an experimenter and β_j's are unknown.

Let Y_i be the response variable on the ith observation $(i = 1, 2, \ldots, N)$, $x_{ij} \in \mathbb{R}$ be the ith observation of regressor x_j and ϵ_i be the random error on the ith observation where $\mathbb{R}$ is the set of real numbers. We denote $\mathbf{x}_i = (x_{i1} \quad x_{i2} \quad \ldots \quad x_{ip})^T$, $\mathbf{Y} = (Y_1 \quad Y_2 \quad \ldots \quad Y_N)^T$, $\tilde{\mathbf{X}} = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \ldots \quad \mathbf{x}_N)^T$, $\mathbf{X} = \begin{pmatrix} \mathbf{1}_N & \tilde{\mathbf{X}} \end{pmatrix}$, $\boldsymbol{\beta} = (\beta_0 \quad \beta_1 \quad \ldots \quad \beta_p)^T$, and $\boldsymbol{\epsilon} = (\epsilon_1 \quad \epsilon_2 \quad \ldots \quad \epsilon_N)^T$, 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 $\mathbf{1}_N = (1 \quad 1 \quad \ldots \quad 1)^T_{N \times 1}$. The vector $\mathbf{x}_i^T$ denotes the transpose of the vector $\mathbf{x}_i$.

Then, the *standard ordinary linear regression model* corressponding to Eq. (1.1.1) is written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{1.1.2}$$

It is assumed that the expected value of $\boldsymbol{\epsilon}$, denoted by $E(\boldsymbol{\epsilon})$, is equal to $\mathbf{0}$ and the variance matrix of $\boldsymbol{\epsilon}$, denoted by $E(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T)$, is equal to $\sigma^2\mathbf{I}_N$, where the matrix $\mathbf{I}_N$ denotes the $N \times N$ identity matrix and $\sigma^2 \in \mathbb{R}_+$. Matrix $\mathbf{X}$ is called the *regression matrix*.

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

$$\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{Y}. \tag{1.1.3}$$

Eq. (1.1.3) is called the *least squares normal equations*. The procedure to obtain $\hat{\boldsymbol{\beta}}$ by solving Eq. (1.1.3) is called the *ordinary least squares* (*OLS*) method. Note that, $\mathbf{X}^T\mathbf{X}$ is a symmetric and positive semidefinite matrix, implying that the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are nonnegative real numbers [1]. 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 [27, 41]. If collinearity exists on $\mathbf{X}$ then there are infinitely many solutions of Eq. (1.1.3), which makes it difficult to choose the "best" linear multiple regression model. This implication is known as

the *effect of collinearity*.

In addition, we say that *multicollinearity* exists on $\mathbf{X}$ if $\mathbf{X}^T\mathbf{X}$ is a nearly singular matrix, i.e., if some eigenvalues of $\mathbf{X}^T\mathbf{X}$ are close to zero. In [22, 25, 30, 27, 41, 44], they considered the standard multiple linear regression model where the column vectors of $\mathbf{X}$ are linearly independent. In this case, the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are positive real numbers, and the variance of $\hat{\beta}_j$ for $j = 0, 1, \ldots, p$, denoted by $Var(\hat{\beta}_j)$, is given by

$$Var(\hat{\beta}_j) = \sigma^2((\mathbf{X}^T\mathbf{X})^{-1})_{j+1,j+1}, \qquad j = 0, 1, \ldots, p. \tag{1.1.4}$$

where $(\mathbf{X}^T\mathbf{X})^{-1}$ is the inverse of $\mathbf{X}^T\mathbf{X}$. If multicollinearity exists on $\mathbf{X}$ then the estimator of some β_j can be have wrong sign [5], $Var(\hat{\beta}_j)$ can be a large number and under the assumption that ε_i is normally distributed, the tests for inferences β_j ($j = 0, 1, \ldots, p$) have low power and the confidence interval can be large [44]. Therefore, it will be difficult to decide if a variable x_j makes a significant contribution to the regression. These implications are known as the *effects of multicollinearity*.

After the observations are taken, we obtain the observed data corresponding to $\mathbf{Y}$. Let $\mathbf{y} = (y_1 \quad y_2 \quad \ldots \quad y_N)^T \in \mathbb{R}^N$ be the observed data corresponding to $\mathbf{Y}$ and $\hat{\boldsymbol{\beta}}^* = \left(\hat{\beta}_0^* \quad \hat{\beta}_1^* \quad \ldots \quad \hat{\beta}_p^*\right)^T \in \mathbb{R}^{p+1}$ be the value of $\hat{\boldsymbol{\beta}}$ when $\mathbf{Y}$ is replaced by $\mathbf{y}$ in the Eq. (1.1.3). Under the assumption that the column vectors of $\mathbf{X}$ are linearly independent, we obtain

$$\hat{\boldsymbol{\beta}}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{1.1.5}$$

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

$$\hat{\mathbf{y}} = \left(\hat{y}_1 \quad \hat{y}_2 \quad \ldots \quad \hat{y}_N\right)^T := \mathbf{X}\hat{\boldsymbol{\beta}}^*, \tag{1.1.6}$$

and the *residual* between $\mathbf{y}$ and $\hat{\mathbf{y}}$ is given by

$$\hat{\mathbf{e}} = \left(\hat{e}_1 \quad \hat{e}_2 \quad \ldots \quad \hat{e}_N\right)^T := \mathbf{y} - \hat{\mathbf{y}}. \tag{1.1.7}$$

The *root mean square error* (*RMSE*) by OLR is given by

$$RMSE_{olr} := \sqrt{\frac{\hat{\mathbf{e}}^T\hat{\mathbf{e}}}{N}}, \tag{1.1.8}$$

and the *prediction by OLR* is given by

$$f_{olr}(\mathbf{x}) := \hat{\beta}_0^* + \sum_{j=1}^{p} \hat{\beta}_j^* x_j, \tag{1.1.9}$$

where $\mathbf{x} = (x_1 \quad x_2 \quad \dots \quad x_p)^T \in \mathbb{R}^p$ and f_{olr} is a function from $\mathbb{R}^p$ to $\mathbb{R}$.

1.2 Motivations

As known that OLR model yields a linear prediction which has limitations on applications since the most of real problems are nonlinear. Beside that, OLR is sensitive to multicollinearity (collinearity) where existence of multicollinearity (collinearity) on matrix regression can seriously deteriorate the prediction by OLR. To avoid the effects of multicollinearity (collinearity), we can use *principal component regression* (*PCR*) or *ridge regression* (*RR*). However, those methods also yield linear prediction. To overcome the limitation of linearity, Rosipal *et al.* [33, 34, 35], Hoegaerts *et al.* [18] and Jade *et al.* [20] used the *kernel principal component regression* (*KPCR*). However, the proposed KPCR still has theoretical difficulty in the procedure to derive the prediction of KPCR. Therefore, we revise the proposed KPCR to overcome the difficulty.

In some cases, however, we face the regression model with variance of random errors having unequal values in diagonal elements. *Weighted least-squares* (*WLS*) is a widely used to handle the limitations. However, applying WLS in linear regression yields a linear prediction model and there is no guarantee that this method can avoid the effects of multicollinearity. Although we can use KPCR and KRR to overcome the limitations of linearity and multicollinearity, KPCR and KRR can be inappropriate in this regression model. Since KPCR and KRR were constructed on the different assumption, that

is, the variance of random errors having equal values in diagonal elements.

We also noticed that the main disadvantage of the OLS method is its sensitivity to *outliers*, i.e., residuals of the observed data are large numbers. Outliers have a large influence on the prediction value because squaring residuals magnifies the effect of the outliers. If the outliers are contained in the observed data, the predictions of OLR, PCR, RR, KPCR and KRR become inappropriate to be used, which are referred to as the *effects of outliers*, since those methods were constructed based on OLS method.

1.3 Purposes and Results

Kernel-based approaches to classification and regression have become very popular in recent years [56], following their introduction by Vapnik [47] and their further development by many researchers, see for examples [9, 40]. The basic insight gained by Vapnik was that problems that are difficult to solve in low dimensions space and can be easier when the set of given data is mapped to a high dimensional space, i.e., $\mathbf{x}_i$ is mapped into $\mathcal{F}$ by using a function $\psi : \mathbb{R}^p \to \mathcal{F}$ for $i = 1, 2, \ldots, N$. The set $\mathcal{F}$ is called the *feature space* which we assume is an Euclidean space of higher dimension than p, say p_F. The important point is that the function ψ is not explicitly defined. We say that a function φ is 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 a *positive semidefinite function* if for every natural number m, and for $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m \in \mathbb{R}^p$ it gives rise to a positive semidefinite matrix $\mathbf{W} = (\varphi(\mathbf{w}_i, \mathbf{w}_j))_{i,j=1,2,\ldots,m}$ (see [40] for the detailed discussion). The function ψ is provided by another function[1], say $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, where κ is a symmetric, continuous and positive semidefinite function [10, 26, 39]. The function κ is called the *kernel* function.

As mentioned before, the previous works of KPCR have theoretical difficulties in the procedure to derive the prediction of KPCR. In this dissertation, we revise the KPCR to overcome its limitations. The procedure to derive our proposed KPCR is straightforward as the procedure of PCR, except that

[1]See Theorem 2.2.2 (Mercer Theorem).

Table 1.1: The linear and nonlinear predictions in regression analysis which are observed in this dissertation (Our proposed methods to obtain a nonlinear prediction are marked by *).

	Linear	Nonlinear (Non Kernel)	Nonlinear (Kernel)
OLS	OLR	Jukic's Regression	KPCR Revised KPCR*
Ridge	RR		KRR
WLS	WLS LR		WLS KPCR* WLS KRR*
Robust	M-Estimation	Famenko *et al.* [13] based on M-estimation	R-KPCR* R-KRR*
Nonparametric		Nadaraya [28] and Watson [48]	

some mathematical techniques are done to obtain a nonlinear prediction and to avoid the effects of multicollinearity (collinearity). Another technique to overcome the limitation of linearity and multicollinearity (collinearity) is *kernel ridge regression* (*KRR*) which was studied by Hoegaerts *et al.* [18], Rosipal *et al.* [33, 34, 35], and Saunders *et al.* [37]. The procedure to derive the KRR is also straightforward as the procedure of RR with applying some mathematical techniques to obtain a nonlinear prediction and to avoid the effects of multicollinearity (collinearity). We refer to the procedures to obtain the nonlinear predictions by using kernels as the *kernel method*.

However, if we face the regression model with variance of random errors having unequal values in diagonal elements, KPCR and KRR become inappropriate to be used. Here, we propose two methods, which are a combination of WLS and KPCR and a combination of WLS and KRR, to overcome the limitation of KPCR and KRR in this regression model. These methods yield nonlinear predictions and they can also avoid the effects of multicollinearity (collinearity).

The predictions of OLR, PCR, RR, KPCR and KRR can also be inappropriate to be used when outliers are contained in the observed data. A robust regression method is widely used technique to eliminate the effects of outliers. *M-estimation* is one of the most widely used methods for the robust regression, where the method yields a linear prediction. We noticed that

Fomengko *et al.* [13] proposed a nonlinear robust prediction based on the M-estimation; their method, however, needs a specific nonlinear regression model in advance. In this dissertation, we propose two methods, which are a combination of M-estimation and KPCR and a combination of M-estimation and KRR, to obtain a nonlinear robust prediction without specifying a nonlinear model in advance. Furthermore, we compare the proposed methods with some other methods. The linear and the nonlinear predictions in regression analysis which are compared in this dissertation are given in Table 1.1.

We noticed that the Nadaraya-Watson regression is categorized as a kernel regression in statistic community. In statistics, *kernel regression* is a non-parametric technique to estimate the conditional expectation of a random variable. In the Nadaraya-Watson regression, we do not need to map the given data to a high dimensional space. The idea of prediction by Nadaraya-Watson is given in the following manner. Let $Y, X_1, X_2, \ldots, X_p$ be the random variables defined on a sample space Ω and $(X_1, X_2, \ldots, X_p)^T$ be the p-dimensional random vector. We assume that $\mathbf{x}_i^T$, $(i = 1, 2, ..., N)$, is the observed data corresponding to $(X_1, X_2, \ldots, X_p)^T$. Let $g_1(y, \mathbf{x})$ be the joint probability density function of $Y, X_1, X_2, \ldots, X_p$ and $g_2(\mathbf{x})$ be the joint probability density function of $X_1, X_2, \ldots, X_p$. The expected value of Y given the value $\mathbf{X} = \mathbf{x}$ is given by

$$E(Y|\mathbf{X} = \mathbf{x}) = \frac{\int_{-\infty}^{\infty} y g_1(y, \mathbf{x}) dy}{g_2(\mathbf{x})}. \tag{1.3.1}$$

We assume that $g_2(\mathbf{x})$ is not equal to zero. The unknown quantities on the right hand side of (1.3.1) are $g_1(y, \mathbf{x})$ and $g_2(\mathbf{x})$. To estimate $g_1(y, \mathbf{x})$ and $g_2(\mathbf{x})$, we employ the *multiplicative kernel density estimators* of $g_1(y, \mathbf{x})$ and $g_2(\mathbf{x})$ which are given by

$$\hat{g}_1(y, \mathbf{x}) := \frac{1}{N h_1^p h_2} \sum_{i=1}^{N} \prod_{j=1}^{p} \kappa_1(\frac{x_j - x_{ij}}{h_1}) \kappa_1(\frac{y - y_i}{h_2}), \tag{1.3.2}$$

and

$$\hat{g}_2(\mathbf{x}) := \frac{1}{Nh_1^p} \sum_{i=1}^{N} \prod_{j=1}^{p} \kappa_1(\frac{x_j - x_{ij}}{h_1}), \tag{1.3.3}$$

respectively; where κ_1 is a function from $\mathbb{R}$ to $\mathbb{R}$, y is a value of Y, h_1 and h_2 are smoothing parameters of the multiplicative kernel density estimators. We assume that $\hat{g}_2(\mathbf{x})$ is not equal to zero. The function κ_1 is required to satisfy the following conditions:

$$\int_{-\infty}^{\infty} \kappa_1(u) d(u) = 1, \tag{1.3.4}$$

$$\int_{-\infty}^{\infty} u\kappa_1(u) d(u) = 0, \tag{1.3.5}$$

and

$$\kappa_1(u) \geq 0 \qquad \text{for any } u \in \mathbb{R}. \tag{1.3.6}$$

Then, the prediction by Nadaraya-Watson is given by

$$\begin{aligned} \hat{m}(\mathbf{x}) &:= \frac{\int_{-\infty}^{\infty} y\hat{g}_1(y, \mathbf{x}) dy}{\hat{g}_2(\mathbf{x})} \\ &= \frac{\sum_{i=1}^{N} \prod_{j=1}^{p} \kappa_1(\frac{x_j - x_{ij}}{h_1}) y_i}{\sum_{i=1}^{N} \prod_{j=1}^{p} \kappa_1(\frac{x_j - x_{ij}}{h_1})}. \end{aligned} \tag{1.3.7}$$

The "best" value of h_1 can be chosen by several methods, see example [31, 16]. Readers may consult other literatures for the detailed discussion, for example [31, 16, 28, 29, 38, 48, 17].

1.4 Outline of This Dissertation

This dissertation is organized as follows: In Chapter 2, we present principal component analysis and apply kernel method in principal component analysis to obtain a nonlinear principal component analysis. In Chapter 3, we present PCR and RR as linear methods for dealing with multicollinearity (collinearity). We also consider WLS as a linear method for regression model with variance of random errors having unequal values in diagonal elements

and M-estimation as a linear method for dealing with the observed data that are contaminated by outlier.

In Chapter 4, we show the theoretical difficulty of the previous works of KPCR and revise it. We also propose a combination of WLS and KPCR to overcome the limitation of WLS method, while a combination of M-Estimation and KPCR as a nonlinear method for dealing with outliers is given in the end of this chapter. In Chapter 5, we review KRR as an alternative nonlinear method for dealing with multicollinearity (collinearity). In this chapter, we also propose a combination of WLS and KRR to overcome the limitation of WLS method. Afterward, we present a combination of M-Estimation and KRR as an alternative nonlinear method for dealing with outliers.

In Chapter 6, we present some case studies in which our proposed methods are compared with some other methods. Then, conclusions are presented in Chapter 7. Finally, some important concepts of random vectors (matrices) and proofs of some of theorems and lemmas are given in appendices.

Chapter 2

Principal Component Analysis and Kernel Principal Component Analysis

In this chapter, we introduce principal components analysis (PCA) and a nonlinear PCA which is obtained by applying kernel method in PCA. These methods will be used to avoid the effects of multicollinearity and collinearity in regression models. PCA is an orthogonal transformation of a coordinate system. The new coordinate values by which we represent the data are called the *principal components* [39]. PCA and the nonlinear PCA are introduced in Subchapter 3.1 and Subchapter 3.2, respectively.

2.1 Principal Component Analysis

PCA is one of the most important techniques in multivariate data analysis. It is often applied to multivariate data analysis, such as to visualize the data structure, to detect outliers and to reduce the data dimensionality, and many important methods based on PCA [55]. The idea of PCA was firstly introduced by Pearson (1901) and developed independently by Hoteling (1933) [22]. Even though the idea of PCA appeared about 100 years ago, PCA research and applications are still very hot topics [11]. There have been a number of survey papers and books that have reported on PCA algorithms.

For examples, [8, 20, 24, 36, 45] have been published in the last decade.

Let us first start with a set of N centered data of $\tilde{\mathbf{X}}$ which is given by the following matrix

$$\mathbf{Z} := (\mathbf{z}_1 \quad \mathbf{z}_2 \quad \cdots \quad \mathbf{z}_N)^T = (I_N - \tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\tilde{\mathbf{X}},$$

where sizes of $\mathbf{Z}$ is $N \times p$ and $\tilde{\mathbf{X}}$ is given in the previous chapter. Then, the *sample covariance matrix of* $\mathbf{Z}$ is given by

$$\begin{aligned} \mathbf{C} &:= \frac{1}{N}\mathbf{Z}^T\mathbf{Z} \\ &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{z}_i\mathbf{z}_i^T, \end{aligned} \tag{2.1.1}$$

which has the following important properties.

Lemma 2.1.1. $\mathbf{C}$ *is a symmetric matrix.*

Proof.

$$\mathbf{C}^T = \frac{1}{N}(\mathbf{Z}^T\mathbf{Z})^T = \frac{1}{N}\mathbf{Z}^T\mathbf{Z} = \mathbf{C}.$$

□

Lemma 2.1.2. $\mathbf{C}$ *is a positive semidefinite matrix.*

Proof. Let $\mathbf{d}$ be a vector in $\mathbb{R}^p$. Then,

$$\begin{aligned} \mathbf{d}^T\mathbf{C}\mathbf{d} &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{d}^T\mathbf{z}_i\mathbf{z}_i^T\mathbf{d} \\ &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{d}^T\mathbf{z}_i)^2 \geq 0 \qquad (\text{Since } \mathbf{d}^T\mathbf{z}_i = \mathbf{z}_i^T\mathbf{d} \text{ and real numbers}). \end{aligned}$$

□

By using Theorem (A.9), the eigenvalues of $\mathbf{C}$ are nonnegative real numbers. Let $\lambda_1, \lambda_2, \ldots, \lambda_p$ be the eigenvalues of $\mathbf{C}$ and $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p$ be normalized eigenvector of $\mathbf{C}$ corresponding to $\lambda_1, \lambda_2, \ldots, \lambda_p$, respectively.

Lemma 2.1.3. *Trace of* $\mathbf{C}$ *is given by*

$$tr(\mathbf{C}) = \frac{1}{N} \sum_{i=1}^{p} \lambda_i. \tag{2.1.2}$$

Proof. Since $\mathbf{Z}^T\mathbf{Z}$ is symmetric, $\mathbf{Z}^T\mathbf{Z}$ is orthogonally diagonalizable (by using Theorem A.6). Hence, there exists $\mathbf{Q} \in \mathbb{R}^{p\times p}$ and $\mathbf{Q}^T = \mathbf{Q}^{-1}$ such that $\mathbf{Q}^T\mathbf{Z}^T\mathbf{Z}\mathbf{Q} = \mathbf{D}$ where

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \lambda_p \end{pmatrix}.$$

This implies $\mathbf{Z}^T\mathbf{Z} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T = \mathbf{Q}\mathbf{D}\mathbf{Q}^{-1}$. Hence,

$$\begin{aligned} Tr(\mathbf{Z}^T\mathbf{Z}) &= Tr(\mathbf{Q}\mathbf{D}\mathbf{Q}^{-1}) \\ &= Tr(\mathbf{D}\mathbf{Q}\mathbf{Q}^{-1}) \\ &= Tr(\mathbf{D}) \\ &= \sum_{i=1}^{p} \lambda_i. \end{aligned} \tag{2.1.3}$$

Hence, we obtain that $Tr(\mathbf{C}) = \frac{1}{N}\sum_{i=1}^{p} \lambda_i$. □

Let us choose an arbitrary observation of $\mathbf{Z}$, say $\mathbf{z}_k \in \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\} \subseteq \mathbb{R}^p$. Since $\mathbb{R}^p$ is a finite dimensional inner product space, $\mathbb{R}^p$ has an orthonormal basis. Let $\{\mathcal{H}_l\}_{l\in\mathcal{I}}$ be a collection of q dimensional subspaces which $\mathcal{H}_l$ has orthonormal basis $\{\mathbf{u}_{l1}, \mathbf{u}_{l1}, \ldots, \mathbf{u}_{lq}\}$ for every $l \in \mathcal{I}$, where $\mathcal{I}$ is an

index set. Let us choose an arbitrary $\mathcal{H} \in \{\mathcal{H}_l\}_{l\in\mathcal{I}}$ and $\{\mathbf{u}_1, \mathbf{u}_1, \ldots, \mathbf{u}_q\}$ be the orthonormal basis for $\mathcal{H}$. Moreover, we define

$$\begin{aligned}
\mathbf{U} &:= (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \ldots \quad \mathbf{u}_q), \\
\hat{\mathbf{z}}_k &:= Proj_{\mathcal{H}}\mathbf{z}_k = \sum_{i=1}^{q} (\mathbf{z}_k^T \mathbf{u}_i)\mathbf{u}_i, \\
y_{ki} &:= \mathbf{z}_k^T \mathbf{u}_i = \mathbf{u}_i^T \mathbf{z}_k, \text{for } i = 1, 2, \ldots, q,
\end{aligned}$$

and construct the vector

$$\mathbf{y_k} = \begin{pmatrix} y_{k1} \\ y_{k2} \\ \cdot \\ \cdot \\ y_{kq} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \cdot \\ \cdot \\ \mathbf{u}_q^T \end{pmatrix} \mathbf{z}_k = \mathbf{U}^T \mathbf{z}_k.$$

Note that $\hat{\mathbf{z}}_k$ is the projection $\mathbf{z}_k$ onto subspace $\mathcal{H}$. Then, the vector $\hat{\mathbf{z}}_k$ can now be written as

$$\hat{\mathbf{z}}_k = \sum_{i=1}^{q} y_{ki}\mathbf{u}_i = \mathbf{U}\mathbf{y}_k = \mathbf{U}\mathbf{U}^T\mathbf{z}_k. \tag{2.1.4}$$

The aim of PCA is to find the best representation of $\mathbf{z}_k$ $(k = 1, 2, ..., N)$, say $Proj_{\hat{\hat{H}}}\mathbf{z}_k$, such that

$$\sum_{i=1}^{N} \|\mathbf{z}_i - Proj_{\hat{\mathcal{H}}}\mathbf{z}_i\|^2 \leq \sum_{i}^{N} \|\mathbf{z}_i - Proj_{\mathcal{H}}\mathbf{z}_i\|^2 \quad \text{for } \mathcal{H} \in \{\mathcal{H}_l\}_{l\in\mathcal{I}} \tag{2.1.5}$$

or equivalent to

$$\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{N} \|\mathbf{z}_i - Proj_{\mathcal{H}}\mathbf{z}_i\|^2 \\
\text{s.t.} \quad & \mathcal{H} \in \{\mathcal{H}_l\}_{l\in\mathcal{I}}.
\end{aligned} \tag{2.1.6}$$

Problem (2.1.6) can be written as

$$\begin{aligned} &\min && \textstyle\sum_{i=1}^{N} \|\mathbf{z}_i - \sum_{k=1}^{q} (\mathbf{z}_k^T \mathbf{u}_k)\mathbf{u}_k\|^2 \\ &\text{s.t.} && \mathbf{u}_j^T \mathbf{u}_k = \begin{cases} 1 & \text{if } j = k, \quad \text{for } j,k = 1,2,\ldots,q, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2.1.7}$$

Let $\mathbf{r}_i = \mathbf{z}_i - \sum_{k=1}^{q} (\mathbf{z}_k^T \mathbf{u}_k)\mathbf{u}_k = \mathbf{z}_i - \hat{\mathbf{z}}_i$. This implies that $\mathbf{z}_i^T \mathbf{z}_i = (\hat{\mathbf{z}}_i + \mathbf{r}_i)^T(\hat{\mathbf{z}}_i + \mathbf{r}_i) = \hat{\mathbf{z}}_i^T \hat{\mathbf{z}}_i + 2\hat{\mathbf{z}}_i^T \mathbf{r}_i + \mathbf{r}_i^T \mathbf{r}_i$. Since $\hat{\mathbf{z}}_i$ is an orthogonal projection of $\mathbf{z}_i$ into $\mathcal{H}$, we have $\hat{\mathbf{z}}_i^T \mathbf{r}_i$ is equal zero (by Definition (A.3)). Hence

$$\begin{aligned} \sum_{i=1}^{N} \|\mathbf{z}_i - \sum_{k=1}^{q} (\mathbf{z}_k^T \mathbf{u}_k)\mathbf{u}_k\|^2 &= \sum_{i=1}^{N} \mathbf{r}_i^T \mathbf{r}_i \\ &= \sum_{i=1}^{N} \mathbf{z}_i^T \mathbf{z}_i - \sum_{i=1}^{N} \hat{\mathbf{z}}_i^T \hat{\mathbf{z}}_i. \end{aligned} \tag{2.1.8}$$

By using Lemma (2.1.3), we obtain that $\sum_{i=1}^{N} \mathbf{z}_i^T \mathbf{z}_i = tr(\sum_{i=1}^{N} \mathbf{z}_i^T \mathbf{z}_i) = tr(\sum_{i=1}^{N} \mathbf{z}_i \mathbf{z}_i^T)$ which implies $\sum_{i=1}^{N} \mathbf{z}_i^T \mathbf{z}_i = N \sum_{i=1}^{N} \lambda_i$. Hence, Eq.(2.1.8) becomes

$$\begin{aligned} \sum_{i=1}^{N} \mathbf{r}_i^T \mathbf{r}_i &= N \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} \hat{\mathbf{z}}_i^T \hat{\mathbf{z}}_i \\ &= N \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} (\mathbf{U}\mathbf{U}^T \mathbf{z}_k)^T (\mathbf{U}\mathbf{U}^T \mathbf{z}_k) \\ &= N \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} \mathbf{z}_k^T \mathbf{U}\mathbf{U}^T \mathbf{U}\mathbf{U}^T \mathbf{z}_k \\ &= N \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} \mathbf{z}_k^T \mathbf{U}\mathbf{I}_q \mathbf{U}^T \mathbf{z}_k \\ &= N \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} \mathbf{z}_k^T \mathbf{U}\mathbf{U}^T \mathbf{z}_k. \end{aligned} \tag{2.1.9}$$

Since $\sum_{i=1}^{N} \mathbf{z}_k^T \mathbf{U}\mathbf{U}^T \mathbf{z}_k \in \mathbb{R}$, we have

$$\begin{aligned}
\sum_{i=1}^{N} \mathbf{z}_k^T \mathbf{U}\mathbf{U}^T \mathbf{z}_k &= tr(\sum_{i=1}^{N} \mathbf{z}_k^T \mathbf{U}\mathbf{U}^T \mathbf{z}_k) \\
&= tr(\sum_{i=1}^{N} (\mathbf{z}_k^T \mathbf{U})(\mathbf{U}^T \mathbf{z}_k)) \\
&= tr(\sum_{i=1}^{N} (\mathbf{U}^T \mathbf{z}_k)(\mathbf{z}_k^T \mathbf{U})) \\
&= tr(\sum_{i=1}^{N} \mathbf{U}^T \mathbf{z}_k \mathbf{z}_k^T \mathbf{U}) \\
&= tr(\mathbf{U}^T (\sum_{i=1}^{N} \mathbf{z}_k \mathbf{z}_k^T) \mathbf{U}) \\
&= tr(\mathbf{U}^T (N\mathbf{C}) \mathbf{U}) \\
&= N tr(\mathbf{U}^T (\mathbf{C}) \mathbf{U}) \\
&= N \sum_{k=1}^{q} \mathbf{u}_k^T \mathbf{C} \mathbf{u}_k. \qquad (2.1.10)
\end{aligned}$$

Hence, problem (2.1.7) becomes

$$\begin{aligned}
&\min && N(\textstyle\sum_{i=1}^{N} \lambda_i - \sum_{k=1}^{q} \mathbf{u}_k^T \mathbf{C} \mathbf{u}_k) && (2.1.11) \\
&\text{s.t.} && \mathbf{u}_j^T \mathbf{u}_k = \begin{cases} 1 & \text{if } j = k, \quad \text{for } j, k = 1, 2, \ldots, q, \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}$$

which is equivalent to

$$\begin{aligned}
&N(\sum_{i=1}^{N} \lambda_i - \max \sum_{k=1}^{q} \mathbf{u}_k^T \mathbf{C} \mathbf{u}_k) && (2.1.12) \\
&\text{s.t.} \quad \mathbf{u}_j^T \mathbf{u}_k = \begin{cases} 1 & \text{if } j = k, \quad \text{for } j, k = 1, 2, \ldots, q, \\ 0 & \text{otherwise.} \end{cases}.
\end{aligned}$$

It is evident that $N\sum_{i=1}^{N}\lambda_i$ is a constant. Hence, we solve the following problem

$$\begin{aligned} \max \quad & \sum_{k=1}^{q}\mathbf{u}_k^T\mathbf{C}\mathbf{u}_k \\ \text{s.t.} \quad & \mathbf{u}_j^T\mathbf{u}_k = \begin{cases} 1 & \text{if } j=k, \quad \text{for } j,k=1,2,\ldots,q, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2.1.13}$$

to obtain the optimal solution of problem (2.1.12). Let us consider the following theorem to obtain the optimal solution of the problem.

Theorem 2.1.4. *The optimal solution of problem (2.1.13) is obtained by the choice* $\mathbf{u}_k = \mathbf{a}_k$.

Then, the k-th *principal component* of $\mathbf{x}$, $k = 1, 2, \ldots, q$, is given by

$$y_k(\mathbf{x}) := \mathbf{x}^T\mathbf{a}_k, \qquad \mathbf{x} \in \mathbb{R}^p. \tag{2.1.14}$$

Although PCA has been extensively applied in almost every discipline, chemistry, biology, engineering, meteorology, etc., but there have also been difficulties in its applications. Because PCA is a linear method and most real problems are nonlinear. Applying PCA to nonlinear problems can sometimes be inadequate [11]. To overcome such a drawback, various techniques have been developed by Dong and McAvoy [11], Saegusa *et al.* [36] and so on. Among the nonlinear techniques, Schölkopf *et al.* [39] have developed an attractive algorithm because it does not involve nonlinear optimization, it is as simple as the PCA, and it does not need to specify the number of principal components prior to modeling compared to other nonlinear methods. The technique is called the *kernel principal component analysis* (*KPCA*). In the next subchapter, the detailed KPCA will be presented.

2.2 Kernel Principal Component Analysis

Assume we have a function $\psi : \mathbb{R}^p \to \mathcal{F}$, where $\mathcal{F}$ is the feature space which we have assumed is an Euclidean space of higher dimension, say p_F, than p. Then, we define $\mathbf{\Psi} = \begin{pmatrix} \psi(\mathbf{x}_1) & \dots & \psi(\mathbf{x}_N) \end{pmatrix}^T$, $\tilde{\mathbf{C}} := \frac{1}{N}\mathbf{\Psi}^T\mathbf{\Psi} = \frac{1}{N}\sum_{i=1}^{N}\psi(\mathbf{x}_i)\psi(\mathbf{x}_i)^T$ and $\mathbf{K} = \mathbf{\Psi}\mathbf{\Psi}^T$, where sizes of $\mathbf{\Psi}$, $\tilde{\mathbf{C}}$ and $\mathbf{K}$ are $N \times p_F$, $p_F \times p_F$ and $N \times N$, respectively. We assume that $\sum_{i=1}^{N}\psi(\mathbf{x}_i) = \mathbf{0}$. If $\mathcal{F}$ is infinite-dimensional, we consider the linear operator $\psi(\mathbf{x}_i)\psi(\mathbf{x}_i)^T$ instead of the matrix $\tilde{\mathbf{C}}$ [40]. The relation of eigenvalues and eigenvectors of the matrices $\tilde{\mathbf{C}}$ and $\mathbf{K}$ was studied by Scholkopf *et al.* [40]. However, we restate it in the following theorem.

Theorem 2.2.1. [50, 51, 53] *Suppose* $\hat{\lambda} \neq 0$ *and* $\hat{\mathbf{a}} \in \mathcal{F} \setminus \{\mathbf{0}\}$. *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* $\mathbf{a} = \mathbf{\Psi}^T\mathbf{b}$,
   for some $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \dots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

3. $\hat{\lambda}$ *and* $\hat{\mathbf{a}}$ *satisfy* $\lambda N\tilde{\boldsymbol{b}} = \mathbf{K}\tilde{\boldsymbol{b}}$ *and* $\mathbf{a} = \mathbf{\Psi}^T\tilde{\mathbf{b}}$,
   for some $\tilde{\mathbf{b}} = \begin{pmatrix} \tilde{b}_1 & \tilde{b}_2 & \dots & \tilde{b}_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Proof. See Appendix B.1. □

Let $\hat{p}_F$ be the rank of $\mathbf{\Psi}$ where $\hat{p}_F \leq \min\{N, p_F\}$. It is well known that the rank($\mathbf{\Psi}$) is equal to rank($\mathbf{K}$) and rank ($\mathbf{\Psi}^T\mathbf{\Psi}$). Hence, the rank($\mathbf{K}$) and rank($\mathbf{\Psi}^T\mathbf{\Psi}$) are equal to $\hat{p}_F$. As we see that the matrix $\mathbf{K}$ is symmetric and positive semidefinite, implying that the eigenvalues of $\mathbf{K}$ are nonnegative real numbers. Let $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_{\tilde{r}} \geq \tilde{\lambda}_{\tilde{r}+1} \geq \ldots \geq \tilde{\lambda}_{\hat{p}_F} > \tilde{\lambda}_{\hat{p}_F+1} = \ldots = \tilde{\lambda}_N = 0$ be the eigenvalues of $\mathbf{K}$ and $\tilde{\mathbf{B}} = (\tilde{\mathbf{b}}_1 \quad \tilde{\mathbf{b}}_2 \quad \dots \quad \tilde{\mathbf{b}}_N)$ be the matrix of the corresponding normalized eigenvectors $\tilde{\mathbf{b}}_l$ of $\mathbf{K}$. Then, let

$\boldsymbol{\alpha}_l = \begin{pmatrix} \alpha_{l1} & \alpha_{l2} & \dots & \alpha_{lN} \end{pmatrix}^T = \frac{\tilde{\mathbf{b}}_l}{\sqrt{\tilde{\lambda}_l}}$ and $\tilde{\mathbf{a}}_l = \boldsymbol{\Psi}^T \boldsymbol{\alpha}_l$ for $l = 1, 2, \dots, \hat{p}_F$. By Theorem 2.2.1 we obtain

$$\begin{aligned} \tfrac{\tilde{\lambda}_l}{N} \tilde{\mathbf{a}}_l &= \tilde{\mathbf{C}} \tilde{\mathbf{a}}_l \quad \text{for } l = 1, 2, \dots, \hat{p}_F \\ \tilde{\mathbf{a}}_i^T \tilde{\mathbf{a}}_j &= \begin{cases} 1 & \text{if } i = j, \quad \text{for } i, j = 1, 2, \dots, \hat{p}_F, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

or equivalent to

$$\begin{aligned} \tilde{\lambda}_l \tilde{\mathbf{a}}_l &= \boldsymbol{\Psi}^T \boldsymbol{\Psi} \tilde{\mathbf{a}}_l \quad \text{for } l = 1, 2, \dots, \hat{p}_F \\ \tilde{\mathbf{a}}_i^T \tilde{\mathbf{a}}_j &= \begin{cases} 1 & \text{if } i = j, \quad \text{for } i, j = 1, 2, \dots, \hat{p}_F, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Note that $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ has p_F eigenvalues. Since the rank of $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ is equal to $\hat{p}_F$, then the remaining $(p_F - \hat{p}_F)$ eigenvalues of $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ are zero eigenvalues. Let $\tilde{\lambda}_h$, $(h = \hat{p}_F + 1, \hat{p}_F + 2, \dots, p_F)$, be the zero eigenvalues of $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ and $\tilde{\mathbf{a}}_h$ be the normalized eigenvectors of $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ corresponding to $\tilde{\lambda}_k$. Theorem (A.7) ensures that $\tilde{\mathbf{a}}_l$ $(l = 1, 2, \dots, \hat{p}_F)$ and $\tilde{\mathbf{a}}_h$ $(h = \hat{p}_F + 1, \hat{p}_F + 2, \dots, p_F)$ are orthogonal. Hence, we have

$$\begin{aligned} \tilde{\lambda}_h \tilde{\mathbf{a}}_h &= \boldsymbol{\Psi}^T \boldsymbol{\Psi} \tilde{\mathbf{a}}_h \quad \text{for } h = 1, 2, \dots, p_F \\ \tilde{\mathbf{a}}_i^T \tilde{\mathbf{a}}_j &= \begin{cases} 1 & \text{if } i = j, \quad \text{for } i, j = 1, 2, \dots, p_F, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The eigenvectors $\tilde{\mathbf{a}}_h$ $(h = 1, 2, \dots, p_F)$, however, cannot be found explicitly since we do not know $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ explicitly. However, we can obtain the principal component of $\psi(\mathbf{x})$ corresponding to nonzero eigenvalues of $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ by using the kernel method. The l-th principal component of $\psi(\mathbf{x})$, $l = 1, 2, \dots, \hat{p}_F$,

is given by

$$\begin{aligned} \tilde{y}_l(\mathbf{x}) &:= \psi(\mathbf{x})^T \tilde{\mathbf{a}}_l \\ &= \psi(\mathbf{x})^T \mathbf{\Psi}^T \boldsymbol{\alpha}_l \\ &= \sum_{i=1}^{N} \alpha_{li} \psi(\mathbf{x})^T \psi(\mathbf{x}_i) \quad \text{for } \mathbf{x} \in \mathbb{R}^p. \end{aligned} \tag{2.2.1}$$

Unfortunately, we do not know the term $\psi(\mathbf{x})^T \psi(\mathbf{x_i})$ explicitly yet. To overcome this limitation, we use the following theorem.

Theorem 2.2.2. (Mercer's Theorem) [26, 39] *For any symmetric, continuous and positive semidefinite kernel* $\xi : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$*, there exists a function* $\phi : \mathbb{R}^p \to \mathcal{F}$ *such that*

$$\xi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y}) \quad \textit{for } \mathbf{x}, \mathbf{y} \in \mathbb{R}^p.$$

By using Theorem 2.2.2, if we choose a continuous, symmetric and positive semidefinite kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ then there exists $\phi : \mathbb{R}^p \to \mathcal{F}$ such that $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. Instead of choosing ψ explicitly, we choose a kernel κ and employ the corresponding function ϕ as ψ. Let $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. Hence, we have

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

and it is explicitly known now.

Furthermore, Eq. (2.2.1) can be written as

$$\tilde{y}_l(\mathbf{x}) = \sum_{i=1}^{N} \alpha_{li} \kappa(\mathbf{x}, \mathbf{x}_i) \quad \text{for } \mathbf{x} \in \mathbb{R}^p. \tag{2.2.2}$$

and called it the *the l-th nonlinear principal components corresponding to* κ. The key difference between KPCA and PCA is in the extraction of principal components. For data $\tilde{\mathbf{X}}$, the linear PCA can find at most p principal components while the KPCA can find up to N principal components.

In summary, the following steps were necessary to compute the nonlinear principal components : (1) compute the matrix $\mathbf{K}$, (2) diagonalize matrix $\mathbf{K}$ and construct $\boldsymbol{\alpha}_l = \frac{\tilde{\mathbf{b}}_l}{\sqrt{\tilde{\lambda}_l}}$, (3) compute projection of a vector $\mathbf{x}$ onto the eigenvectors $\tilde{\mathbf{a}}_l$ which is given by Eq. (2.2.1).

Chapter 3

Principal Component Regression, Ridge Regression, Weighted Least Squares and M-Estimation

We introduce principal component regression (PCR) and ridge regression (RR) as linear methods for dealing with multicollinearity and collinearity. PCR and RR will be presented in Subchapter 3.1 and Subchapter 3.2, respectively. In Subchapter 3.3, we consider the regression model with variance of random errors having unequal values in diagonal elements. In this subchapter, we present weighted least-squares (WLS) which is a widely used technique to this regression. Afterward, we consider the regression model where the observed data are contaminated by outliers. In subchapter 3.4, we present M-estimation which is a widely used technique to eliminate the effects of outliers in the regression model.

3.1 Principal Component Regression

The *standard centered multiple linear regression model* corresponding to Eq. (1.1.2) is given by

$$\mathbf{Y}_o = \mathbf{Z}\boldsymbol{\beta}_{-0} + \boldsymbol{\epsilon}_o, \tag{3.1.1}$$

where $\mathbf{Z} = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\tilde{\mathbf{X}}$, $\boldsymbol{\epsilon}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{\epsilon}$, $\boldsymbol{\beta}_{-0} = (\beta_1 \quad \beta_2 \quad \dots \quad \beta_p)^T$, $\mathbf{Y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{Y}$ and β_j $(j = 1, 2, \dots, p)$ are defined in Subchapter 1.1. 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 symmetric and positive semidefinite, the eigenvalues of $\mathbf{Z}^T\mathbf{Z}$ are nonnegative real numbers. Let $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_r \geq \hat{\lambda}_{r+1} \geq \ldots \geq \hat{\lambda}_{\hat{p}} > \hat{\lambda}_{\hat{p}+1} = \ldots = \hat{\lambda}_p = 0$ be the eigenvalues of $\mathbf{Z}^T\mathbf{Z}$ and $\hat{\mathbf{A}} = (\hat{\mathbf{a}}_1 \quad \hat{\mathbf{a}}_2 \quad \dots \quad \hat{\mathbf{a}}_p)$ be the matrix of the corresponding normalized eigenvectors $\hat{\mathbf{a}}_l$ of $\mathbf{Z}^T\mathbf{Z}$. Then $\hat{\mathbf{A}}^T = \hat{\mathbf{A}}^{-1}$ and

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

where

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

$$\hat{\mathbf{D}}_{(\hat{p})} = \begin{pmatrix} \hat{\lambda}_1 & 0 & \dots & 0 \\ 0 & \hat{\lambda}_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \hat{\lambda}_{\hat{p}} \end{pmatrix},$$

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

Using $\hat{\mathbf{A}}\hat{\mathbf{A}}^T = \mathbf{I}_p$, we rewrite the model (3.1.1) as

$$\mathbf{Y}_o = \mathbf{U}\boldsymbol{\omega} + \boldsymbol{\epsilon}_o, \tag{3.1.2}$$

where $\mathbf{U} = \mathbf{Z}\hat{\mathbf{A}}$ and $\boldsymbol{\omega} = \hat{\mathbf{A}}^T\boldsymbol{\beta}_{-0}$. Let

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

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. The model (3.1.2) can now be written as

$$\mathbf{Y}_o = \mathbf{U}_{(\hat{p})}\boldsymbol{\omega}_{(\hat{p})} + \mathbf{U}_{(p-\hat{p})}\boldsymbol{\omega}_{(p-\hat{p})} + \boldsymbol{\epsilon}_o. \tag{3.1.3}$$

As we know that $\hat{\mathbf{D}} = \hat{\mathbf{A}}^T\mathbf{Z}^T\mathbf{Z}\hat{\mathbf{A}} = \mathbf{U}^T\mathbf{U}$, we obtain

$$\mathbf{U}_{(\hat{p})}^T\mathbf{U}_{(\hat{p})} = \hat{\mathbf{D}}_{(\hat{p})},$$

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

and

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

Since $(\mathbf{U}_{(p-\hat{p})}\boldsymbol{\omega}_{(p-\hat{p})})^T \mathbf{U}_{(p-\hat{p})}\boldsymbol{\omega}_{(p-\hat{p})} = 0$, we see that $\mathbf{U}_{(p-\hat{p})}\boldsymbol{\omega}_{(p-\hat{p})}$ is equal to $\mathbf{0}$. Thus, the model (3.1.3) reduces to

$$\mathbf{Y}_o = \mathbf{U}_{(\hat{p})}\boldsymbol{\omega}_{(\hat{p})} + \boldsymbol{\epsilon}_o. \tag{3.1.4}$$

Let us assume that $\lambda_{r+1}, \lambda_{r+2}, \ldots, \lambda_{\hat{p}}$ are close to zero. Let

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

and

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

where

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

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

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. The model (3.1.4) can now be written as

$$\mathbf{Y}_o = \mathbf{U}_{(r)}\boldsymbol{\omega}_{(r)} + \mathbf{U}_{(\hat{p}-r)}\boldsymbol{\omega}_{(\hat{p}-r)} + \boldsymbol{\epsilon}_o. \tag{3.1.5}$$

Since $\hat{\mathbf{D}}_{(\hat{p})} = \mathbf{U}_{(\hat{p})}^T \mathbf{U}_{(\hat{p})}$, we obtain that

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

and

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

It is evident that the estimator of $\boldsymbol{\omega}_{(\hat{p}-r)}$, say $\hat{\boldsymbol{\omega}}_{(\hat{p}-r)} = (\hat{\omega}_{r+1} \quad \hat{\omega}_{r+2} \quad \dots \quad \hat{\omega}_{\hat{p}})^T$, is given by

$$\hat{\boldsymbol{\omega}}_{(\hat{p}-r)} = (\mathbf{U}_{(\hat{p}-r)}^T\mathbf{U}_{(\hat{p}-r)})^{-1}\mathbf{U}_{(\hat{p}-r)}^T\mathbf{Y}_o = \mathbf{D}_{(\hat{p}-r)}^{-1}\mathbf{U}_{(\hat{p}-r)}^T\mathbf{Y}_o. \tag{3.1.6}$$

The terms $\mathbf{U}_{(\hat{p}-r)}^T\mathbf{Y}_o$ and $\mathbf{U}_{(\hat{p}-r)}^T\mathbf{Y}$ are related by the following lemma.

Lemma 3.1.1. $\mathbf{U}_{(\hat{p}-r)}^T\mathbf{Y} = \mathbf{U}_{(\hat{p}-r)}^T\mathbf{Y}_o$.

Proof. See Appendix B.2. □

By using Lemma 3.1.1, we obtain

$$\hat{\boldsymbol{\omega}}_{(\hat{p}-r)} = \hat{\mathbf{D}}_{(\hat{p}-r)}^{-1}\mathbf{U}_{(\hat{p}-r)}^T\mathbf{Y}, \tag{3.1.7}$$

and the variance of $\hat{\omega}_j$ is

$$Var(\hat{\omega}_j) = \sigma^2(\hat{\mathbf{D}}_{(\hat{p}-r)}^{-1})_{jj}, \qquad j = r+1, \dots, \hat{p}. \tag{3.1.8}$$

Since $\hat{\lambda}_{r+1}, \hat{\lambda}_{r+2}, \dots, \hat{\lambda}_{\hat{p}}$ are close to zero, the diagonal elements of $\hat{\mathbf{D}}_{(\hat{p}-r)}^{-1}$ and also the variance of $\hat{\omega}_j$ $(j = r+1, \dots, \hat{p})$ will be very large numbers. Thus, we encounter the ill effect of multicollinearity in the model (3.1.5). To avoid the effect of multicollinearity, we drop the term $\mathbf{U}_{(\hat{p}-r)}\boldsymbol{\omega}_{(\hat{p}-r)}$ as in [44] and obtain

$$\mathbf{Y}_o = \mathbf{U}_{(r)}\boldsymbol{\omega}_{(r)} + \boldsymbol{\acute{\epsilon}}_o, \tag{3.1.9}$$

where $\boldsymbol{\acute{\epsilon}}_o$ is a random vector influenced by dropping $\mathbf{U}_{(\hat{p}-r)}\boldsymbol{\omega}_{(\hat{p}-r)}$ in the model (3.1.5). The model (3.1.9) shows that the effects of collinearity and

multicollinearity on $\mathbf{Z}$ are avoided by using the orthogonal matrix $\hat{\mathbf{A}}$[1].

Note that $\mathbf{U}_{(r)}^T\mathbf{U}_{(r)} = \hat{\mathbf{D}}_{(r)}$, which is invertible. Hence, the estimator of $\boldsymbol{\omega}_{(r)}$, say $\hat{\boldsymbol{\omega}}_{(r)}$, is given as

$$\hat{\boldsymbol{\omega}}_{(r)} = (\mathbf{U}_{(r)}^T\mathbf{U}_{(r)})^{-1}\mathbf{U}_{(r)}^T\mathbf{Y}_o. \tag{3.1.10}$$

Let $\mathbf{y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y} \in \mathbb{R}^N$ be the observed data corresponding to $\mathbf{Y}_o$. Let $\hat{\boldsymbol{\omega}}_{(r)}^* \in \mathbb{R}^r$ be the value of $\hat{\boldsymbol{\omega}}_{(r)}$ when $\mathbf{Y}_o$ is replaced by $\mathbf{y}_o$ in the Eq. (3.1.10). By using the fact that $\mathbf{U}_{(r)}^T\mathbf{y} = \mathbf{U}_{(r)}^T\mathbf{y}_o$ (see Lemma 3.1.1), we obtain

$$\hat{\boldsymbol{\omega}}_{(r)}^* = (\mathbf{U}_{(r)}^T\mathbf{U}_{(r)})^{-1}\mathbf{U}_{(r)}^T\mathbf{y}. \tag{3.1.11}$$

Then the prediction value of $\mathbf{y}$, say $\check{\mathbf{y}}$, is given by

$$\check{\mathbf{y}} := \bar{y}\mathbf{1}_N + \mathbf{U}_{(r)}\hat{\boldsymbol{\omega}}_{(r)}^*. \tag{3.1.12}$$

where $\bar{y} = \frac{1}{N}\mathbf{1}_N^T\mathbf{y}$. Since

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

we obtain $\mathbf{U}_{(r)} = \mathbf{Z}\hat{\mathbf{A}}_{(r)}$. The Eq. (3.1.12) can now be written as

$$\check{\mathbf{y}} = \bar{y}\mathbf{1}_N + \mathbf{Z}\hat{\mathbf{A}}_{(r)}\hat{\boldsymbol{\omega}}_{(r)}^*. \tag{3.1.13}$$

The *prediction by PCR model* is given by

$$f_{pcr}(\mathbf{z}) := \bar{y} + \mathbf{z}^T\hat{\mathbf{A}}_{(r)}\hat{\boldsymbol{\omega}}_{(r)}^*, \tag{3.1.14}$$

where f_{pcr} is a function from $\mathbb{R}^p$ to $\mathbb{R}$, $\mathbf{z} = \begin{pmatrix}x_1 - \bar{x}_1 & x_2 - \bar{x}_2 & \dots & x_p - \bar{x}_p\end{pmatrix}^T$

[1]To detect multicollinearity (collinearity) on $\mathbf{Z}$, we use the ratio λ_l/λ_1 for $l = 1, 2, \dots, p$. If λ_l/λ_1 is smaller than, say $< \frac{1}{1000}$, then we consider that multicollinearity (collinearity) exists on $\mathbf{Z}$ [27].

and $\bar{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$ $(j = 1, 2, \ldots, p)$.

3.2 Ridge Regression

As mentioned before, we can use ridge regression to avoid the effects of multicollinearity and collinearity. Let us consider the standard ordinary multiple linear regression model (1.1.2) again. The ridge estimator of $\boldsymbol{\beta}$ is found by solving the following problem

$$\min \quad (\mathbf{Y} - \boldsymbol{\beta})^T(\mathbf{Y} - \boldsymbol{\beta}) + c\boldsymbol{\beta}^T\boldsymbol{\beta} \tag{3.2.1}$$

for some $c > 0$. Let $\hat{\boldsymbol{\beta}}_R(c)$ be the solution of problem (3.2.1). Then, we have

$$\boldsymbol{\beta}_R(c) = (\mathbf{X}^T\mathbf{X} + c\mathbf{I}_{p+1})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{3.2.2}$$

$\hat{\boldsymbol{\beta}}_R(c)$ is called the ridge estimator of $\boldsymbol{\beta}$.

Furthermore, let $\hat{\boldsymbol{\beta}}^*{}_R(c) = \begin{pmatrix} \hat{\beta}^*_{R0}(c) & \hat{\beta}^*_{R1}(c) & \ldots & \hat{\beta}^*_{Rp}(c) \end{pmatrix}^T \in \mathbb{R}^{p+1}$ be the value of $\hat{\boldsymbol{\beta}}$ when $\mathbf{Y}$ is replaced by $\mathbf{y}$ in the Eq. (3.2.2). Hence,

$$\hat{\boldsymbol{\beta}}^*{}_R(c) = (\mathbf{X}^T\mathbf{X} + c\mathbf{I}_{p+1})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.2.3}$$

Then, the *prediction by RR* is given by

$$f_R(\mathbf{x}) := \hat{\beta}^*_{R0}(c) + \sum_{j=1}^{p} \hat{\beta}^*_{Rj}(c)x_j, \tag{3.2.4}$$

where f_R is a function from $\mathbb{R}^p$ to $\mathbb{R}$. The appropriate c of the prediction by RR can be found by the cross validation (CV) method[2] or other methods.

[2]The detailed CV method is given in Subchapter 4.1

3.3 Weighted Least Squares

In some cases, we face the regression model with variance of random errors having unequal values in diagonal elements. The model is given by the following regression model

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \\ E(\boldsymbol{\epsilon}) &= \mathbf{0}, \\ Var(\boldsymbol{\epsilon}) &= \sigma_1^2 \grave{\mathbf{V}}, \end{aligned} \tag{3.3.1}$$

where $\grave{\mathbf{V}} = \text{diag}(1/w_1, 1/w_2, \ldots, 1/w_N)$ and w_i is a positive number for $i = 1, 2, \ldots, N$. The weight w_i is estimated by using the data $\mathbf{y}$ and $\mathbf{X}$, see for example [14, 30, 27]. An implication of the assumption $Var(\boldsymbol{\epsilon}) = \sigma^2 \grave{\mathbf{V}}$ are the OLS estimator and hypothesis testing based on the OLS estimator of the variance matrix become invalid [14, 27, 30, 41, 44]. This limitation is avoided by transforming the model (3.3.1) to a new model that satisfies the assumption of the OLR model. This technique is known as the *weighted least-squares* (*WLS*) in linear regression.

Let $\mathbf{L} = \text{diag}(1/\sqrt{w_1}, 1/\sqrt{w_2}, \ldots, 1/\sqrt{w_N})$. It is evident that $\mathbf{L}^T = \mathbf{L}$, $\mathbf{L}\mathbf{L}^T = \grave{\mathbf{V}}$ and $\mathbf{L}^{-1} = \text{diag}(\sqrt{w_1}, \sqrt{w_2}, \ldots, \sqrt{w_N})$. The above difficulties can be avoided by multiplying the model with $\mathbf{L}^{-1}$. Then, we have

$$\mathbf{L}^{-1}\mathbf{Y} = \mathbf{L}^{-1}\mathbf{X}\boldsymbol{\beta} + \mathbf{L}^{-1}\boldsymbol{\epsilon}. \tag{3.3.2}$$

Let $\mathbf{Y_1} = \mathbf{L}^{-1}\mathbf{Y}$, $\mathbf{X_1} = \mathbf{L}^{-1}\mathbf{X}$ and $\boldsymbol{\epsilon_1} = \mathbf{L}^{-1}\boldsymbol{\epsilon}$. It is easy to verify that $E(\boldsymbol{\epsilon_1}) = \mathbf{0}$ and $Var(\boldsymbol{\epsilon_1}) = \sigma^2 \mathbf{I}_N$. Hence, model (3.3.1) becomes

$$\begin{aligned} \mathbf{Y_1} &= \mathbf{X_1}\boldsymbol{\beta} + \boldsymbol{\epsilon_1}, \\ E(\boldsymbol{\epsilon_1}) &= \mathbf{0}, \\ Var(\boldsymbol{\epsilon_1}) &= \sigma^2 \mathbf{I}_N. \end{aligned} \tag{3.3.3}$$

It is evident that the error $\boldsymbol{\epsilon_1}$ in the model (3.3.3) satisfies the ordinary linear

model assumption. Then, the least-squares function is

$$\begin{aligned} \mathcal{S}(\boldsymbol{\beta}) &= \boldsymbol{\epsilon_1}^T \boldsymbol{\epsilon_1}, \\ &= (\mathbf{Y_1} - \mathbf{X_1}\boldsymbol{\beta})^T (\mathbf{Y_1} - \mathbf{X_1}\boldsymbol{\beta}), \\ &= (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T \grave{\mathbf{V}}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}). \end{aligned} \tag{3.3.4}$$

To obtain the estimator of $\boldsymbol{\beta}$ in model (3.3.3), we solve

$$\min \quad (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T \grave{\mathbf{V}}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}). \tag{3.3.5}$$

with respect to $\boldsymbol{\beta}$. Let $\hat{\boldsymbol{\beta}}_\mathbf{1}$ be the solution of the problem (3.3.5). Hence, $\hat{\boldsymbol{\beta}}_\mathbf{1}$ satisfies the least-squares normal equations

$$(\mathbf{X}^T \grave{\mathbf{V}}^{-1} \mathbf{X}) \hat{\boldsymbol{\beta}}_\mathbf{1} = \mathbf{X}^T \grave{\mathbf{V}}^{-1} \mathbf{Y}. \tag{3.3.6}$$

It is evident that if the row vectors of $\mathbf{X}$ are linearly independent, then the row vectors of $\mathbf{X_1}$ are also linearly independent. Hence, $\mathbf{X_1}^T \mathbf{X_1} = \mathbf{X}^T \grave{\mathbf{V}}^{-1} \mathbf{X}$ is invertible and we obtain

$$\hat{\boldsymbol{\beta}}_\mathbf{1} = (\mathbf{X}^T \grave{\mathbf{V}}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \grave{\mathbf{V}}^{-1} \mathbf{Y}. \tag{3.3.7}$$

Here, $\hat{\boldsymbol{\beta}}_\mathbf{1}$ is called the *WLS estimator* of $\boldsymbol{\beta}$. The covariance matrix of $\hat{\boldsymbol{\beta}}_\mathbf{1}$ is

$$Var(\hat{\boldsymbol{\beta}}_\mathbf{1}) = \sigma^2 (\mathbf{X}^T \grave{\mathbf{V}}^{-1} \mathbf{X})^{-1}. \tag{3.3.8}$$

Note that, elements of $\mathbf{X}$ can be chosen such that multicollinearity does not exist in $\mathbf{X}$. Unfortunately, eigenvalues of $\mathbf{X}^T\mathbf{X}$ are not equal to eigenvalues of $\mathbf{X_1}^T\mathbf{X_1}$. Hence, there is no guarantee that multicollinearity does not exist in $\mathbf{X_1}$.

Let $\hat{\boldsymbol{\beta}}^*_\mathbf{1} = \begin{pmatrix} \hat{\beta}^*_{\mathbf{10}} & \hat{\beta}^*_{\mathbf{11}} & \dots & \hat{\beta}^*_{\mathbf{1}p} \end{pmatrix}^T \in \mathbb{R}^{p+1}$ be the value of $\hat{\boldsymbol{\beta}}_\mathbf{1}$ when $\mathbf{Y}$ is replaced by $\mathbf{y}$ in the Eq. (3.3.7). The prediction value of $\mathbf{y_1}(= \mathbf{L}^{-1}\mathbf{y})$, say

$\hat{\mathbf{y}}_\mathbf{1}$, is given by

$$\hat{\mathbf{y}}_\mathbf{1} := \begin{pmatrix} \hat{y}_{11} & \hat{y}_{12} & \dots & \hat{y}_{1N} \end{pmatrix}^T = \mathbf{X}_1 \hat{\boldsymbol{\beta}}^*_\mathbf{1}, \tag{3.3.9}$$

and the residual between $\mathbf{y}_\mathbf{1}$ and $\hat{\mathbf{y}}_\mathbf{1}$ is given by

$$\mathbf{e}_\mathbf{1} := \begin{pmatrix} e_{11} & e_{12} & \dots & e_{1N} \end{pmatrix}^T = \mathbf{y}_\mathbf{1} - \hat{\mathbf{y}}_\mathbf{1}. \tag{3.3.10}$$

The RMSE for the WLS regression model is given by

$$RMSE_{wls} := \sqrt{\frac{\mathbf{e}_\mathbf{1}{}^T \mathbf{e}_\mathbf{1}}{N}} \tag{3.3.11}$$

and the *prediction by the WLS-LR* is given by

$$f_{wls}(\mathbf{x}) := \hat{\beta}^*_{\mathbf{10}} + \sum_{j=1}^{p} \hat{\beta}^*_{\mathbf{1}j} x_j, \tag{3.3.12}$$

where f_{wls} is a function from $\mathbb{R}^p$ to $\mathbb{R}$.

As we see that WLS-LR yields a prediction in the linear form. Since the most of real problems are nonlinear, the model has limitations on applications. Beside that, there is no guarantee that multicollinearity does not exist in $\mathbf{L}^{-1}\mathbf{X}$. Although we can use KPCR or KRR to overcome the limitations of the linearity and multicollinearity, these methods can be inappropriate in this regression model. Since these methods were constructed by the assumption that the variance of random errors having equal values in diagonal elements.

3.4 M-Estimation

Let us consider again model (1.1.2) again

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{3.4.1}$$

Note that $\mathbf{y}$ is the observed data corresponding to $\mathbf{Y}$. Hence, we have

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{e}, \tag{3.4.2}$$

where $\boldsymbol{e} \in \mathbb{R}^N$ is a vector of residuals. Note that the aim of regression analysis is to find the estimator of $\boldsymbol{\beta}$, say $\hat{\boldsymbol{\beta}}$, such that the *least-squares function*,

$$\begin{aligned} \mathcal{S}(\boldsymbol{\beta}) &= \boldsymbol{e}^T\boldsymbol{e} \\ &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \end{aligned} \tag{3.4.3}$$

is minimized and the procedure to obtain the estimator of $\boldsymbol{\beta}$ by solving Eq. (3.4.3) is called the OLS method.

The main disadvantage of the OLS method is its sensitivity to outliers, i.e., residuals of the observed data are large numbers. Outliers have a large influence on the prediction value because squaring residuals magnifies the effect of the outliers. If the outliers are contained in the observed data, the predictions of OLR, PCR, RR, KPCR and KRR become inappropriate to be used since those methods were constructed based on OLS method.

Andrews, Carol and Ruppert; Hogg, Hubber, Krasker and Welsch; and Rousseeuw and Leroy proposed *robust regression* methods to eliminate the influence of the outliers [27]. M-estimation is one of the most widely used methods for the robust regression but the method yields a linear prediction. We notice that Fomengko [13] proposed a nonlinear robust prediction based on M-estimation with specifying a nonlinear regression model in advance. In many situations, however, a specific nonlinear regression model for a set of data is unknown in advance. Hence, the proposed method has limitations in applications.

M-estimation method can be considered as a modification of both regression based on OLS and maximum likelihood estimation that eliminate the effects of outlying observation on the regression estimation. Note that, Eq. (3.4.3) can be written as

$$\sum_{i=1}^{N} e_i^2, \tag{3.4.4}$$

where $e_i = y_i - \acute{\mathbf{x}}_i^T \boldsymbol{\beta}$ and $\acute{\mathbf{x}}_i^T = \begin{pmatrix} 1 & \mathbf{x}_i^T \end{pmatrix}$. In the M-estimation method, the term e_i^2 is replaced by $\rho(e_i)$ where ρ is a function from $\mathbb{R}$ to $\mathbb{R}$. Hence, we must find the estimator of $\boldsymbol{\beta}$ such that the function

$$\sum_{i=1}^{N} \rho(e_i) = \sum_{i=1}^{N} \rho(y_i - \acute{\mathbf{x}}_i^T \boldsymbol{\beta}), \tag{3.4.5}$$

is minimized. Consequently, RMSE of linear robust regression based on M-estimation is calculated by using $\rho(e_i)$. The function ρ should be symmetric ($\rho(e_i) = \rho(-e_i)$), positive ($\rho(e_i) \geq 0$), strictly monotonically increasing ($\rho(|e_{i1}|) > \rho(|e_{i2}|)$ if $|e_{i1}| > |e_{i2}|$), and convex on $\mathbb{R}$. The most common choice of ρ is the Huber function [3]

$$\rho(z) = \begin{cases} 1/2z^2 & |z| \leq k, \\ k|z| - 1/2k^2 & |z| > k, \end{cases} \tag{3.4.6}$$

where $k \in \mathbb{R}$. Another choice of ρ is the Tukey biweighted function

$$\rho(z) = \begin{cases} 1/6[(1 - (1 - z^2)^3] & |z| \leq 1, \\ 1/6 & |z| > 1. \end{cases} \tag{3.4.7}$$

To minimize Eq. (3.4.5), equate the first partial derivatives of ρ with respect to β_j ($j = 0, 1, \ldots, p$) to zero. This gives the system of $p+1$ equations

$$\sum_{i=1}^{N} \rho'(e_i) \acute{\mathbf{x}}_i^T = \mathbf{0}^T, \tag{3.4.8}$$

where ρ' is the derivative of ρ. Then, we define the weight function

$$w(z) = \begin{cases} \rho'(z)/z & \text{if } z \neq 0, \\ 1 & \text{if } z = 0, \end{cases} \tag{3.4.9}$$

Then, Eq. (3.4.8) can be written as

$$\sum_{i=1}^{N}(\frac{\rho'(e_i)}{e_i})e_i\acute{\mathbf{x}}_i^T = \sum_{i=1}^{N} w_i e_i \acute{\mathbf{x}}_i^T = \mathbf{0}^T, \tag{3.4.10}$$

where $w_i = w(e_i)$. Since $e_i = y_i - \acute{\mathbf{x}}_i^T\boldsymbol{\beta}$, we obtain

$$\sum_{i=1}^{N} w_i y_i \acute{\mathbf{x}}_i^T = \sum_{i=1}^{N} w_i \acute{\mathbf{x}}_i^T \boldsymbol{\beta} \acute{\mathbf{x}}_i^T. \tag{3.4.11}$$

In matrix form, Eq. (3.4.11) becomes

$$\mathbf{X}^T\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{W}\mathbf{y}, \tag{3.4.12}$$

where $\mathbf{W} = \text{diag}(w_1, w_2, \ldots, w_N)$ and called the *weigthed least squares equations*. Let $\hat{\boldsymbol{\beta}}_M^*$ be the solution of Eq. (3.4.12). Hence, we have

$$\mathbf{X}^T\mathbf{W}\mathbf{X}\hat{\boldsymbol{\beta}}_M^* = \mathbf{X}^T\mathbf{W}\mathbf{y}. \tag{3.4.13}$$

The estimator $\hat{\boldsymbol{\beta}}_M^*$ is called the *robust estimator of* $\boldsymbol{\beta}$. The weights, however, depend upon the residuals, the residuals depend upon the estimated regression coefficients and the the estimated regression coefficients depends upon the weights. An iterative solution, called *iteratively reweighted least-squares* (*IRLS*), is therefore required. The IRLS algorithm is given in the following steps:

1. Select the initial estimator of $\boldsymbol{\beta}$, say $\hat{\boldsymbol{\beta}}_{\boldsymbol{M}}^{*(\mathbf{0})}$, by OLS.
2. At each iteration t, calculate residual $e_i^{(t-1)} = y_i - \acute{\mathbf{x}}_i^T\boldsymbol{\beta}_M^{*(t-1)}$, $w_i^{(t-1)} = w(e_i^{(t-1)})$ and $\mathbf{W}^{(t-1)} = \text{diag}(w_1^{(t-1)}, w_2^{(t-1)}, \ldots, w_N^{(t-1)})$.
3. Solve the new weighted least squares equations

$$\mathbf{X}^T\mathbf{W}^{(t-1)}\mathbf{X}\hat{\boldsymbol{\beta}}_M^{*(t)} = \mathbf{X}^T\mathbf{W}^{(t-1)}\mathbf{y}.$$

Step 2 and Step 3 are repeated until the estimated regression coefficients converges.

In general, the convergence of IRLS algorithm is not quarantee. However, IRLS works well in practice [7, 19] and is frequently used in the computational statistic community [12, 32]. By adding some other assumptions, the convergence proof of IRLS can be found in [54].

Chapter 4

Nonlinear Regressions Based on Kernel Principal Component Analysis

In this chapter, we consider some nonlinear methods based on kernel principal component analysis (KPCA). In Subchapter 4.1, we present kernel principal component regression (KPCR) which was constructed to overcome the limitations of PCR. We show that the previous works of KPCR have theoretical difficulty in the procedure to derive the prediction by the KPCR. The revised method of the KPCR will also be presented in this subchapter. In Subchapter 4.2, we propose a combination of WLS and KPCR to overcome the limitation of WLS method. Then, a combination of M-Estimation and KPCR as a nonlinear method for dealing with outliers is given in Subchapter 4.3.

4.1 Kernel Principal Component Regression

4.1.1 The Previous Works

As mentioned before that PCR can be used eliminate the effect of multicollinearity and collinearity. However, PCR still yield predictions in the linear forms. Since the most of real problems are nonlinear, PCR has limi-

tations on applications. Rosipal *et al.* [33, 34, 35], Hoegaerts *et al.* [18] and Jade *et al.* [20] used the KPCR to overcome the limitations. We refer their KPCR as the *previous KPCR*.

They transformed $\mathbf{x}_i$ $(i = 1, 2, \ldots, N)$ into $\mathcal{F}$ by using a function $\tilde{\psi}$: $\mathbb{R}^p \rightarrow \mathcal{F}$. Note that the function $\tilde{\psi}$ is not explicitly known. Then, they constructed two matrices

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

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

where sizes of $\tilde{\boldsymbol{\Psi}}$ and $\tilde{\mathbf{K}}$ are $N \times p_F$ and $N \times N$, respectively. Roman *et al.* [33, 34, 35], Hoegaerts *et al.* [18] and Jade *et al.* [20] defined the *standard multiple linear regression model in the feature space* as the following model

$$\mathbf{Y} = \tilde{\boldsymbol{\Psi}}\boldsymbol{\eta} + \boldsymbol{\acute{\epsilon}}, \tag{4.1.1}$$

where $\boldsymbol{\eta} = \begin{pmatrix} \eta_1 & \eta_2 & \ldots & \eta_{p_F} \end{pmatrix}^T$ is a vector of regression coefficients in the feature space and $\boldsymbol{\acute{\epsilon}}$ is a vector of random errors in the feature space which is assumed that $\sum_{i=1}^{N} \psi(\mathbf{x}_i) = \mathbf{0}$, $E(\boldsymbol{\acute{\epsilon}}) = \mathbf{0}$, $E(\boldsymbol{\acute{\epsilon}}\boldsymbol{\acute{\epsilon}}^T) = \delta^2\mathbf{I}_N$ where $\delta^2 \in \mathbb{R}$. They denoted that $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_{\hat{r}} \geq \mu_{\hat{r}+1} \geq \ldots \geq \mu_{p_F}$ be the eigenvalues of $\tilde{\boldsymbol{\Psi}}^T\tilde{\boldsymbol{\Psi}}$, $\boldsymbol{\varrho}_k = (\varrho_{k1} \quad \varrho_{k2} \quad \ldots \quad \varrho_{kN})^T$ be the normalized eigenvector of $\tilde{\mathbf{K}}$ corresponding to μ_k and $\mathbf{V} = (\mathbf{v}_1 \quad \mathbf{v}_2 \quad \ldots \quad \mathbf{v}_{p_F})$ be the matrix of the corresponding normalized eigenvectors $\mathbf{v}_k$ of $\tilde{\boldsymbol{\Psi}}^T\tilde{\boldsymbol{\Psi}}$ where $\mathbf{v}_k = \tilde{\boldsymbol{\Psi}}^T\frac{\boldsymbol{\varrho}_k}{\sqrt{\mu_k}}$ $(k = 1, 2, ...p_F)$. It is evident that $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{p_F}$. They rewrote the model (4.1.1) as

$$\mathbf{Y} = \mathbf{B}\boldsymbol{\varpi} + \boldsymbol{\acute{\epsilon}}, \tag{4.1.2}$$

where $\mathbf{B} = \tilde{\boldsymbol{\Psi}}\mathbf{V}$ and $\boldsymbol{\varpi} = (\varpi_1 \quad \varpi_2 \quad \ldots \quad \varpi_{p_F})^T = \mathbf{V}^T\boldsymbol{\eta}$. As we see that the element $\tilde{\psi}(\mathbf{x}_i)^T\mathbf{v}_k$ of $\mathbf{B}$ is the k-th principal component of $\tilde{\psi}(\mathbf{x}_i)$ for $k = 1, 2, ..., p_F$. By choosing a kernel function κ and applying Theorem 2.2.2, they obtained that the element $\tilde{\psi}(\mathbf{x}_i)^T\mathbf{v}_k$ is equal to $\sum_{j=1}^{N}\frac{\varrho_{kj}}{\sqrt{\mu_k}}\kappa(\mathbf{x}_i, \mathbf{x}_j)$ $(i = 1, 2, \ldots, N)$. They stated the estimator of $\boldsymbol{\varpi}$, say $\hat{\boldsymbol{\varpi}} = (\hat{\varpi}_1 \quad \hat{\varpi}_2 \quad \ldots \quad \hat{\varpi}_{p_F})^T$,

is given by

$$\hat{\boldsymbol{\varpi}} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{Y} \tag{4.1.3}$$

$$= \boldsymbol{\Lambda}^{-1}\mathbf{B}^T\mathbf{Y}, \tag{4.1.4}$$

where

$$\boldsymbol{\Lambda}^{-1} = \begin{pmatrix} \frac{1}{\tilde{\mu}_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\tilde{\mu}_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{\tilde{\mu}_{p_F}} \end{pmatrix}.$$

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

$$\hat{\boldsymbol{\eta}} = \mathbf{V}\hat{\boldsymbol{\varpi}} = \sum_{i=1}^{p_F} \mu_i^{-1}\mathbf{v}_i\mathbf{v}_i^T\boldsymbol{\Psi}^T\mathbf{Y}, \tag{4.1.5}$$

and its corresponding covariance matrix as

$$cov(\hat{\boldsymbol{\eta}}) = \delta^2 \sum_{i=1}^{p_F} \mu_i^{-1}\mathbf{v}_i\mathbf{v}_i^T. \tag{4.1.6}$$

It is evident from (4.1.6) that the influence of small eigenvalues can significantly increase the overall variance of the estimator of $\boldsymbol{\eta}$. To avoid the effect of multicollinearity, PCR deletes some eigenvectors of $\tilde{\boldsymbol{\Psi}}^T\tilde{\boldsymbol{\Psi}}$ corresponding to small eigenvalues μ_i. Let $\hat{\boldsymbol{\varpi}}^* = (\hat{\varpi}_1^* \quad \hat{\varpi}_2^* \quad \dots \quad \hat{\varpi}_{p_F}^*)^T \in \mathbb{R}^{p_F}$ be $\hat{\boldsymbol{\varpi}}$ when $\mathbf{Y}$ is replaced by $\mathbf{y}$ in the Eq. (4.1.3). Using the first $\hat{r}$ vectors of $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{p_F}$, they stated that the *prediction by the previous KPCR model* is written as

$$g_{p-kpcr}(\mathbf{x}) = \sum_{i=1}^{N} a_i\kappa(\mathbf{x}_i, \mathbf{x}) + d, \tag{4.1.7}$$

where g is a function from $\mathbb{R}^p$ to $\mathbb{R}$, $a_i = \sum_{k=1}^{\hat{r}} \hat{\varpi}_k^* \frac{\varrho_{ik}}{\sqrt{\mu_k}}$ for $i = 1, \dots, N$, and d is a bias term. The term d will vanish when $\sum_{i=1}^{N} y_i = 0$ as shown in [18, 20, 33, 34, 35], where y_i is the ith element of $\mathbf{y}$. The number $\hat{r}$ is

called the *retained number of nonlinear principal components (PCs) for the previous KPCR model*.

Note that the size of $\mathbf{B}$ $(= \tilde{\mathbf{\Psi}}\mathbf{V})$ is $N \times p_F$. We want to show the theoretical difficulty of the previous KPCR. Let us consider in the cases of $N < p_F$ and $N \geq p_F$. In the case of $N < p_F$, the column vectors of $\mathbf{B}$ are linearly dependent. It is well known that $\text{rank}(\mathbf{B}^T\mathbf{B}) = \text{rank}(\tilde{\mathbf{\Psi}}^T\tilde{\mathbf{\Psi}})$. Since column vectors of $\mathbf{B}$ are linearly dependent, $\text{rank}(\mathbf{B}^T\mathbf{B}) < p_F$, and hence some eigenvalues of $\tilde{\mathbf{\Psi}}^T\tilde{\mathbf{\Psi}}$ are equal to zero. As a result, Eq. (4.1.3)-(4.1.6) become undefined. Consequently, $g_{p-kpcr}(\mathbf{x})$ is undefined either. As we see that we do not know $\tilde{\psi}(\mathbf{x}_i)^T\mathbf{v}_{\hat{k}}$ explicitly when $\mu_{\hat{k}}$ is equal to zero for $\hat{k} \in \{1, 2, ..., p_F\}$ since $\mathbf{v}_{\hat{k}}$ is not known explicitly either[1]. It implies that some elements of $\mathbf{B}$ are not known explicitly. Hence, we cannot use the generalized inverse of $\mathbf{B}^T\mathbf{B}$ to obtain the estimator of $\boldsymbol{\varpi}$[2].

In the case of $N \geq p_F$, the column vectors of $\mathbf{B}$ can be linearly dependent, which offers the difficulty that we have explained as above. In OLR model, we can construct $\mathbf{X}$ so that it consists of linearly independent columns. In the KPCR, however, the matrix $\mathbf{B}$ is defined by the function $\tilde{\psi}$ and $\tilde{\psi}$ is provided implicitly by a kernel function κ. It is a difficult task to choose the kernel function κ in order to make the column vectors of $\mathbf{B}$ be linearly independent. Thus, the procedure to derive the KPCR suffers theoretical difficulties.

We also note that they [34] used the CV technique for model selection in the KPCR. In the CV technique, the original data are partitioned into L disjoint subsets data where L is a positive integer. A subset data, say G_k $(k = 1, 2, \ldots, L)$, is chosen as the validation for testing the prediction model and the remaining $L - 1$ subsets data are used to estimate the regression coefficients $\boldsymbol{\varpi}$. The CV technique uses the *prediction error sum of squares*

[1]KPCA can find up to N principal components corresponding to nonzero eigenvalues.

[2]$\hat{\boldsymbol{\varpi}} = (\mathbf{B}^T\mathbf{B})^-\mathbf{B}^T\mathbf{Y}$ where $(\mathbf{B}^T\mathbf{B})^-$ is a generalized inverse of $\mathbf{B}^T\mathbf{B}$. $(\mathbf{B}^T\mathbf{B})^-$ is said to be a generalized inverse of $\mathbf{B}^T\mathbf{B}$ if $\mathbf{B}^T\mathbf{B}(\mathbf{B}^T\mathbf{B})^-\mathbf{B}^T\mathbf{B} = \mathbf{B}^T\mathbf{B}$.

(PRESS) to obtain the appropriate $\hat{r}$. The PRESS of G_k is given by

$$PRESS(G_k) = \sum_{s=1}^{m_k} (y_s^k - g_{p-kpcr}(\mathbf{x}_s^k))^2, \tag{4.1.8}$$

where $\mathbf{x}_s^k$ and y_s^k are contained in G_k and m_k is the cardinality of G_k. Then, PRESS(G_k) is summed over all the subsets data. As we see, the PRESS(G_k) becomes undefined when $g_{p-kpcr}(\mathbf{x})$ is undefined or the PRESS(G_k) is difficult to define when we have difficulty to obtain $g_{p-kpcr}(\mathbf{x})$. Hence, the procedure based on CV to obtain the appropriate $\hat{r}$ also suffers theoretical difficulty.

4.1.2 The Revised of KPCR

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

$$\mathbf{Y}_o = \mathbf{\Psi}\boldsymbol{\gamma} + \tilde{\boldsymbol{\epsilon}}, \tag{4.1.9}$$

where $\boldsymbol{\gamma} = \begin{pmatrix} \gamma_1 & \gamma_2 & \dots & \gamma_{p_F} \end{pmatrix}^T$ is a vector of regression coefficients in the feature space, $\tilde{\boldsymbol{\epsilon}}$ is a vector of random errors in the feature space and $\mathbf{Y}_o$ is defined in Subchapter 3.1. We assume $E(\tilde{\boldsymbol{\epsilon}}) = \mathbf{0}$, $E(\tilde{\boldsymbol{\epsilon}}\tilde{\boldsymbol{\epsilon}}^T) = \tilde{\sigma}^2\mathbf{I}_N$ where $\tilde{\sigma}^2 \in \mathbb{R}$. Here, we cannot use the generalized inverse matrix to obtain the estimator of $\boldsymbol{\gamma}$ since $\mathbf{\Psi}$ is not known explicitly. We refer to our proposed KPCR as the *revised KPCR*.

As mentioned in Subchapter 2.2 that $\mathbf{K}$ is explicitly known by choosing a kernel function κ and the rank of $\mathbf{K}$ is $\hat{p}_F$ where $\hat{p}_F \leq \min(N, p_F)$. The eigenvalues of $\mathbf{K}$ are $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_{\tilde{r}} \geq \tilde{\lambda}_{\tilde{r}+1} \geq \ldots \geq \tilde{\lambda}_{\hat{p}_F} > \tilde{\lambda}_{\hat{p}_F+1} = \ldots = \tilde{\lambda}_N = 0$, $\tilde{\mathbf{B}} = (\tilde{\mathbf{b}}_1 \quad \tilde{\mathbf{b}}_2 \quad \ldots \quad \tilde{\mathbf{b}}_N)$ are the matrix of the corresponding normalized eigenvectors $\tilde{\mathbf{b}}_l$ of $\mathbf{K}$, $\boldsymbol{\alpha}_l = \begin{pmatrix} \alpha_{l1} & \alpha_{l2} & \dots & \alpha_{lN} \end{pmatrix}^T = \frac{\tilde{\mathbf{b}}_l}{\sqrt{\tilde{\lambda}_l}}$ and

$\tilde{\mathbf{a}}_l = \boldsymbol{\Psi}^T \boldsymbol{\alpha}_l$ for $l = 1, 2, \ldots, \hat{p_F}$. Then, we have

$$\begin{aligned} \tilde{\lambda}_h \tilde{\mathbf{a}}_h &= \boldsymbol{\Psi}^T \boldsymbol{\Psi} \tilde{\mathbf{a}}_h \qquad \text{for } h = 1, 2, \ldots, p_F \\ \tilde{\mathbf{a}}_i^T \tilde{\mathbf{a}}_j &= \begin{cases} 1 & \text{if } i = j, \qquad \text{for } i, j = 1, 2, \ldots, p_F, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Furthermore, we define $\tilde{\mathbf{A}} = (\tilde{\mathbf{a}}_1 \quad \tilde{\mathbf{a}}_2 \quad \ldots \quad \tilde{\mathbf{a}}_{p_F})$. It is evident that $\tilde{\mathbf{A}}$ is an orthogonal matrix, that is, $\tilde{\mathbf{A}}^T = \tilde{\mathbf{A}}^{-1}$. It is not difficult to verify that

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

where

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

By using $\tilde{\mathbf{A}} \tilde{\mathbf{A}}^T = \mathbf{I}_{p_F}$, we rewrite the model (4.1.9) as

$$\mathbf{Y}_o = \tilde{\mathbf{U}} \boldsymbol{\vartheta} + \tilde{\boldsymbol{\epsilon}}, \tag{4.1.10}$$

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

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

where sizes of $\tilde{\mathbf{U}}_{(\hat{p}_F)}$, $\tilde{\mathbf{U}}_{(p_F - \hat{p}_F)}$, $\boldsymbol{\vartheta}_{(\hat{p}_F)}$, and $\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)}$ 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. The model (4.1.10) can be written as

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\hat{p}_F)} \boldsymbol{\vartheta}_{(\hat{p}_F)} + \tilde{\mathbf{U}}_{(p_F - \hat{p}_F)} \boldsymbol{\vartheta}_{(p_F - \hat{p}_F)} + \tilde{\boldsymbol{\epsilon}}. \tag{4.1.11}$$

As we see that $\tilde{\mathbf{D}} = \tilde{\mathbf{A}}^T \boldsymbol{\Psi}^T \boldsymbol{\Psi} \tilde{\mathbf{A}} = \tilde{\mathbf{U}}^T \tilde{\mathbf{U}}$, and we obtain

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

and

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

Since $(\tilde{\mathbf{U}}_{(p_F - \hat{p}_F)} \boldsymbol{\vartheta}_{(p_F - \hat{p}_F)})^T \tilde{\mathbf{U}}_{(p_F - \hat{p}_F)} \boldsymbol{\vartheta}_{(p_F - \hat{p}_F)} = 0$, we see that $\tilde{\mathbf{U}}_{(p_F - \hat{p}_F)} \boldsymbol{\vartheta}_{(p_F - \hat{p}_F)}$ is equal to $\mathbf{0}$. Consequently, the model (4.1.11) reduces to

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\hat{p}_F)} \boldsymbol{\vartheta}_{(\hat{p}_F)} + \tilde{\boldsymbol{\epsilon}}. \tag{4.1.12}$$

Let us assume that $\tilde{\lambda}_{\tilde{r}+1}, \tilde{\lambda}_{\tilde{r}+2}, \ldots, \tilde{\lambda}_{\hat{p}_F}$ are close to zero. Let

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

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

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

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

and sizes of $\tilde{\mathbf{U}}_{(\tilde{r})}$, $\tilde{\mathbf{U}}_{(\hat{p}_F - \tilde{r})}$, $\boldsymbol{\vartheta}_{(\tilde{r})}$, and $\boldsymbol{\vartheta}_{(\hat{p}_F - \tilde{r})}$ 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. The model (4.1.12) can now be written as

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\tilde{r})} \boldsymbol{\vartheta}_{(\tilde{r})} + \tilde{\mathbf{U}}_{(\hat{p}_F - \tilde{r})} \boldsymbol{\vartheta}_{(\hat{p}_F - \tilde{r})} + \tilde{\boldsymbol{\epsilon}}. \tag{4.1.13}$$

The term $\tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})}\boldsymbol{\vartheta}_{(\hat{p}_F-\tilde{r})}$ in the model (4.1.13) will give us the ill effect of multicollinearity. To avoid the effect of multicollinearity, we drop the term $\tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})}\boldsymbol{\vartheta}_{(\hat{p}_F-\tilde{r})}$ and obtain

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})} + \tilde{\tilde{\boldsymbol{\epsilon}}}, \tag{4.1.14}$$

where $\tilde{\tilde{\boldsymbol{\epsilon}}}$ is a random vector influenced by dropping $\tilde{\mathbf{U}}_{(\hat{p}_F-\tilde{r})}\boldsymbol{\vartheta}_{(\hat{p}_F-\tilde{r})}$ in the model (4.1.13). The model (4.1.14) shows that the ill effect of collinearity and multicollinearity on $\boldsymbol{\Psi}$ are avoided by using the matrix $\tilde{\mathbf{A}}$.

Note that $\tilde{\mathbf{U}}_{(\tilde{r})}^T\tilde{\mathbf{U}}_{(\tilde{r})} = \tilde{\mathbf{D}}_{(\tilde{r})}$, which is invertible. Hence, the estimator of $\boldsymbol{\vartheta}_{(\tilde{r})}$, say $\hat{\boldsymbol{\vartheta}}_{(\tilde{r})}$, is given by

$$\hat{\boldsymbol{\vartheta}}_{(\tilde{r})} = (\tilde{\mathbf{U}}_{(\tilde{r})}^T\tilde{\mathbf{U}}_{(\tilde{r})})^{-1}\tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{Y}_o. \tag{4.1.15}$$

The terms $\tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{Y}_o$ and $\tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{Y}$ are related by the following lemma.

Lemma 4.1.1. $\tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{Y} = \tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{Y}_o$.

Proof. See Appendix B.3. □

Let $\hat{\boldsymbol{\vartheta}}_{(\tilde{r})}^* \in \mathbb{R}^{\tilde{r}}$ be the value of $\hat{\boldsymbol{\vartheta}}_{(\tilde{r})}$ when $\mathbf{Y}_o$ is replaced by $\mathbf{y}_o$ in the Eq. (4.1.15), where $\mathbf{y}_o$ is the observed data corresponding to $\mathbf{Y}_o$. By using Lemma 4.1.1, we obtain

$$\hat{\boldsymbol{\vartheta}}_{(\tilde{r})}^* = (\tilde{\mathbf{U}}_{(\tilde{r})}^T\tilde{\mathbf{U}}_{(\tilde{r})})^{-1}\tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{y}. \tag{4.1.16}$$

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

$$\tilde{\mathbf{y}}_{kpcr} := \bar{y}\mathbf{1}_N + \tilde{\mathbf{U}}_{(\tilde{r})}\hat{\boldsymbol{\vartheta}}_{(\tilde{r})}^*. \tag{4.1.17}$$

Since

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

we obtain $\tilde{\mathbf{U}}_{(\tilde{r})} = \boldsymbol{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})}$. The Eq. (4.1.17) can be now written as

$$\tilde{\mathbf{y}}_{kpcr} = \bar{y}\mathbf{1}_N + \boldsymbol{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})}\hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})}. \tag{4.1.18}$$

The *prediction by the revised KPCR* is given by

$$g_{kpcr}(\mathbf{x}) := \bar{y} + \psi(\mathbf{x})^T\tilde{\mathbf{A}}_{(\tilde{r})}\hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})}, \tag{4.1.19}$$

where g_{kpcr} is a function from $\mathbb{R}^p$ to $\mathbb{R}$. The elements of $\psi(\mathbf{x})^T\tilde{\mathbf{A}}_{(\tilde{r})} = \begin{pmatrix}\psi(\mathbf{x})^T\tilde{\mathbf{a}}_1 & \dots & \psi(\mathbf{x})^T\tilde{\mathbf{a}}_{\tilde{r}}\end{pmatrix}$ are the 1st, ..., $\tilde{r}$th nonlinear principal components corresponding to ψ, respectively, which are given by Eq. (2.2.2).

Since $\tilde{\mathbf{U}}_{(\tilde{r})} = \boldsymbol{\Psi}\tilde{\mathbf{A}}_{(\tilde{r})}$ and $\tilde{\mathbf{A}}_{(\tilde{r})} = (\tilde{\mathbf{a}}_1 \quad \tilde{\mathbf{a}}_2 \quad \dots \quad \tilde{\mathbf{a}}_{\tilde{r}}) = \boldsymbol{\Psi}^T\boldsymbol{\Gamma}_{(\tilde{r})}$, we obtain that

$$\tilde{\mathbf{U}}_{(\tilde{r})} = \boldsymbol{\Psi}\boldsymbol{\Psi}^T\boldsymbol{\Gamma}_{(\tilde{r})} = \mathbf{K}\boldsymbol{\Gamma}_{(\tilde{r})}. \tag{4.1.20}$$

where $\boldsymbol{\Gamma}_{(\tilde{r})} = \begin{pmatrix}\boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \dots & \boldsymbol{\alpha}_{\tilde{r}}\end{pmatrix}$. Note that $\boldsymbol{\alpha}_l = \frac{\tilde{\mathbf{b}}_l}{\sqrt{\tilde{\lambda}_l}}$ and $\tilde{\mathbf{b}}_l$ is a normalized eigenvector of $\mathbf{K}$ for $l = 1, 2, \dots, \tilde{r}$. Hence, $\tilde{\mathbf{U}}_{(\tilde{r})}$ is explicitly known now and Eq. (4.1.16) becomes

$$\hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})} = ((\mathbf{K}\boldsymbol{\Gamma}_{(\tilde{r})})^T(\mathbf{K}\boldsymbol{\Gamma}_{(\tilde{r})}))^{-1}(\mathbf{K}\boldsymbol{\Gamma}_{(\tilde{r})})^T\mathbf{y}, \tag{4.1.21}$$

the prediction value $\tilde{\mathbf{y}}_{kpcr}$ can now be written as

$$\tilde{\mathbf{y}}_{kpcr} = \bar{y}\mathbf{1}_N + \mathbf{K}\boldsymbol{\Gamma}_{(\tilde{r})}\hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})}, \tag{4.1.22}$$

and the prediction by the revised KPCR is written as

$$g_{kpcr}(\mathbf{x}) = \bar{y} + \sum_{i=1}^{N} c_i \kappa(\mathbf{x}, \mathbf{x}_i), \tag{4.1.23}$$

where $\begin{pmatrix} c_1 & c_2 & \dots & c_N \end{pmatrix}^T = \mathbf{\Gamma}_{(\tilde{r})} \hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})}$. The number $\tilde{r}$ is called the *retained number of nonlinear PCs for the revised KPCR model*.

4.1.3 Revised KPCR's Algorithm

We summarize the procedure in 4.1.2 to obtain the prediction by the revised KPCR.

Algorithm:

1. Given $(y_i, x_{i1}, x_{i2}, \dots, x_{ip})$, $i = 1, 2, \dots, N$.

2. Calculate $\bar{y} = \frac{1}{N} \mathbf{1}_N^T \mathbf{y}$.

3. Choose a kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$.

4. Construct $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{K} = (K_{ij})$.

5. Diagonalize $\mathbf{K}$.

   Let $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \dots \geq \tilde{\lambda}_{\tilde{r}} \geq \tilde{\lambda}_{\tilde{r}+1} \geq \dots \geq \tilde{\lambda}_N \geq 0$ be the eigenvalues of $\mathbf{K}$ and $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2 \dots, \tilde{\mathbf{b}}_{\tilde{r}}, \tilde{\mathbf{b}}_{\tilde{r}+1}, \dots, \tilde{\mathbf{b}}_N$ be the corresponding normalized eigenvectors of $\mathbf{K}$.

6. Detect collinearity and multicollinearity on $\mathbf{K}$.

   Let $\tilde{r}$ be the retained number of nonlinear PCs such that $\tilde{r} = \max\{r | \frac{\lambda_r}{\lambda_1} \geq \frac{1}{1000}\}$.

7. Construct $\boldsymbol{\alpha}_l = \frac{\tilde{\mathbf{b}}_l}{\sqrt{\tilde{\lambda}_l}}$ for $l = 1, 2, \ldots, \tilde{r}$ and $\boldsymbol{\Gamma}_{(\tilde{r})} = \begin{pmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \ldots & \boldsymbol{\alpha}_{\tilde{r}} \end{pmatrix}$.

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

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

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

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 have only to replace $\mathbf{K}$ by $\mathbf{K}_N := \mathbf{K} - \mathbf{EK} - \mathbf{KE} + \mathbf{EKE}$ in Step 4, where $\mathbf{E}$ is the $N \times N$ matrix with all elements equal to $\frac{1}{N}$. Further, we diagonalize $\mathbf{K}_N$ in Step 5 and work based on $\mathbf{K}_N$ in the subsequent steps.

4.2 Weighted Least Squares in Kernel Principal Component Regression

4.2.1 WLS-KPCR

Let us consider the linear regression model in the feature space

$$\mathbf{Y}_o = \boldsymbol{\Psi}\boldsymbol{\gamma} + \boldsymbol{\epsilon}_2, \tag{4.2.1}$$
$$E(\boldsymbol{\epsilon}_2) = \mathbf{0},$$
$$Var(\boldsymbol{\epsilon}_2) = \sigma_2^2 \breve{\mathbf{V}},$$

where $\boldsymbol{\gamma} = (\gamma_1 \quad \gamma_2 \quad \ldots \quad \gamma_{p_F})^T$ is a vector of regression coefficients in the feature space, $\boldsymbol{\epsilon}_2$ is a vector of random errors in the feature space, $\mathbf{Y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{Y}$ and $\breve{\mathbf{V}} = \text{diag}(1/\breve{w}_1, 1/\breve{w}_2, \ldots, 1/\breve{w}_N)$ and $\breve{w}_i$ is a positive number for $i = 1, 2, \ldots, N$. The weight $\breve{w}_i$ is estimated by using the data $\mathbf{y_o} = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y}$ and $\mathbf{X}$. Let $\breve{\mathbf{L}} = \text{diag}(1/\sqrt{\breve{w}_1}, 1/\sqrt{\breve{w}_2}, \ldots, 1/\sqrt{\breve{w}_N})$. Hence, $\breve{\mathbf{L}}^T = \breve{\mathbf{L}}$, $\breve{\mathbf{L}}\breve{\mathbf{L}}^T = \breve{\mathbf{V}}$ and $\breve{\mathbf{L}}^{-1} = \text{diag}(\sqrt{\breve{w}_1}, \sqrt{\breve{w}_2}, \ldots, \sqrt{\breve{w}_N})$. Then, we have

$$\begin{aligned} \mathbf{Z}_o &= \boldsymbol{\theta}\boldsymbol{\gamma} + \tilde{\boldsymbol{\epsilon}}_2, \qquad (4.2.2) \\ E(\tilde{\boldsymbol{\epsilon}}_2) &= \mathbf{0}, \\ Var(\tilde{\boldsymbol{\epsilon}}_2) &= \sigma_2^2\mathbf{I}_N, \end{aligned}$$

where $\mathbf{Z}_o = \breve{\mathbf{L}}^{-1}\mathbf{Y}_o$, $\boldsymbol{\theta} = \breve{\mathbf{L}}^{-1}\boldsymbol{\Psi}$ and $\tilde{\boldsymbol{\epsilon}}_2 = \breve{\mathbf{L}}^{-1}\boldsymbol{\epsilon}_2$. Furthermore, we define two matrices $\breve{\mathbf{K}} := \boldsymbol{\theta}\boldsymbol{\theta}^T = \breve{\mathbf{L}}^{-1}\mathbf{K}\breve{\mathbf{L}}^{-1}$ and $\breve{\mathbf{C}} := \frac{1}{N}\boldsymbol{\theta}^T\boldsymbol{\theta}$ where $\mathbf{K} = \boldsymbol{\Psi}\boldsymbol{\Psi}^T$. Note that $\mathbf{K}$ is explicitly known by choosing a kernel function κ. The relation of eigenvalues and eigenvectors of the matrices $\breve{\mathbf{C}}$ and $\breve{\mathbf{K}}$ are related by Theorem 2.2.1.

Let $\breve{p}_F$ be the rank of $\boldsymbol{\theta}$ where $\breve{p}_F \leq \min\{N, p_F\}$. Since the rank of $\boldsymbol{\theta}$ is equal to the rank of $\breve{\mathbf{K}}$ and the rank of $\boldsymbol{\theta}^T\boldsymbol{\theta}$, then the rank of $\breve{\mathbf{K}}$ and the rank of $\boldsymbol{\theta}^T\boldsymbol{\theta}$ are equal to $\breve{p}_F$. Note that, $\breve{\mathbf{K}}$ is symmetric and positive semidefinite. This implies that the eigenvalues of $\breve{\mathbf{K}}$ are nonnegative real numbers. Let $\breve{\lambda}_1 \geq \breve{\lambda}_2 \geq \ldots \geq \breve{\lambda}_{\breve{r}} \geq \breve{\lambda}_{\breve{r}+1} \geq \ldots \geq \breve{\lambda}_{\breve{p}_F} > \breve{\lambda}_{\breve{p}_F+1} = \ldots = \breve{\lambda}_N = 0$ be the eigenvalues of $\breve{\mathbf{K}}$ and $\breve{\mathbf{B}} = (\breve{\mathbf{b}}_1 \quad \breve{\mathbf{b}}_2 \quad \ldots \quad \breve{\mathbf{b}}_N)$ be the matrix of the corresponding normalized eigenvectors $\breve{\mathbf{b}}_i$ $(i = 1, 2, \ldots, N)$ of $\breve{\mathbf{K}}$. Then, let

$\breve{\boldsymbol{\alpha}}_l = \frac{\breve{\mathbf{b}}_l}{\sqrt{\breve{\lambda}_l}}$ and $\breve{\mathbf{a}}_l = \boldsymbol{\theta}^T \breve{\boldsymbol{\alpha}}_l$ for $l = 1, 2, \ldots, \breve{p}_F$. By Theorem 2.2.1 we obtain

$$\begin{aligned} \tfrac{\breve{\lambda}_l}{N} \breve{\mathbf{a}}_l &= \breve{\mathbf{C}} \breve{\mathbf{a}}_l \quad \text{for } l = 1, 2, \ldots, \breve{p}_F \\ \breve{\mathbf{a}}_i^T \breve{\mathbf{a}}_j &= \begin{cases} 1 & \text{if } i = j, \quad \text{for } i, j = 1, 2, \ldots, \breve{p}_F, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

or equivalent to

$$\begin{aligned} \breve{\lambda}_l \breve{\mathbf{a}}_l &= \boldsymbol{\theta}^T \boldsymbol{\theta} \breve{\mathbf{a}}_l \quad \text{for } l = 1, 2, \ldots, \breve{p}_F \\ \breve{\mathbf{a}}_i^T \breve{\mathbf{a}}_j &= \begin{cases} 1 & \text{if } i = j, \quad \text{for } i, j = 1, 2, \ldots, \breve{p}_F, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Since the rank of $\boldsymbol{\theta}^T \boldsymbol{\theta}$ is equal to $\breve{p}_F$, then the remaining $(p_F - \breve{p}_F)$ eigenvalues of $\boldsymbol{\theta}^T \boldsymbol{\theta}$ are zero eigenvalues. Let $\breve{\lambda}_k$, $(k = \breve{p}_F + 1, \breve{p}_F + 2, \ldots, p_F)$, be the zero eigenvalues of $\boldsymbol{\theta}^T \boldsymbol{\theta}$ and $\breve{\mathbf{a}}_k$ be the normalized eigenvectors of $\boldsymbol{\theta}^T \boldsymbol{\theta}$ corresponding to $\breve{\lambda}_k$. Hence, we have

$$\begin{aligned} \breve{\lambda}_l \breve{\mathbf{a}}_l &= \boldsymbol{\theta}^T \boldsymbol{\theta} \breve{\mathbf{a}}_l \quad \text{for } l = 1, 2, \ldots, p_F \\ \breve{\mathbf{a}}_i^T \breve{\mathbf{a}}_j &= \begin{cases} 1 & \text{if } i = j, \quad \text{for } i, j = 1, 2, \ldots, p_F, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Furthermore, we define $\breve{\mathbf{A}} = (\breve{\mathbf{a}}_1 \quad \breve{\mathbf{a}}_2 \quad \ldots \quad \breve{\mathbf{a}}_{p_F})$. It is evident that $\breve{\mathbf{A}}$ is an orthogonal matrix, that is, $\breve{\mathbf{A}}^T = \breve{\mathbf{A}}^{-1}$. It is not difficult to verify that

$$\breve{\mathbf{A}}^T \boldsymbol{\theta}^T \boldsymbol{\theta} \breve{\mathbf{A}} = \breve{\mathbf{D}},$$

where

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

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

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

By using $\breve{\mathbf{A}}\breve{\mathbf{A}}^T = \mathbf{I}_{p_F}$, we can rewrite the model (4.2.2) as

$$\mathbf{Z}_o = \breve{\mathbf{U}}\boldsymbol{\vartheta} + \tilde{\boldsymbol{\epsilon}}_{\mathbf{2}}, \tag{4.2.3}$$

$$E(\tilde{\boldsymbol{\epsilon}}_2) = \mathbf{0},$$

$$Var(\tilde{\boldsymbol{\epsilon}}_2) = \sigma_2^2 \mathbf{I}_N,$$

where $\breve{\mathbf{U}} = \boldsymbol{\theta}\mathbf{A}$ and $\boldsymbol{\vartheta} = \mathbf{A}^T\boldsymbol{\gamma}$. Let

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

where sizes of $\breve{\mathbf{U}}_{(\breve{p}_F)}$, $\breve{\mathbf{U}}_{(p_F - \breve{p}_F)}$, $\boldsymbol{\vartheta}_{(\breve{p}_F)}$, and $\boldsymbol{\vartheta}_{(p_F - \breve{p}_F)}$ are $N \times \breve{p}_F$, $N \times (p_F - \breve{p}_F)$, $\hat{p_F} \times 1$ and $(p_F - \breve{p}_F) \times 1$, respectively. The model (4.2.3) can be written as

$$\mathbf{Z}_o = \breve{\mathbf{U}}_{(\breve{p}_F)}\boldsymbol{\vartheta}_{(\breve{p}_F)} + \breve{\mathbf{U}}_{(p_F - \breve{p}_F)}\boldsymbol{\vartheta}_{(p_F - \breve{p}_F)} + \tilde{\boldsymbol{\epsilon}}_2, \tag{4.2.4}$$

$$E(\tilde{\boldsymbol{\epsilon}}_2) = \mathbf{0},$$

$$Var(\tilde{\boldsymbol{\epsilon}}_2) = \sigma_2^2 \mathbf{I}_N.$$

As we see that $\breve{\mathbf{D}} = \breve{\mathbf{A}}^T\boldsymbol{\theta}^T\boldsymbol{\theta}\breve{\mathbf{A}} = \breve{\mathbf{U}}^T\breve{\mathbf{U}}$, and we obtain

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

and

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

Since $(\breve{\mathbf{U}}_{(p_F-\breve{p}_F)}\boldsymbol{\vartheta}_{(p_F-\breve{p}_F)})^T(\breve{\mathbf{U}}_{(p_F-\breve{p}_F)}\boldsymbol{\vartheta}_{(p_F-\breve{p}_F)})$ is equal to zero, we see that $\breve{\mathbf{U}}_{(p_F-\breve{p}_F)}\boldsymbol{\vartheta}_{(p_F-\breve{p}_F)}$ is equal to $\mathbf{0}$. Consequently, the model (4.2.4) is simplified to

$$\mathbf{Z}_o = \breve{\mathbf{U}}_{(\breve{p}_F)}\boldsymbol{\vartheta}_{(\breve{p}_F)} + \tilde{\boldsymbol{\epsilon}}_2, \tag{4.2.5}$$
$$E(\tilde{\boldsymbol{\epsilon}}_2) = \mathbf{0},$$
$$Var(\tilde{\boldsymbol{\epsilon}}_2) = \sigma_2^2\mathbf{I}_N.$$

Let us assume that $\breve{\lambda}_{\breve{r}+1}, \breve{\lambda}_{\breve{r}+2}, \ldots, \breve{\lambda}_{\breve{p}_F}$ are close to zero. Let

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

and

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

where

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

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

and sizes of $\breve{\mathbf{U}}_{(\breve{r})}$, $\breve{\mathbf{U}}_{(\breve{p}_F-\breve{r})}$, $\boldsymbol{\vartheta}_{(\breve{r})}$, and $\boldsymbol{\vartheta}_{(\breve{p}_F-\breve{r})}$ are $N \times \breve{r}$, $N \times (\breve{p}_F - \breve{r})$, $\breve{r} \times 1$ and $(\breve{p}_F - \breve{r}) \times 1$, respectively. The model (4.2.5) can now be written as

$$\mathbf{Z}_o = \breve{\mathbf{U}}_{(\breve{r})}\boldsymbol{\vartheta}_{(\breve{r})} + \breve{\mathbf{U}}_{(\breve{p}_F-\breve{r})}\boldsymbol{\vartheta}_{(\breve{p}_F-\breve{r})} + \tilde{\boldsymbol{\epsilon}}_2 \tag{4.2.6}$$

$$E(\tilde{\boldsymbol{\epsilon}}_2) = \mathbf{0},$$

$$Var(\tilde{\boldsymbol{\epsilon}}_2) = \sigma_2^2\mathbf{I}_N,$$

It is evident that the estimator of $\boldsymbol{\vartheta}_{(\breve{p}_F-\breve{r})}$, say $\breve{\boldsymbol{\vartheta}}_{(\breve{p}_F-\breve{r})} = (\breve{\vartheta}_{\breve{r}+1} \quad \breve{\vartheta}_{\breve{r}+2} \quad \dots \quad \breve{\vartheta}_{\breve{p}_F-r})^T$, is given by

$$\breve{\boldsymbol{\vartheta}}_{(\breve{p}_F-\breve{r})} = (\breve{\mathbf{U}}^T_{(\breve{p}_F-\breve{r})}\breve{\mathbf{U}}_{(\breve{p}_F-\breve{r})})^{-1}\breve{\mathbf{U}}^T_{(\breve{p}_F-\breve{r})}\mathbf{Z}_o = \breve{\mathbf{D}}^{-1}_{(\breve{p}_F-\breve{r})}\breve{\mathbf{U}}^T_{(\breve{p}_F-\breve{r})}\mathbf{Z}_o, \tag{4.2.7}$$

and the variance of $\breve{\vartheta}_j$ $(j = \breve{r}+1, \ldots, \breve{p}_F - \breve{r})$ is

$$Var(\breve{\vartheta}_j) = \sigma^2(\breve{\mathbf{D}}^{-1}_{(\breve{p}-\breve{r})})_{jj}. \tag{4.2.8}$$

Since $\breve{\lambda}_{\breve{r}+1}, \breve{\lambda}_{\breve{r}+2}, \ldots, \breve{\lambda}_{\breve{p}_F-\breve{r}}$ are close to zero, the diagonal elements of $\breve{\mathbf{D}}^{-1}_{(\breve{p}_F-\breve{r})}$ and also the variance of $\breve{\vartheta}_j$ $(j = \breve{r}+1, \ldots, \breve{p}_F - \breve{r})$ will be very large numbers. Thus, we encounter the ill effect of multicollinearity in the model (4.2.6). To avoid the effects of multicollinearity, we drop the term $\breve{\mathbf{U}}_{(\breve{p}_F-\breve{r})}\boldsymbol{\vartheta}_{(\breve{p}_F-\breve{r})}$ as in [44] and obtain

$$\mathbf{Z}_o = \breve{\mathbf{U}}_{(\breve{r})}\boldsymbol{\vartheta}_{(\breve{r})} + \tilde{\tilde{\boldsymbol{\epsilon}}}, \tag{4.2.9}$$

where $\tilde{\tilde{\boldsymbol{\epsilon}}}$ is a random vector influenced by dropping $\breve{\mathbf{U}}_{(\breve{p}_F-\breve{r})}\boldsymbol{\vartheta}_{(\breve{p}_F-\breve{r})}$ in the model (4.2.9). The model (4.2.9) shows that the ill effects multicollinearity on $\breve{\mathbf{U}}_{(\breve{p}_F)}$ are avoided by using the matrix $\breve{\mathbf{A}}$.

Note that $\breve{\mathbf{U}}^T_{(\breve{r})}\breve{\mathbf{U}}_{(\breve{r})} = \breve{\mathbf{D}}_{(\breve{r})}$, which is invertible. Hence, the estimator of $\boldsymbol{\vartheta}_{(\breve{r})}$, say $\breve{\boldsymbol{\vartheta}}_{(\breve{r})}$, is given by

$$\breve{\boldsymbol{\vartheta}}_{(\breve{r})} = (\breve{\mathbf{U}}^T_{(\breve{r})}\breve{\mathbf{U}}_{(\breve{r})})^{-1}\breve{\mathbf{U}}^T_{(\breve{r})}\mathbf{Z}_o. \tag{4.2.10}$$

Let $\mathbf{z_o} = \tilde{\mathbf{L}}^{-1}(I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y}$ be the observed data corresponding to $\mathbf{Z_o}$ and $\breve{\boldsymbol{\vartheta}}^*_{(\breve{r})} \in \mathbb{R}^{\breve{r}}$ be the value of $\breve{\boldsymbol{\vartheta}}_{(\breve{r})}$ when $\mathbf{Z}_o$ is replaced by $\mathbf{z}_o$ in the Eq. (4.2.10). Hence

$$\begin{aligned}\breve{\boldsymbol{\vartheta}}^*_{(\breve{r})} &= (\breve{\mathbf{U}}^T_{(\breve{r})}\breve{\mathbf{U}}_{(\breve{r})})^{-1}\breve{\mathbf{U}}^T_{(\breve{r})}\mathbf{z}_o, \\ &= \breve{\mathbf{D}}^{-1}_{(\breve{r})}\breve{\mathbf{U}}^T_{(\breve{r})}\mathbf{z}_o.\end{aligned} \tag{4.2.11}$$

Since

$$\breve{\mathbf{U}} = \begin{pmatrix} \breve{\mathbf{U}}_{(\breve{r})} & \breve{\mathbf{U}}_{(\breve{p}_F-\breve{r})} & \breve{\mathbf{U}}_{(p_F-\breve{p}_F)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\theta}\breve{\mathbf{A}}_{(\breve{r})} & \boldsymbol{\theta}\breve{\mathbf{A}}_{(\breve{p}_F-\breve{r})} & \boldsymbol{\theta}\breve{\mathbf{A}}_{(p_F-\breve{p}_F)} \end{pmatrix},$$

we obtain $\breve{\mathbf{U}}_{(\breve{r})} = \boldsymbol{\theta}\breve{\mathbf{A}}_{(\breve{r})}$. As we see that $\breve{\mathbf{A}}_{(\breve{r})} = \boldsymbol{\theta}^T \begin{pmatrix} \breve{\boldsymbol{\alpha}}_1 & \breve{\boldsymbol{\alpha}}_2 & \dots & \breve{\boldsymbol{\alpha}}_{\breve{r}} \end{pmatrix}$. Hence,

$$\breve{\mathbf{U}}_{(\breve{r})} = \boldsymbol{\theta}\boldsymbol{\theta}^T\breve{\boldsymbol{\Gamma}}_{(\breve{r})} = \breve{\mathbf{K}}\breve{\boldsymbol{\Gamma}}_{(\breve{r})}, \tag{4.2.12}$$

where $\breve{\boldsymbol{\Gamma}}_{(\breve{r})} = \begin{pmatrix} \breve{\boldsymbol{\alpha}}_1 & \breve{\boldsymbol{\alpha}}_2 & \dots & \breve{\boldsymbol{\alpha}}_{\breve{r}} \end{pmatrix}$. Since $\mathbf{K}$ is known explicitly, $\breve{\mathbf{K}} = \breve{\mathbf{L}}^{-1}\mathbf{K}\breve{\mathbf{L}}^{-1}$, $\breve{\boldsymbol{\Gamma}}_{(\breve{r})}$ and $\breve{\mathbf{U}}_{(\breve{r})}$ are also known explicitly.

The *prediction value* of $\mathbf{z}_o$ $(= \tilde{\mathbf{L}}^{-1}(\mathbf{y} - \bar{y}\mathbf{1}_N))$, say $\breve{\mathbf{z}}_o$ $(= \tilde{\mathbf{L}}^{-1}(\breve{\mathbf{y}} - \bar{y}\mathbf{1}_N))$ is given by

$$\breve{\mathbf{z}}_{\mathbf{o}} := \begin{pmatrix} \hat{z}_{\mathbf{o}1} & \hat{z}_{\mathbf{o}2} & \dots & \hat{z}_{\mathbf{o}N} \end{pmatrix}^T = \breve{\mathbf{K}}\breve{\boldsymbol{\Gamma}}_{(\breve{r})}\breve{\boldsymbol{\vartheta}}^*_{(\breve{r})}. \tag{4.2.13}$$

The residual between $\mathbf{z}_o$ and $\breve{\mathbf{z}}_o$ is given by

$$\mathbf{e_2} := \begin{pmatrix} e_{\mathbf{2}1} & e_{\mathbf{2}2} & \dots & e_{\mathbf{2}N} \end{pmatrix}^T = \mathbf{z}_o - \breve{\mathbf{z}}_{\mathbf{o}}, \tag{4.2.14}$$

and the *prediction by the WLS KPCR* is given by

$$g_{wls-kpcr}(\mathbf{x}) := \bar{y} + \sum_{i=1}^{N} \breve{c}_i \kappa(\mathbf{x}, \mathbf{x}_i), \tag{4.2.15}$$

where $g_{wls-kpcr}$ is a function from $\mathbb{R}^p$ to $\mathbb{R}$ and $\begin{pmatrix} \breve{c}_1 & \breve{c}_2 & \dots & \breve{c}_N \end{pmatrix}^T = \breve{\mathbf{L}}^{-1}\breve{\boldsymbol{\Gamma}}_{(\breve{r})}\breve{\boldsymbol{\vartheta}}^*_{(\breve{r})}$. The number $\breve{r}$ is called the *retained number of nonlinear PCs for the WLS KPCR*.

4.2.2 WLS KPCR's Algorithm

We summarize the procedure in 4.2.1 to obtain the prediction by WLS KPCR.

Algorithm:

1. Given $(y_i, x_{i1}, x_{i2}, \ldots, x_{ip})$, $i = 1, 2, \ldots, N$.
2. Calculate $\bar{y} = \frac{1}{N}\mathbf{1}_N^T\mathbf{y}$ and $\boldsymbol{y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y}$.
3. Estimate $\breve{\mathbf{V}}$ and find $\breve{\mathbf{L}}$.
4. Calculate $\mathbf{z}_o = \breve{\mathbf{L}}^{-1}\mathbf{y}_o$.
5. Choose a kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$.
6. Construct $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{K} = (K_{ij})$.
7. Construct $\breve{\mathbf{K}} = \breve{\mathbf{L}}^{-1}\mathbf{K}\breve{\mathbf{L}}^{-1}$.
8. Diagonalize $\breve{\mathbf{K}}$.
   Let $\breve{\lambda}_1 \geq \breve{\lambda}_2 \geq \ldots \geq \breve{\lambda}_r \geq \ldots \geq \breve{\lambda}_{\breve{p}_F} > \breve{\lambda}_{\breve{p}_F+1} = \ldots = \breve{\lambda}_N = 0$ be the eigenvalues of $\breve{\mathbf{K}}$ and $\breve{\mathbf{b}}_1, \breve{\mathbf{b}}_2 \ldots, \breve{\mathbf{b}}_N$ be the corresponding normalized eigenvectors of $\breve{\mathbf{K}}$.
9. Detect collinearity and multicollinearity on $\breve{\mathbf{K}}$.
   Let $\breve{r}$ be the retained number of nonlinear PCs such that $\breve{r} = \max\{s|\frac{\breve{\lambda}_s}{\breve{\lambda}_1} \geq \frac{1}{1000}\}$.
10. Construct $\breve{\boldsymbol{\alpha}}_l = \frac{\breve{\mathbf{b}}_l}{\sqrt{\breve{\lambda}_l}}$ for $l = 1, 2, \ldots, \breve{r}$ and $\breve{\boldsymbol{\Gamma}}_{(\breve{r})} = \begin{pmatrix} \breve{\boldsymbol{\alpha}}_1 & \breve{\boldsymbol{\alpha}}_2 & \ldots & \breve{\boldsymbol{\alpha}}_{\breve{r}} \end{pmatrix}$.
11. Calculate $\breve{\mathbf{U}}_{(\breve{r})} = \breve{\mathbf{K}}\breve{\boldsymbol{\Gamma}}_{(\breve{r})}$, $\breve{\boldsymbol{\vartheta}}^*_{(\breve{r})} = \breve{\mathbf{D}}^{-1}_{(\breve{r})}\breve{\mathbf{U}}^T_{(\breve{r})}\mathbf{z}_o$
    and $\breve{\mathbf{c}} = \begin{pmatrix} \breve{c}_1 & \breve{c}_2 & \ldots & \breve{c}_N \end{pmatrix}^T = \breve{\mathbf{L}}^{-1}\breve{\boldsymbol{\Gamma}}_{(\breve{r})}\breve{\boldsymbol{\vartheta}}^*_{(\breve{r})}$.
12. Given a vector $\mathbf{x} \in \mathbb{R}^p$, the prediction by WLS KPCR is given by

$$g_{wls-kpcr}(\mathbf{x}) = \bar{y} + \textstyle\sum_{j=1}^N \breve{c}_i\kappa(\mathbf{x}, \mathbf{x}_j).$$

We also notice 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 have only to replace $\mathbf{K}$ by $\mathbf{K}_N$ in Step 7.

4.3 KPCR and M-Estimation in Robust Regression Model

4.3.1 Robust Kernel Principal Component Regression

Let us consider again the standard centered multiple linear regression model in the feature space

$$\mathbf{Y}_o = \mathbf{\Psi}\boldsymbol{\gamma} + \tilde{\boldsymbol{\epsilon}}, \tag{4.3.1}$$

where $\boldsymbol{\gamma} = \begin{pmatrix} \gamma_1 & \gamma_2 & \dots & \gamma_{p_F} \end{pmatrix}^T$ is a vector of regression coefficients in the feature space, $\tilde{\boldsymbol{\epsilon}}$ is a vector of random errors in the feature space and $\mathbf{Y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{Y}$. Note that $\mathbf{y}_o$ is the observed data corresponding to $\mathbf{Y}_o$. Hence, we have

$$\mathbf{y}_o = \mathbf{\Psi}\boldsymbol{\gamma} + \tilde{\boldsymbol{e}}, \tag{4.3.2}$$

where $\tilde{\boldsymbol{e}} \in \mathbb{R}^N$ is a vector of residuals.

As mentioned before that $\mathbf{\Psi} = \begin{pmatrix} \psi(\mathbf{x}_1) & \dots & \psi(\mathbf{x}_N) \end{pmatrix}^T$, $\mathbf{K} = \mathbf{\Psi}\mathbf{\Psi}^T$, $\tilde{\mathbf{C}} = \frac{1}{N}\mathbf{\Psi}^T\mathbf{\Psi}$, $\hat{p}_F$ is the rank of $\mathbf{\Psi}$ where $\hat{p}_F \leq \min\{N, p_F\}$. The eigenvalues of $\mathbf{K}$ are $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_{\acute{r}} \geq \tilde{\lambda}_{\acute{r}+1} \geq \ldots \geq \tilde{\lambda}_{\hat{p}_F} > \tilde{\lambda}_{\hat{p}_F+1} = \ldots = \tilde{\lambda}_N = 0$ and $\mathbf{B} = (\mathbf{b}_1 \quad \mathbf{b}_2 \quad \dots \quad \mathbf{b}_N)$ are the matrix of the corresponding normalized eigenvectors $\mathbf{b}_l$ of $\mathbf{K}$. We have defined that $\boldsymbol{\alpha}_l = \frac{\mathbf{b}_l}{\sqrt{\tilde{\lambda}_l}}$ and $\tilde{\mathbf{a}}_l = \mathbf{\Psi}^T\boldsymbol{\alpha}_l$ for $l = 1, 2, \ldots, \hat{p}_F$. We have also defined that $\tilde{\mathbf{A}} = (\tilde{\mathbf{a}}_1 \quad \tilde{\mathbf{a}}_2 \quad \dots \quad \tilde{\mathbf{a}}_{p_F})$,

$\mathbf{U}_{(\hat{p}_F)} = \mathbf{\Psi}\tilde{\mathbf{A}}$ and $\boldsymbol{\vartheta}_{(\hat{p}_F)} = \tilde{\mathbf{A}}^T\boldsymbol{\gamma}$. Note that $\tilde{\mathbf{A}}$ is an orthogonal matrix, that is, $\tilde{\mathbf{A}}^T = \tilde{\mathbf{A}}^{-1}$. Then, Eq. (4.3.2) reduces to

$$\mathbf{y}_o = \mathbf{U}_{(\hat{p}_F)}\boldsymbol{\vartheta}_{(\hat{p}_F)} + \tilde{\boldsymbol{e}}, \tag{4.3.3}$$

where $\mathbf{U}_{(\hat{p}_F)}$ is explicitly known. Note that, $\mathbf{U}_{(\hat{p}_F)} = \mathbf{K}\mathbf{\Gamma}_{(\hat{p}_F)}$ and $\mathbf{\Gamma}_{(\hat{p}_F)} = \begin{pmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \dots & \boldsymbol{\alpha}_{\hat{p}_F} \end{pmatrix}$. Furthermore, model (4.3.3) is written as

$$\mathbf{y}_o = \mathbf{U}_{(\acute{r})}\boldsymbol{\vartheta}_{(\acute{r})} + \mathbf{U}_{(\hat{p}_F-\acute{r})}\boldsymbol{\vartheta}_{(\hat{p}_F-\acute{r})} + \tilde{\boldsymbol{e}}. \tag{4.3.4}$$

If we only use the first $\acute{r}$ vectors of $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_{\hat{p}_F}$, model (4.3.4) becomes

$$\mathbf{y}_o = \mathbf{U}_{(\acute{r})}\boldsymbol{\vartheta}_{(\acute{r})} + \tilde{\mathbf{e}}_{\mathbf{1}}, \tag{4.3.5}$$

where $\tilde{\mathbf{e}}_{\mathbf{1}} = \begin{pmatrix} \tilde{e}_{\mathbf{11}} & \tilde{e}_{\mathbf{12}} & \dots & \tilde{e}_{\mathbf{1}N} \end{pmatrix}^T$ is a vector of residuals influenced by dropping the term $\mathbf{U}_{(\hat{p}_F-\acute{r})}\boldsymbol{\vartheta}_{(\hat{p}_F-\acute{r})}$ in model (4.3.4). Let $\mathbf{U}_{(\acute{r})} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_N)^T$. Now, we apply M-estimation method for model (4.3.5) which minimize

$$\sum_{i=1}^{N} \rho(\tilde{e}_{\mathbf{1}i}) = \sum_{i=1}^{N} \rho(y_{oi} - \mathbf{u}_i^T\boldsymbol{\vartheta}_{(\acute{r})}), \tag{4.3.6}$$

with respect to $\boldsymbol{\vartheta}_{(\acute{r})}$. To minimize Eq. (4.3.6), equate the first partial derivatives of ρ with respect to ϑ_j $(j = 1, \dots, \hat{p}_F)$ to zero. This gives the system of $\hat{p}_F$ equations

$$\sum_{i=1}^{N} \rho'(\tilde{e}_{\mathbf{1}i})\mathbf{u}_i^T = \mathbf{0}^T, \tag{4.3.7}$$

Then, we define the weight function

$$\tilde{w}(z) = \begin{cases} \rho'(z)/z & \text{if } z \neq 0, \\ 1 & \text{if } z = 0. \end{cases} \tag{4.3.8}$$

Now, Eq. (4.3.7) can be written as

$$\sum_{i=1}^{N} (\frac{\rho'(\tilde{e}_{\mathbf{1}i})}{\tilde{e}_{\mathbf{1}i}}) \tilde{e}_{\mathbf{1}i} \mathbf{u}_i^T = \sum_{i=1}^{N} \tilde{w}_i \tilde{e}_{\mathbf{1}i} \mathbf{u}_i^T = \mathbf{0}^T, \tag{4.3.9}$$

where $\tilde{w}_i = \tilde{w}(\tilde{e}_{\mathbf{1}i})$. Since $\tilde{e}_{\mathbf{1}i} = y_{oi} - \mathbf{u}_i^T \boldsymbol{\vartheta}_{(\acute{r})}$, we obtain

$$\sum_{i=1}^{N} \tilde{w}_i y_{oi} \mathbf{u}_i^T = \sum_{i=1}^{N} \tilde{w}_i \mathbf{u}_i^T \boldsymbol{\vartheta}_{(\acute{r})} \mathbf{u}_i^T. \tag{4.3.10}$$

In matrix form, Eq. (4.3.10) becomes

$$\mathbf{U}_{(\acute{r})}^T \tilde{\mathbf{W}} \mathbf{U}_{(\acute{r})} \boldsymbol{\vartheta}_{(\acute{r})} = \mathbf{U}_{(\acute{r})}^T \tilde{\mathbf{W}} \mathbf{y_o}, \tag{4.3.11}$$

where $\tilde{\mathbf{W}} = \text{diag}(\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_N)$. Let $\hat{\boldsymbol{\vartheta}}^*_{M(\acute{r})} = (\hat{\vartheta}^*_{M1} \quad \hat{\vartheta}^*_{M2} \quad \ldots \quad \hat{\vartheta}^*_{M\acute{r}})^T$ be the solution of Eq. (4.3.11). Hence, we have

$$\mathbf{U}_{(\acute{r})}^T \tilde{\mathbf{W}} \mathbf{U}_{(\acute{r})} \hat{\boldsymbol{\vartheta}}^*_{M(\acute{r})} = \mathbf{U}_{(\acute{r})}^T \tilde{\mathbf{W}} \mathbf{y_o}. \tag{4.3.12}$$

As mentioned before, the weights, however, depend upon the residuals, the residuals depend upon the estimated regression coefficients and the estimated regression coefficients depends upon the weights. Therefore, we use IRLS algorithm to obtain $\hat{\boldsymbol{\vartheta}}^*_{M(\acute{r})}$.

The *prediction of* $\mathbf{y}$ *with the first* $\acute{r}$ *vectors of* $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{\hat{p}_F}$, say $\tilde{\mathbf{y}}_M$, is given by

$$\tilde{\mathbf{y}}_M := \bar{y}\mathbf{1}_N + \mathbf{K}\boldsymbol{\Gamma}_{(\acute{r})}\hat{\boldsymbol{\vartheta}}^*_{(\acute{r})}. \tag{4.3.13}$$

The residual between $\mathbf{y}$ and $\tilde{\mathbf{y}}$ is given by

$$\hat{\tilde{\mathbf{e}}}_{\mathbf{1}} := \mathbf{y} - \tilde{\mathbf{y}}_M, \tag{4.3.14}$$

Then, the *prediction by the R-KPCR with the first* $\acute{r}$ *vectors of* $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{\hat{p}_F}$ is given by

$$g_{rkpcr(\acute{r})}(\mathbf{x}) := \bar{y} + \sum_{i=1}^{N} \tilde{c}_i \kappa(\mathbf{x}, \mathbf{x}_i), \tag{4.3.15}$$

where $g_{rkpcr(\acute{r})}$ is a function from $\mathbb{R}^p$ to $\mathbb{R}$, $\begin{pmatrix} \tilde{c}_1 & \tilde{c}_2 & \ldots & \tilde{c}_N \end{pmatrix}^T = \boldsymbol{\Gamma}_{(\acute{r})}\hat{\boldsymbol{\vartheta}}^*_{M(\acute{r})}$. The number $\acute{r}$ is called the *retained number of nonlinear PCs for the R-KPCR.*

4.3.2 R-KPCR's Algorithm

We summarize the procedure in 4.3.1 to obtain the prediction by R-KPCR.

Algorithm:

1. Given $(y_i, x_{i1}, x_{i2}, \ldots, x_{ip})$, $i = 1, 2, \ldots, N$.
2. Calculate $\bar{y} = \frac{1}{N}\mathbf{1}_N^T\mathbf{y}$ and $\boldsymbol{y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y}$.
3. Choose a kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ and $\rho : \mathbb{R} \to \mathbb{R}$
4. Construct $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{K} = (K_{ij})$.
5. Diagonalize $\mathbf{K}$.
   Let $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_{\acute{r}} \geq \tilde{\lambda}_{\acute{r}+1} \geq \ldots \geq \tilde{\lambda}_{\hat{p}_F} > \tilde{\lambda}_{\hat{p}_F+1} = \ldots =$

$\tilde{\lambda}_N = 0$ be the eigenvalues of $\mathbf{K}$ and $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N$ be the corresponding normalized eigenvectors of $\mathbf{K}$.

6. Construct $\boldsymbol{\alpha}_l = \frac{\mathbf{b}_l}{\sqrt{\tilde{\lambda}_l}}$ for $l = 1, 2, \ldots, \acute{r}$ and $\boldsymbol{\Gamma}_{(\acute{r})} = \begin{pmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \ldots & \boldsymbol{\alpha}_{\acute{r}} \end{pmatrix}$ where $\acute{r} \in \{1, 2, \ldots, \hat{p}_F\}$.

7. Calculate $\mathbf{U}_{(\acute{r})} = \mathbf{K}\boldsymbol{\Gamma}_{(\acute{r})}$.

8. Find estimator of $\boldsymbol{\vartheta}_{(\acute{r})}$ by IRLS

   (a) Select an initial estimator of $\boldsymbol{\vartheta}_{M(\acute{r})}$, say $\hat{\boldsymbol{\vartheta}}^{*(0)}_{M(\acute{r})}$, by OLS.

   (b) At each iteration t, calculate residual $\tilde{e}^{(t-1)}_{\mathbf{1}i} = y_{oi} - \mathbf{u}_i^T \hat{\boldsymbol{\vartheta}}^{*(t-1)}_{M(\acute{r})}$,

   $$\tilde{w}_i^{(t-1)} = \begin{cases} \frac{\rho'(\tilde{e}^{(t-1)}_{\mathbf{1}i})}{\tilde{e}^{(t-1)}_{\mathbf{1}i}} & \text{if } \tilde{e}^{(t-1)}_{\mathbf{1}i} \neq 0, \\ 1 & \text{if } \tilde{e}^{(t-1)}_{\mathbf{1}i} = 0, \end{cases}$$

   and $\tilde{\mathbf{W}}^{(t-1)} = \text{diag}(\tilde{w}_1^{(t-1)}, \tilde{w}_2^{(t-1)}, \ldots, \tilde{w}_N^{(t-1)})$.

   (c) Solve the new weighted least squares equations

   $$\mathbf{U}^T_{(\acute{r})} \tilde{\mathbf{W}}^{(t-1)} \mathbf{U}_{(\acute{r})} \hat{\boldsymbol{\vartheta}}^{*(t)}_{M(\acute{r})} = \mathbf{U}^T_{(\acute{r})} \tilde{\mathbf{W}}^{(t-1)} \mathbf{y_o}.$$

   Step (b) and (c) are repeated until the estimated regression coefficients converges. Let the estimated regression coefficients is convergence at $\hat{\boldsymbol{\vartheta}}^{*(\hat{t})}_{M(\acute{r})} = \begin{pmatrix} \hat{\vartheta}^{*(\hat{t})}_{M1} & \hat{\vartheta}^{*(\hat{t})}_{M2} & \ldots & \hat{\vartheta}^{*(\hat{t})}_{M\acute{r}} \end{pmatrix}^T$

9. Calculate $\tilde{\mathbf{c}} = \begin{pmatrix} \tilde{c}_1 & \tilde{c}_2 & \ldots & \tilde{c}_N \end{pmatrix}^T = \boldsymbol{\Gamma}_{(\acute{r})} \hat{\boldsymbol{\vartheta}}^{*(\hat{t})}_{M(\acute{r})}$.

10. Given a vector $\mathbf{x} \in \mathbb{R}^p$, the prediction by R-KPCR with the first $\acute{r}$ vectors of $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{\hat{p}_F}$ is given by

    $$g_{rkpc\acute{r}}(\mathbf{x}) = \bar{y} + \textstyle\sum_{j=1}^N \tilde{c}_i \kappa(\mathbf{x}, \mathbf{x}_j),$$

Note that the above algorithms work 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 have only to replace $\mathbf{K}$ by $\mathbf{K}_N$. Further, we diagonalize $\mathbf{K}_N$ in Step 5 and work based on $\mathbf{K}_N$ in the subsequent steps.

Chapter 5

Nonlinear Regressions Based on Ridge and Kernel Method

In this chapter, we consider some nonlinear methods based on ridge and kernel method. In Subchapter 5.1, we review kernel ridge regression (KRR) to overcome the limitations of RR. In Subchapter 5.2, we propose a combination of WLS and KRR to overcome the limitation of WLS method. Then, a combination of M-Estimation and KRR for dealing with outliers is presented in Subchapter 5.3.

5.1 Kernel Ridge Regression

Let us consider model (4.1.9) again. The ridge estimator of $\boldsymbol{\gamma}$ is found by solving the following problem [49]

$$\min \quad (\mathbf{Y}_o - \boldsymbol{\Psi}\boldsymbol{\gamma})^T(\mathbf{Y}_o - \boldsymbol{\Psi}\boldsymbol{\gamma}) + \tilde{c}\boldsymbol{\gamma}^T\boldsymbol{\gamma} \tag{5.1.1}$$

for some $\tilde{c} > 0$. Let $\hat{\boldsymbol{\gamma}}_R(\tilde{c})$ be the solution of problem (5.1.1). Then, we have

$$\hat{\boldsymbol{\gamma}}_R(\tilde{c}) = (\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \tilde{c}\mathbf{I}_{p_F})^{-1}\boldsymbol{\Psi}^T\mathbf{Y}_o. \tag{5.1.2}$$

Let $\hat{\boldsymbol{\gamma}}^*{}_R(c)$ be the value of $\hat{\boldsymbol{\gamma}}$ when $\mathbf{Y}_o$ is replaced by $\mathbf{y}_o$ in the Eq. (5.1.2). Since $\boldsymbol{\Psi}^T\mathbf{y}_o = \boldsymbol{\Psi}^T\mathbf{y}$ (see Appendix B.3), Eq.(5.1.2) becomes

$$\hat{\boldsymbol{\gamma}}^*_R(\tilde{c}) = (\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \tilde{c}\mathbf{I}_{p_F})^{-1}\boldsymbol{\Psi}^T\mathbf{y}. \tag{5.1.3}$$

Let us consider the following lemma.

Lemma 5.1.1. [49] $(\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \tilde{c}\mathbf{I}_{p_F})^{-1}\boldsymbol{\Psi}^T\mathbf{y} = \boldsymbol{\Psi}^T(\boldsymbol{\Psi}\boldsymbol{\Psi}^T + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y}$.

Proof. See Appendix B.4. □

By using Lemma 5.1.1, Eq. (5.1.3) can be written as

$$\hat{\boldsymbol{\gamma}}^*_R(\tilde{c}) = \boldsymbol{\Psi}^T(\boldsymbol{\Psi}\boldsymbol{\Psi}^T + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y}. \tag{5.1.4}$$

Then, the *prediction by KRR* is given by

$$\begin{aligned} g_{krr}(\mathbf{x}) &:= \bar{y} + \psi(\mathbf{x})^T\hat{\boldsymbol{\gamma}}^*_R(\tilde{c}) \\ &:= \bar{y} + \psi(\mathbf{x})^T\boldsymbol{\Psi}^T(\boldsymbol{\Psi}\boldsymbol{\Psi}^T + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y} \\ &:= \bar{y} + \psi(\mathbf{x})^T\boldsymbol{\Psi}^T(\mathbf{K} + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y} \end{aligned} \tag{5.1.5}$$

where g_{krr} is a function from $\mathbb{R}^p$ to $\mathbb{R}$.

As we see that the elements of $\psi(\mathbf{x})^T\boldsymbol{\Psi}^T = \begin{pmatrix}\psi(\mathbf{x})^T\psi(\mathbf{x}_1) & \dots & \psi(\mathbf{x})^T\psi(\mathbf{x}_N)\end{pmatrix}$ are provided implicitly by choosing a kernel function κ. Hence, Eq. (5.1.5) can be written as

$$g_{krr}(\mathbf{x}) = \bar{y} + \sum_{i=1}^{N} \acute{c}_i\kappa(\mathbf{x}, \mathbf{x}_i) \tag{5.1.6}$$

where $\acute{\mathbf{c}} = \begin{pmatrix}\acute{c}_1 & \acute{c}_2 & \dots & \acute{c}_N\end{pmatrix}^T = (\mathbf{K} + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y}$. The appropriate $\acute{c}$ of the prediction by KRR can be found by the cross validation method or other methods.

5.2 Weighted Least Squares in Kernel Ridge Regression

5.2.1 WLS-KRR

Let us consider the model (4.2.2) again. Here, we use the ridge regression to avoid the effects of multicollinearity in model (4.2.2). Hence, we solve

$$\min \quad (\mathbf{Z}_o - \boldsymbol{\theta}\boldsymbol{\gamma})^T(\mathbf{Z}_o - \boldsymbol{\theta}\boldsymbol{\gamma}) + \tilde{q}\boldsymbol{\gamma}^T\boldsymbol{\gamma}, \tag{5.2.1}$$

with respect to $\boldsymbol{\gamma}$ and for some $\tilde{q} > 0$. The solution of the problem (5.2.1) can be found by solving the following equations

$$(\boldsymbol{\theta}^T\boldsymbol{\theta} + \tilde{q}\mathbf{I}_{p_F})\boldsymbol{\gamma} = \boldsymbol{\theta}^T\mathbf{Z}_o. \tag{5.2.2}$$

It is evident that matrix $\boldsymbol{\theta}^T\boldsymbol{\theta} + \tilde{q}\mathbf{I}_{p_F}$ is invertible. Let $\breve{\boldsymbol{\gamma}}(\tilde{q})$ be the solution of the problem (5.2.1). Hence, we obtain

$$\breve{\boldsymbol{\gamma}}(\tilde{q}) = (\boldsymbol{\theta}^T\boldsymbol{\theta} + \tilde{q}\mathbf{I}_{p_F})^{-1}\boldsymbol{\theta}^T\mathbf{Z}_o. \tag{5.2.3}$$

Since $(\boldsymbol{\theta}^T\boldsymbol{\theta} + \tilde{q}\mathbf{I}_{p_F})^{-1}\boldsymbol{\theta}^T\mathbf{Z}_o = \boldsymbol{\theta}^T(\boldsymbol{\theta}\boldsymbol{\theta}^T + \tilde{q}\mathbf{I}_N)^{-1}\mathbf{Z}_o = \boldsymbol{\theta}^T(\breve{\mathbf{K}} + \tilde{q}\mathbf{I}_N)^{-1}\mathbf{Z}_o$, we obtain

$$\breve{\boldsymbol{\gamma}}(\tilde{q}) = \boldsymbol{\theta}^T(\breve{\mathbf{K}} + \tilde{q}\mathbf{I}_N)^{-1}\mathbf{Z}_o, \tag{5.2.4}$$

where $\breve{\mathbf{K}} = \breve{\mathbf{L}}^{-1}\mathbf{K}\breve{\mathbf{L}}^{-1}$ is known explicitly.

Let $\mathbf{z}_o = \breve{\mathbf{L}}^{-1}(I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y} \in \mathbb{R}^N$ be the observed data corresponding to $\mathbf{Z}_o$. Let $\breve{\boldsymbol{\gamma}}^*(q) = \begin{pmatrix}\breve{\gamma}_1^*(q) & \breve{\gamma}_1^*(q) & \dots & \breve{\gamma}_{p_F}^*(q)\end{pmatrix}^T \in \mathbb{R}^{p+1}$ be the value of $\breve{\boldsymbol{\gamma}}(\tilde{q})$ when $\mathbf{Z}_o$ is replaced by $\mathbf{z}_o$ in the Eq. (5.2.4). The *prediction value* of $\mathbf{z}_o$, say

$\hat{\mathbf{z}}_o$ $(= \tilde{\mathbf{L}}^{-1}(\hat{\mathbf{y}} - \bar{y}\mathbf{1}_N))$ is given by

$$\begin{aligned} \hat{\mathbf{z}}_{\mathbf{o}} &:= \begin{pmatrix} \hat{z}_{\mathbf{o}1} & \hat{z}_{\mathbf{o}2} & \dots & \hat{z}_{\mathbf{o}N} \end{pmatrix}^T \\ &= \boldsymbol{\theta}\breve{\boldsymbol{\gamma}}^*(q), \\ &= \breve{\mathbf{K}}(\breve{\mathbf{K}} + \tilde{q}\mathbf{I}_N)^{-1}\mathbf{z}_o. \end{aligned} \tag{5.2.5}$$

The residual between $\mathbf{z}_o$ and $\hat{\mathbf{z}}_o$ is given by

$$\mathbf{e_2} := \begin{pmatrix} e_{\mathbf{2}1} & e_{\mathbf{2}2} & \dots & e_{\mathbf{2}N} \end{pmatrix}^T = \mathbf{z}_o - \hat{\mathbf{z}}_o, \tag{5.2.6}$$

and the *prediction by the WLS KRR* is given by

$$g_{w-krr}(\mathbf{x}) := \bar{y} + \sum_{i=1}^{N} \acute{c}_i \kappa(\mathbf{x}, \mathbf{x}_i), \tag{5.2.7}$$

where g_{w-krr} is a function from $\mathbb{R}^p$ to $\mathbb{R}$ and $\begin{pmatrix} \acute{c}_1 & \acute{c}_2 & \dots & \acute{c}_N \end{pmatrix}^T = (\breve{\mathbf{K}}\breve{\mathbf{L}} + \tilde{q}\breve{\mathbf{L}})^{-1}\mathbf{z}_o$. The appropriate value of $\tilde{q}$ can be obtained by the cross validation method or other methods.

5.2.2 WLS-KRR's Algorithm

We summarize the procedure in Subsection 5.2.1 to obtain the prediction by WLS KRR.

Algorithm:

1. Given $(y_i, x_{i1}, x_{i2}, \dots, x_{ip})$, $i = 1, 2, \dots, N$.
2. Calculate $\bar{y} = \frac{1}{N}\mathbf{1}_N^T\mathbf{y}$ and $\boldsymbol{y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y}$.
3. Estimate $\breve{\mathbf{V}}$ and find $\breve{\mathbf{L}}$.
4. Calculate $\mathbf{z}_o = \breve{\mathbf{L}}^{-1}\mathbf{y}_o$.

5. Choose a kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ and a positive number $\tilde{q}$.

6. Construct $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{K} = (K_{ij})$.

7. Construct $\breve{\mathbf{K}} = \breve{\mathbf{L}}^{-1}\mathbf{K}\breve{\mathbf{L}}^{-1}$.

8. Calculate $\begin{pmatrix} \acute{c}_1 & \acute{c}_2 & \dots & \acute{c}_N \end{pmatrix}^T = (\breve{\mathbf{K}}\breve{\mathbf{L}} + \tilde{q}\breve{\mathbf{L}})^{-1}\mathbf{z}_o$.

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

$$g_{w-krr}(\mathbf{x}) = \bar{y} + \textstyle\sum_{j=1}^{N} \acute{c}_i \kappa(\mathbf{x}, \mathbf{x}_j).$$

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 have only to replace $\mathbf{K}$ by $\mathbf{K}_N$ in Step 7.

5.3 KRR and M-Estimation in Robust Regression Model

5.3.1 Robust Kernel Ridge Regression

Let us consider the model (4.3.2) again. We solve

$$\min \quad (\mathbf{y}_o - \mathbf{\Psi}\boldsymbol{\gamma})^T(\mathbf{y}_o - \mathbf{\Psi}\boldsymbol{\gamma}) + \acute{q}\boldsymbol{\gamma}^T\boldsymbol{\gamma}, \tag{5.3.1}$$

with respect to $\boldsymbol{\gamma}$ and for some $\acute{q} > 0$ if we use the ridge regression method to obtain the estimator of $\boldsymbol{\gamma}$. If we use the M-estimation method, the term $(\mathbf{y}_o - \mathbf{\Psi}\boldsymbol{\gamma})^T(\mathbf{y}_o - \mathbf{\Psi}\boldsymbol{\gamma}) = \sum_{i=1}^{N}(y_{oi} - \psi(\mathbf{x}_i))^2$ is replaced by $\sum_{i=1}^{N} \rho(y_{oi} - \psi(\mathbf{x}_i))$. Hence, we find the estimator of $\boldsymbol{\gamma}$ such that minimizes the function

$$\sum_{i=1}^{N} \rho(y_{oi} - \psi(\mathbf{x}_i)^T\boldsymbol{\gamma}) + \acute{q}\boldsymbol{\gamma}^T\boldsymbol{\gamma}. \tag{5.3.2}$$

To minimize Eq. (5.3.2), equate the first partial derivatives of ρ with respect to γ_j $(j = 1, \ldots, p_F)$ to zero. This gives the system of p_F equations

$$-\sum_{i=1}^{N} \rho'(\tilde{e}_i)\psi(\mathbf{x}_i)^T + 2\acute{q}\boldsymbol{\gamma}^T = \mathbf{0}^T, \tag{5.3.3}$$

where ρ' is the derivative of ρ. Then, we define the weight function

$$\breve{w}(z) = \begin{cases} \rho'(z)/z & \text{if } z \neq 0, \\ 1 & \text{if } z = 0. \end{cases} \tag{5.3.4}$$

Then, Eq. (5.3.3) can be written as

$$-\sum_{i=1}^{N} (\frac{\rho'(\tilde{e}_i)}{\tilde{e}_i})\tilde{e}_i\psi(\mathbf{x}_i)^T + 2\acute{q}\boldsymbol{\gamma}^T = -\sum_{i=1}^{N} \breve{w}_i\tilde{e}_i\psi(\mathbf{x}_i)^T + 2\acute{q}\boldsymbol{\gamma}^T = \mathbf{0}^T, \tag{5.3.5}$$

where $\breve{w}_i = \breve{w}(\tilde{e}_i)$. Since $\tilde{e}_i = y_{oi} - \psi(\mathbf{x}_i)^T\boldsymbol{\gamma}$, we obtain

$$\sum_{i=1}^{N} \breve{w}_i y_{oi}\psi(\mathbf{x}_i)^T = \sum_{i=1}^{N} \breve{w}_i\psi(\mathbf{x}_i)^T\boldsymbol{\gamma}\psi(\mathbf{x}_i)^T + 2\acute{q}\boldsymbol{\gamma}^T. \tag{5.3.6}$$

In matrix form, Eq. (5.3.6) becomes

$$(\boldsymbol{\Psi}^T\breve{\mathbf{W}}\boldsymbol{\Psi} + 2\acute{q}\mathbf{I}_{p_F})\boldsymbol{\gamma} = \boldsymbol{\Psi}^T\breve{\mathbf{W}}\mathbf{y_o}, \tag{5.3.7}$$

where $\breve{\mathbf{W}} = \text{diag}(\breve{w}_1, \breve{w}_2, \ldots, \breve{w}_N)$. Let $\hat{\boldsymbol{\gamma}}(\acute{q})$ be the solution of Eq. (5.3.6). Since $(\boldsymbol{\Psi}^T\breve{\mathbf{W}}\boldsymbol{\Psi} + 2\acute{q}\mathbf{I}_{p_F})$ is invertible, we have

$$\hat{\boldsymbol{\gamma}}(\acute{q}) = (\boldsymbol{\Psi}^T\breve{\mathbf{W}}\boldsymbol{\Psi} + 2\acute{q}\mathbf{I}_{p_F})^{-1}\boldsymbol{\Psi}^T\breve{\mathbf{W}}\mathbf{y_o}. \tag{5.3.8}$$

Let $\boldsymbol{\theta} = \breve{\mathbf{W}}^{1/2}\boldsymbol{\Psi}$ and $\mathbf{z}_o = \breve{\mathbf{W}}^{1/2}\mathbf{y_o}$. Hence, Eq. (5.3.8) can be written as

$$\hat{\boldsymbol{\gamma}}(\acute{q}) = (\boldsymbol{\theta}^T\boldsymbol{\theta} + 2\acute{q}\mathbf{I}_{p_F})^{-1}\boldsymbol{\theta}^T\mathbf{z_o}. \tag{5.3.9}$$

Since $(\boldsymbol{\theta}^T\boldsymbol{\theta} + 2\acute{q}\mathbf{I}_{p_F})^{-1}\boldsymbol{\theta}^T\mathbf{z}_o = \boldsymbol{\theta}^T(\boldsymbol{\theta}\boldsymbol{\theta}^T + 2\acute{q}\mathbf{I}_N)^{-1}\mathbf{z}_o$, we obtain

$$\begin{aligned}
\hat{\boldsymbol{\gamma}}(\acute{q}) &= \boldsymbol{\theta}^T(\boldsymbol{\theta}\boldsymbol{\theta}^T + 2\acute{q}\mathbf{I}_N)^{-1}\mathbf{z}_o \\
&= \boldsymbol{\Psi}^T\breve{\mathbf{W}}^{1/2}(\breve{\mathbf{W}}^{1/2}\boldsymbol{\Psi}\boldsymbol{\Psi}^T\breve{\mathbf{W}}^{1/2} + 2\acute{q}\mathbf{I}_N)^{-1}\breve{\mathbf{W}}^{1/2}\mathbf{y}_o \\
&= \boldsymbol{\Psi}^T\breve{\mathbf{W}}^{1/2}(\breve{\mathbf{W}}^{1/2}\mathbf{K}\breve{\mathbf{W}}^{1/2} + 2\acute{q}\mathbf{I}_N)^{-1}\breve{\mathbf{W}}^{1/2}\mathbf{y}_o. \qquad (5.3.10)
\end{aligned}$$

The *prediction of* $\mathbf{y}$, say $\tilde{\mathbf{y}}_{MR}$, is given by

$$\begin{aligned}
\tilde{\mathbf{y}}_{MR} &:= \bar{y}\mathbf{1}_N + \boldsymbol{\Psi}\hat{\boldsymbol{\gamma}}(\acute{q}) \\
&= \bar{y}\mathbf{1}_N + \boldsymbol{\Psi}\boldsymbol{\Psi}^T\breve{\mathbf{W}}^{1/2}(\breve{\mathbf{W}}^{1/2}\mathbf{K}\breve{\mathbf{W}}^{1/2} + 2\acute{q}\mathbf{I}_N)^{-1}\breve{\mathbf{W}}^{1/2}\mathbf{y}_o \\
&= \bar{y}\mathbf{1}_N + \mathbf{K}\breve{\mathbf{W}}^{1/2}(\breve{\mathbf{W}}^{1/2}\mathbf{K}\breve{\mathbf{W}}^{1/2} + 2\acute{q}\mathbf{I}_N)^{-1}\breve{\mathbf{W}}^{1/2}\mathbf{y}_o, \qquad (5.3.11)
\end{aligned}$$

where $\bar{y} = \frac{1}{N}\mathbf{1}_N^T\mathbf{y}$. The residual between $\mathbf{y}$ and $\tilde{\mathbf{y}}$ is given by

$$\hat{\tilde{\mathbf{e}}} := \mathbf{y} - \tilde{\mathbf{y}}_{MR}. \tag{5.3.12}$$

As mentioned before that the IRLS is a widely used technique in the robust method based on M-estimation. Unfortunately, we cannot use the IRLS to obtain the estimator of $\boldsymbol{\gamma}$ since we do not know $\boldsymbol{\Psi}$ explicitly. The estimator of $\boldsymbol{\gamma}$ is required to obtain the prediction by the R-KRR. Alternatively, we use the prediction value $\tilde{\mathbf{y}}$ to obtain the prediction by the R-KRR. As we see that prediction value $\tilde{\mathbf{y}}$ depends upon $\breve{\mathbf{W}}$, $\breve{\mathbf{W}}$ depends upon the residual, the residual depends on $\tilde{\mathbf{y}}$. An iterative solution to obtain $\tilde{\mathbf{y}}$ is therefore required. Furthermore, the *prediction by the R-KRR* is given by

$$g_{r-krr(\tilde{r})}(\mathbf{x}) := \bar{y} + \sum_{i=1}^{N} c_i\kappa(\mathbf{x}, \mathbf{x}_i), \tag{5.3.13}$$

where $g_{r-krr(\tilde{r})}$ is a function from $\mathbb{R}^p$ to $\mathbb{R}$ and

$$\begin{pmatrix} c_1 & c_2 & \dots & c_N \end{pmatrix}^T = \breve{\mathbf{W}}^{1/2}(\breve{\mathbf{W}}^{1/2}\mathbf{K}\breve{\mathbf{W}}^{1/2} + 2\acute{q}\mathbf{I}_N)^{-1}\breve{\mathbf{W}}^{1/2}\mathbf{y}_o.$$

5.3.2 R-KRR's Algorithm

We summarize the procedure in Subchapter 5.3.1 to obtain the prediction by R-KRR [52].

Algorithm:

1. Given $(y_i, x_{i1}, x_{i2}, \dots, x_{ip})$, $i = 1, 2, \dots, N$.
2. Calculate $\bar{y}$ and $\boldsymbol{y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{y}$.
3. Choose a kernel $\kappa : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, $\rho : \mathbb{R} \to \mathbb{R}$ and a positive number $\acute{q}$.
4. Construct $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{K} = (K_{ij})$.
5. Find the prediction value of $\mathbf{y}$:
   (a) Select an initial prediction value of $\mathbf{y}$, say $\tilde{\mathbf{y}}_{MR}^{(0)}$, by OLS.
   (b) At each iteration t, calculate residual $\hat{\tilde{\mathbf{e}}}^{(t-1)} = \mathbf{y} - \tilde{\mathbf{y}}^{(t-1)}$, $\hat{\tilde{e}}_i^{(t-1)} = (\hat{\tilde{\mathbf{e}}}^{(t-1)})_i$, $\breve{w}_i^{(t-1)} = \breve{w}(\hat{\tilde{e}}_i^{(t-1)})$, and $\breve{\mathbf{W}}_{(t-1)} = \mathrm{diag}(\breve{w}_1^{(t-1)}, \breve{w}_2^{(t-1)}, \dots, \breve{w}_N^{(t-1)})$.
   (c) Find the new prediction value

$$\tilde{\mathbf{y}}_{MR}^{(t)} = \bar{y}\mathbf{1}_N + \mathbf{K}\breve{\mathbf{W}}_{(t-1)}^{1/2}(\breve{\mathbf{W}}_{(t-1)}^{1/2}\mathbf{K}\breve{\mathbf{W}}_{(t-1)}^{1/2} + 2\acute{q}\mathbf{I}_N)^{-1}\breve{\mathbf{W}}_{(t-1)}^{1/2}\mathbf{y}_o.$$

   Step (b) and (c) are repeated until the prediction value converges. Let the prediction value is convergence at $\hat{t}$-th iteration.
6. Calculate:

$$\tilde{\mathbf{c}} = \begin{pmatrix} \tilde{c}_1 & \tilde{c}_2 & \dots & \tilde{c}_N \end{pmatrix}^T = \breve{\mathbf{W}}_{(\hat{t}-1)}^{1/2}(\breve{\mathbf{W}}_{(\hat{t}-1)}^{1/2}\mathbf{K}\breve{\mathbf{W}}_{(\hat{t}-1)}^{1/2} + 2\acute{q}\mathbf{I}_N)^{-1}\breve{\mathbf{W}}_{(\hat{t}-1)}^{1/2}\mathbf{y}_o.$$

7. Given a vector $\mathbf{x} \in \mathbb{R}^p$, the prediction by R-KRR is given by

$$g_{r-krr(\tilde{r})}(\mathbf{x}) = \bar{y} + \sum_{j=1}^{N} \tilde{c}_i \kappa(\mathbf{x}, \mathbf{x}_j),$$

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 have only to replace $\mathbf{K}$ by $\mathbf{K}_N$ throughout the steps of the algorithm.

Chapter 6

Case Studies

In this chapter, we present the performances of our proposed methods (See Table 1.1). We wrote the programs of our proposed methods by using Matlab R2007a. In Subchapter 6.1, we compare ordinary linear regression (OLR), principal component regression (PCR), ridge regression (RR), the revised KPCR, kernel ridge regression (KRR) and other nonlinear regressions. In Subchapter 6.2, we show that both of KPCR and KRR can be inappropriate to be used in regression model with variance of random errors having unequal values in diagonal elements, while both of WLS-KPCR and WLS-KRR give better results than that of WLS-LR, KPCR and KRR. In Subchapter 6.3, we present the comparisons between robust linear regression, KPCR, KRR, R-KPCR and R-KRR in regression model with the observation contaminated by outliers.

6.1 Case Studies for The Revised KPCR

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

6.1.1 The Household Consumption Data

As an illustration of the problem introduced by multicollinearity, we consider the household consumption data which are given in Table 6.1 [5]. The OLR of the household consumption data is given by

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + e_i \tag{6.1.1}$$

where y_i is the i-th household consumption expenditures, x_{i1} is the i-th household income and x_{i2} is the i-th household wealth. In this study, e_i is a real number generated by a normally distributed random noise with zero mean and standard deviation 0.01.

Table 6.1: The household consumption data.

y_i	70	65	90	95	110	115	120	140	155	150
x_{i1}	80	100	120	140	160	180	200	220	240	260
x_{i2}	810	1009	1273	1425	1633	1876	2052	2201	2435	2686

The prediction by OLR is given by

$$f_{olr}(x_1, x_2) = 24.7807 + 0.9359x_1 - 0.0419x_2. \tag{6.1.2}$$

The eigenvalues of $\mathbf{X}^T\mathbf{X}$ of the household consumption data are $\lambda_1 = 3.4032e+007$, $\lambda_2 = 6.7952e+001$ and $\lambda_3 = 1.0165$. We obtain $\frac{\lambda_1}{\lambda_1} = 1$, $\frac{\lambda_2}{\lambda_1} = 1.99667e-006$ and $\frac{\lambda_2}{\lambda_1} = 2.9868e-008$. Hence, multicollineary exists on the $\mathbf{X}$ of the household consumption data. The 95% confident interval of β_2 in Eq (6.1.2) is $[-0.2331, 0.1485]$ which contains include zero. It means that we cannot be confident whether x_2 makes contribution to Eq (6.1.2) or not.

Let us use the principal component regression (PCR) to the household consumption data. Note that the mean of y_i, x_{i1} and x_{i2} are 111, 170 and

1740, respectively. The matrix $\mathbf{Z}$ of the household consumption data is

$$\mathbf{Z} = \begin{pmatrix} -90 & 640 \\ -70 & 839 \\ -50 & 1103 \\ -30 & 1255 \\ -10 & 1463 \\ 10 & 1706 \\ 30 & 1882 \\ 50 & 2031 \\ 70 & 2265 \\ 90 & 2516 \end{pmatrix},$$

and the eigenvalues of $\mathbf{Z}$ are $\hat{\lambda}_1 = 3.8525e + 005$ and $\hat{\lambda}_2 = 7.5329$. Hence, we obtain $\frac{\hat{\lambda}_1}{\hat{\lambda}_1} = 1$, $\frac{\hat{\lambda}_2}{\hat{\lambda}_1} = 1.9553e - 005$. The normalized eigenvectors corresponding to $\hat{\lambda}_1 = 1.8621e + 003$ and $\hat{\lambda}_2 = 8.2338$ are $(0.09746 \quad 0.9952)^T$ and $(-0.9952 \quad 0.0975)^T$. As we see that multicollinearity exists on $\mathbf{Z}$. To avoid the effects of multicollinearity, PCR only uses the eigenvector corresponding to $\hat{\lambda}_1$. The prediction by principal component regression (PCR) is given by

$$\begin{aligned} f_{pcr}(x_1, x_2) &= 111 + (x_1 - 170 \quad x_2 - 1740)(0.09746 \quad 0.9952)^T 0.0496 \\ &= 111 + (x_1 - 170 \quad x_2 - 1740)(0.0048 \quad 0.0494)^T, \\ &= 24.2280 + 0.0048x_1 + 0.0494x_2. \end{aligned} \tag{6.1.3}$$

The 95% confident interval of β_1 in Eq (6.1.3) is $[0.0409, 0.0581]$. According to the t-test, we can be confident to accept the value 0.0496 as the estimator of β_1. It implies that x_1 and x_2 make contribution to Eq (6.1.3). When we use ridge regression, the prediction by RR with $c = 20$ is given by

$$f_{rr}(x_1, x_2) = 1.1945 + 0.6236x_1 + 0.0008x_2. \tag{6.1.4}$$

Let us now use the revised KPCR to the household consumption data.

Note that the eigenvalues of $\mathbf{K}$ of the household consumption data are $\tilde{\lambda}_1 = \tilde{\lambda}_2 = \ldots = \tilde{\lambda}_9 = 1.0000$ and $\tilde{\lambda}_{10} = 0.0000$. The prediction by the revised KPCR with the Gaussian kernel ($\varrho = 5$) is given by

$$\begin{aligned} g_{kpcr}(x_1, x_2) &= 111 - 40.9912\kappa((x_1, x_2), (80, 810)) - 46.001\kappa((x_1, x_2), (100, 1009)) \\ &\quad -21.0064\kappa((x_1, x_2), (120, 1273)) - 15.9753\kappa((x_1, x_2), (140, 1425)) \\ &\quad -0.9636\kappa((x_1, x_2), (160, 1633)) + 3.9886\kappa((x_1, x_2), (180, 1876)) \\ &\quad +9.0012\kappa((x_1, x_2), (200, 2052)) + 28.9829\kappa((x_1, x_2), (220, 2201)) \\ &\quad +44.0127\kappa((x_1, x_2), (240, 2435)) + 38.9518\kappa((x_1, x_2), (260, 2686)). \end{aligned} \tag{6.1.5}$$

The prediction by the revised KPCR only used the first nine nonlinear principal components corresponding to κ to avoid the effects of multicollinearity. The RMSE of OLR, PCR, RR and KPCR are 5.6960, 6.2008, 9.4442 and 0.0021, respectively.

Besides that, we can use the Akaike Information Criterion (AIC) to select the best model among the four models. Readers may consult other statistics books for the detailed discussion, see for example [41, 42]. The AIC of the linear model is given by

$$AIC = N \ln(2\pi\hat{\sigma}^2) + N + 2(p + 1) \tag{6.1.6}$$

where $\hat{\sigma}^2$ the estimator of σ^2 (See Appendix C). For example, the unbiased estimator of σ^2 in Eq. (6.1.1) is $\frac{1}{N}(\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}})$. A model is selected as the best model when it has the smallest AIC among other models. However, selection of the best model by using AIC can be inappropriate since AIC does not use a set of testing data in its calculation. The AIC of OLR, PCR, RR and the revised KPCR model are 72.7589, 71.1182, 109.3479 and -215.0172, respectively. According to those values, the revised KPCR model is the best for the household consumption data.

6.1.2 The Sinc Function

The toy data were constructed by the *sinc* function

$$f(x) = \begin{cases} \frac{|\sin(x)|}{|x|} & \text{for } x \neq 0, \\ 1 & \text{for } x = 0. \end{cases} \tag{6.1.7}$$

The toy data were constructed by the Eq. (6.1.7) with $x_{i1} = -10 + 0.2 \times (i-1), y_i = f(x_{i1}) + \acute{e}_i, i = 1, 2, \ldots, N$, where N is equal to 101, $\mathbf{y} = (y_1 \quad y_2 \quad \ldots \quad y_N)^T$ and $\mathbf{f} := (f(x_{11}) \quad f(x_{21}) \quad \ldots \quad f(x_{N1}))^T$. The number $\acute{e}_i$ is a real number generated by a random noise. We assume that the random noise is normally distributed with zero mean and standard deviation $\sigma_1 \in [0, 1]$. We also generated another set of data for the predictions by the linear regression, the Nadaraya-Watson regression and the revised KPCR. The set of data was also constructed by the Eq. (6.1.7) with $\hat{x}_{j1} = -10 + 0.25 \times (j-1), \hat{y}_j = f(\hat{x}_{j1}) + \hat{e}_j, j = 1, 2, \ldots, M$, where M is equal to 81, $\hat{\mathbf{y}} := (\hat{y}_1 \quad \hat{y}_2 \quad \ldots \quad \hat{y}_M)^T$ and $\hat{\mathbf{f}} := (f(\hat{x}_{11}) \quad f(\hat{x}_{21}) \quad \ldots \quad f(\hat{x}_{M1}))^T$. The number $\hat{e}_j$ is a real number generated by a normally distributed random noise with zero mean and standard deviation $\sigma_2 \in [0, 1]$. The set of the data (y_i, x_{i1}) and set of the data $(\hat{y}_j, \hat{x}_{j1})$ are called the *training data set* and the *testing data set*, respectively.

To test the performance of the three methods, we generated Q sets of the training data and the testing data. Call the set of $(y_i^{(k)}, x_{i1})$ and the set of $(\hat{y}_j^{(k)}, \hat{x}_{j1})$, $k = 1, 2, \ldots, Q$, the kth training data set and the kth testing data set, respectively, where $y_i^{(k)} = f(x_{i1}) + \acute{e}_i^{(k)}$, $\hat{y}_j^{(k)} = f(\hat{x}_{j1}) + \hat{e}_j^{(k)}$, $\acute{e}_i^{(k)}$ is a real number generated by a normally distributed random noise with zero mean and standard deviation σ_1, and $\hat{e}_j^{(k)}$ is a real number generated by a normally distributed random noise with zero mean and standard deviation σ_2. Let $\acute{\mathbf{y}}^{(k)}$ and $\acute{\hat{\mathbf{y}}}^{(\mathbf{k})}$ be the predictions of $\mathbf{f}$ and $\hat{\mathbf{f}}$ corresponding to the kth training data set and the kth testing data set by the revised KPCR, respectively. The RMSE for the kth training data set and for the kth testing data set by the revised KPCR are given by $\frac{1}{\sqrt{N}}\|\acute{\mathbf{y}}^{(k)} - \mathbf{f}\|$ and $\frac{1}{\sqrt{M}}\|\acute{\hat{\mathbf{y}}}^{(k)} - \hat{\mathbf{f}}\|$, respectively. The *mean of RMSEs* (*MRMSE*) for the training data sets and for the testing data

Table 6.2: The comparison of the linear regression, Nadaraya-Watson regression and the revised KPCR for the Sinc function data (N-W: Nadaraya-Watson, ♯: N-W with the Bowman's and Azzalini's method, §:N-W with the Silverman's method).

Noise σ_1 (σ_2)	MRMSE				
	OLR	N-W	Revised KPCR	Kernel	$\tilde{r}$
0	0.3516	0.0222$^\sharp$	5.0679e-004	Gaussian	34
(0.5)	(0.3513)	(0.0222)$^\sharp$	(5.1257e-004)	$\varrho = 1$	
0.02	0.3516	0.0221$^\sharp$	0.0133		
(0.5)	(0.3513)	(0.0221)$^\sharp$	(0.0134)		
0.2	0.3535	0.0598$^\sharp$	0.0967		
(0.5)	(0.3532)	(0.0598)$^\sharp$	(0.0966)		
0		0.2662§	3.6712e-004	Gaussian	16
(0.5)		(0.2824)§	(3.7534e-004)	$\varrho = 5$	
0.02		0.2628§	0.0084		
(0.5)		(0.2790)§	(0.0085)		
0.2		0.2885§	0.0835		
(0.5)		(0.3079)§	(0.0835)		
0			1.6912e-004	Gaussian	11
(0.5)			(1.7086e-004)	$\varrho = 10$	
0.02			0.0085		
(0.5)			(0.0085)		
0.2			0.0511		
(0.5)			(0.0513)		
0.02			0.3134	Polynomial	1
(0.5)			(0.3131)	$d = 2$	
0.02			0.3364	Polynomial	1
(0.5)			(0.3360)	$d = 4$	
0.02			0.1066	Sigmoid	8
(0.5)			(0.6895)	$r_1 = 2$,$r_2 = 2$	
				$\theta = 0.1$	
0.02			0.1116	Sigmoid	9
(0.5)			(0.6337)	$r_1 = 2$,$r_2 = 4$	
				$\theta = 0.1$	

sets are given by $\frac{1}{Q\sqrt{N}}\sum_{k=1}^{Q}\|\acute{\mathbf{y}}^{(k)}-\mathbf{f}\|$ and $\frac{1}{Q\sqrt{M}}\sum_{k=1}^{Q}\|\acute{\hat{\mathbf{y}}}^{(k)}-\hat{\mathbf{f}}\|$, respectively. In this study we set Q to 1000. The RMSEs (MRMSEs) for the linear regression and the Nadaraya-Watson regression are calculated in the same manner.

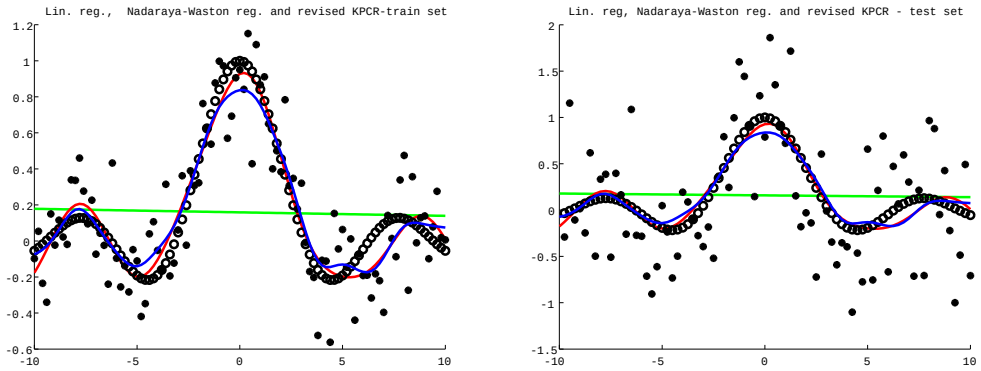


Figure 6.1: The linear regression (green), Nadaraya-Watson Regression (blue, $\hat{h}_{1ba} = 0.6987$) and the revised KPCR (red and $\tilde{r} = 11$) by applying the Gaussian kernel with $\varrho = 10$ for the first toy data. The black circles are the original training (testing) data. The black dots are the original training (testing) data by adding the random noise. The standard deviation of the noise for the training data is 0.2 and for the testing data is 0.5: (a) training data (b) testing data.

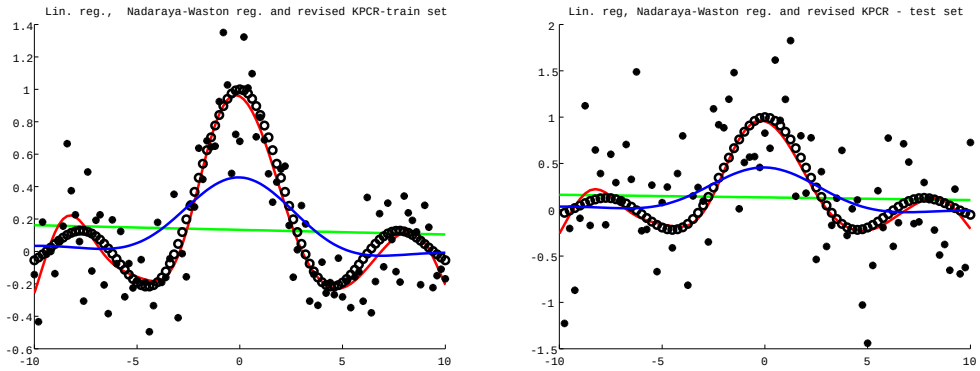
To estimate the Nadaraya-Watson regression, we used the Matlab program which was construted by Yi Cao [6]. In this program, Yi Cao used the function

$$\kappa_1(u) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}u^2)$$

and h_1 is estimated by the Bowman's and Azzalini's method [4]. The estimator of h_1 by Bowman's and Azzalini's method, say $\hat{h}_{1ba}$, is given by

$$\hat{h}_{1ba} = \sqrt{h_{1x}h_{1y}},$$

where $h_{1x} = \frac{1}{0.6745}\text{median}(|\text{x}_1-\text{median}(\text{x}_1)|)(\frac{4}{3\text{N}})^{0.2}$ and $h_{1y} = \frac{1}{0.6745}\text{median}(|\text{y}-$



(a) (b)

Figure 6.2: The linear regression (green), Nadaraya-Watson Regression (blue, $\hat{h}_{1s} = 2.4680$) and the revised KPCR (red and $\tilde{r} = 11$) by applying the Gaussian kernel with $\varrho = 10$ for the first toy data. The standard deviation of the noise for the training data is 0.2 and for the testing data is 0.5: (a) training data (b) testing data.

median(y)|)$(\frac{4}{3N})^{0.2}$. Another choice of the estimator of h_1 is

$$\hat{h}_{1s} = 1.06\hat{s}N^{-1/5},$$

where $\hat{s} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1} - \bar{x}_1)}$ and $\bar{x}_1 = \frac{1}{N}\sum_{i=1}^{N} x_{i1}$. We refer to $\hat{h}_{1s}$ as the estimator of h_1 by Silverman's method [16].

The comparison of the three methods is shown in Table 6.2, where the standard deviations and the MRMSEs of the training data sets are represented without parentheses while the standard deviations and the MRMSEs of the testing data sets are represented with parentheses. According to this study, the revised KPCR together with the Gaussian kernel provides the small enough MRMSEs. Two plots of the prediction of linear regression, the Nadaraya-Watson regression and the revised KPCR for the toy data are given in Figure 6.1 and Figure 6.2.

Table 6.3: Growth of the Son of the Count de Montheillard

Age (yr, mth [day]	Height (cm)	Age (yr, mth [day]	Height (cm)
0	51.4	9,0	137.0
0,6	65.0	9,7[12]	140.1
1,0	73.1	10,0	141.6
1,6	81.2	11,6	141.9
2,0	90.0	12,0	149.9
2,6	92.8	12,8	154.1
3,0	98.8	13,0	155.3
3,6	100.4	13,6	158.6
4,0	105.2	14,0	162.9
4,7	109.5	14,6[10]	169.2
5,0	111.7	15,0[2]	175.0
5,7	111.7	15,6[8]	177.5
6,0	117.8	16,3[8]	181.4
6,6[19]	122.9	16,6[6]	183.3
7,0	124.3	17,0[2]	184.6
7,3	127.0	17,1[9]	185.4
7,6	128.9	17,5[5]	186.5
8,0	130.8	17,7[4]	186.8
8,6	134.3		

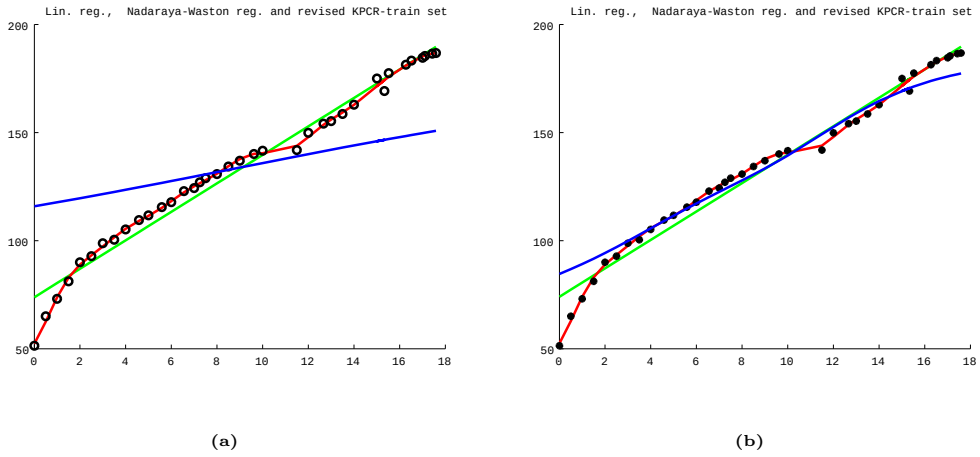


Figure 6.3: The linear regression (green), Nadaraya-Watson Regression (blue) and the revised KPCR (red, $\varrho = 5$ and $\tilde{r} = 19$) for the growth of the son of the Count de Montbeillard. The black circles are the given data: (a) Nadaraya-Watson Regression with $\hat{h}_{1ba} = 9.1208$ (b) Nadaraya-Watson Regression with $\hat{h}_{1s} = 2.8747$.

6.1.3 Growth of the Son of the Count de Montbeillard

We use a subset of the famous set of observation taken on the height of the son of the Count de Montbeillard between 1959 and 1977. Only the first ten years of data were used in this analysis. The growth of the son of the Count de Montbeillard data are given in the Table 6.3 [42]. A plot of the prediction of linear regression, Nadaraya-Watson regression and the revised KPCR for this data is given in Figure 6.3. The comparison of the linear regression, the revised KPCR, KRR and Nadaraya-Watson regression are shown in Table 6.4.

6.1.4 The Puromycin Data

In this case study, we want to predict the reaction velocity and substrate concentration for the puromycin data. The reaction velocity (y) and substrate concentration for puromycin (x) are given in the Table 6.5 [41]. A plot of the prediction of linear regression, Nadaraya-Watson regression and the

Table 6.4: The comparison of the linear regression, the revised KPCR, KRR, and N-W regression (N-W: Nadaraya-Watson, ♯: N-W with the Bowman's and Azzalini's method, §:N-W with the Silverman's method).

Data	Model	RMSE
The son of the Count de Montbeillard	Linear regression	5.8055
	The revised KPCR ($\varrho = 5, \tilde{r} = 14$)	1.3856
	KRR($\varrho = 5$, $\tilde{c} = 0.1$)	3.8689
	N-W regression$^{\sharp}$	25.9011
	N-W regression§	8.5353
The puromycin data	Linear regression	28.2062
	The revised KPCR ($\varrho = 1, \tilde{r} = 3$)	10.7231
	KRR($\varrho = 5$, $\tilde{c} = 0.00001$)	65.8873
	N-W regression$^{\sharp}$	49.7743
	N-W regression§	28.9019
The radioactive tracer data	Linear regression	0.0991
	The revised KPCR ($\varrho = 5, \tilde{r} = 8$)	0.0002
	KRR($\varrho = 5$, $\tilde{c} = 1$)	0.0004
	N-W regression$^{\sharp}$	0.0086
	N-W regression§	0.1182

Table 6.5: The Puromycin Data

i	x_i	y_i
1	0.02	76
2	0.02	47
3	0.02	97
4	0.06	107
5	0.11	123
6	0.11	139
7	0.22	159
8	0.22	152
9	0.56	191
10	0.56	201
11	1.10	207
12	1.10	200

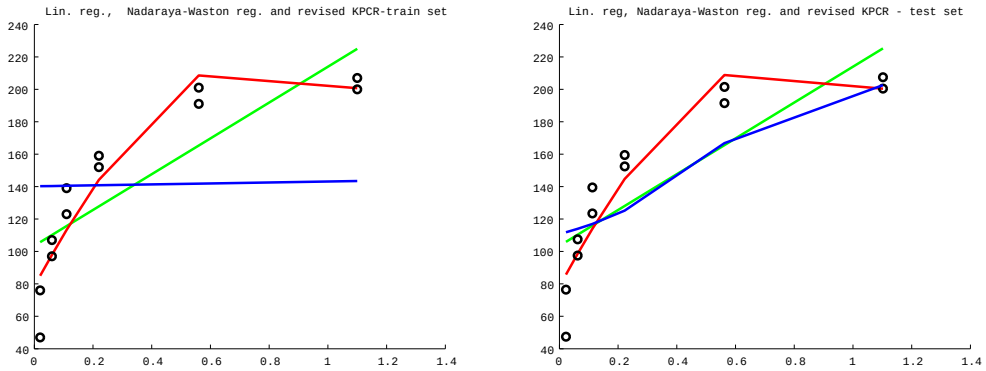


Figure 6.4: The linear regression (green), Nadaraya-Watson Regression (blue) and the revised KPCR (red, $\varrho = 5$ and $\tilde{r} = 19$) for the puromycin data. The black circles are the given data: (a) Nadaraya-Watson Regression with $\hat{h}_{1ba} = 2.3170$ (b) Nadaraya-Watson Regression with $\hat{h}_{1s} = 0.2571$.

revised KPCR for this data is given in Figure 6.4. The comparison of the linear regression, the revised KPCR, KRR and Nadaraya-Watson regression are shown in Table 6.4.

6.1.5 The Radioactive Tracer Data

In this case study, we consider the radioactive tracer data. The radioactive tracer data are given in the Table 6.6 [42]. A plot of the prediction of linear regression, Nadaraya-Watson regression and the revised KPCR for this data is given in Figure 6.5. The comparison of the linear regression, the revised KPCR, KRR and Nadaraya-Watson regression are shown in Table 6.4.

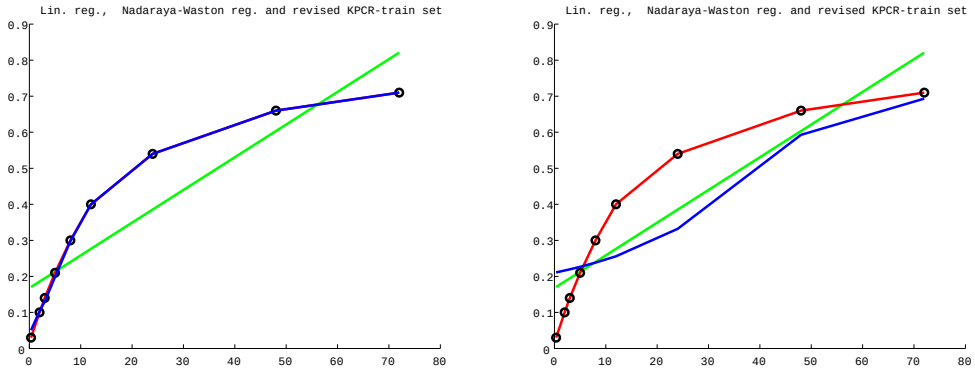
6.1.6 The Linear Distributed Data

In this case study, we observed the toy data which were constructed by the linear function

$$f(x) = 3 + 2x \qquad x \in [-1, 2]. \tag{6.1.8}$$

Table 6.6: Radioactive Tracer Data.

i	x_i (hours)	y_i
1	0.33	0.03
2	2	0.01
3	3	0.14
4	5	0.21
5	8	0.30
6	12	0.40
7	24	0.54
8	48	0.66
9	72	0.71



(a) (b)

Figure 6.5: The linear regression (green), Nadaraya-Watson Regression (blue) and the revised KPCR (red, $\varrho = 5$ and $\tilde{r} = 19$) for the radioactive tracer data. The black circles are the given data: (a) Nadaraya-Watson Regression with $\hat{h}_{1ba} = 9.1208$ (b) Nadaraya-Watson Regression with $\hat{h}_{1s} = 1.1079$.

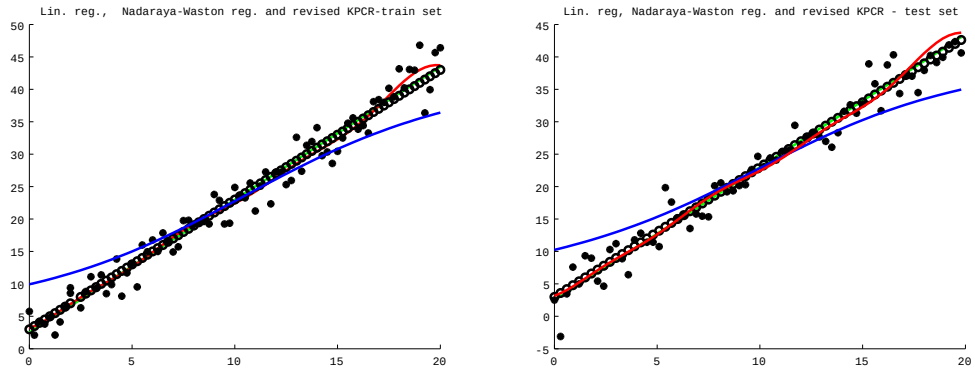


Figure 6.6: The linear regression (green), Nadaraya-Watson Regression (blue, $\hat{h}_{1ba} = 4.5724$) and the revised KPCR (red and $\tilde{r} = 8$) by applying the Gaussian kernel with $\varrho = 20$ for the second toy data. The black circles are the original training (testing) data. The black dots are the original training (testing) data by adding the random noise. The standard deviation of the noise for the training data is 2 and for the testing data is 2: (a) training data (b) testing data.

The training data were constructed by the Eq. (6.1.8) with $x_{i1} = 0.25 \times (i - 1), y_i = f(x_{i1}) + \acute{e}_i, i = 1, 2, \ldots, 81$. The testing data were also constructed by the Eq. (6.1.8) with $\hat{x}_{j1} = 0.3 \times (j-1), \hat{y}_j = f(\hat{x}_{j1}) + \hat{e}_j, j = 1, 2, \ldots, 67$. The comparison of the linear regression, the revised KPCR and the Nadaraya-Watson regression are shown in Table 6.7. A plot of the prediction of linear regression, the Nadaraya-Watson regression and the revised KPCR for this toy data is given in Figure 6.6.

6.1.7 The Cars and Chickens Data

In this case study, we used the stock of cars in the Netherlands (period 1965-1989) and the weight of a certain kind of female chickens in [23]. The data of the stock of cars and the weight of female chickens are shown in the Table 6.8 and Table 6.9, respectively; and used the Gaussian kernel for the revised KPCR. Jukic *et al.* [23] used the Gompertz function given below to obtain

Table 6.7: The comparison of the linear regression, Nadaraya-Watson regression and the revised KPCR for the linear distributed data (N-W: Nadaraya-Watson, ♯: N-W with the Bowman's and Azzalini's method, §:N-W with the Silverman's method).

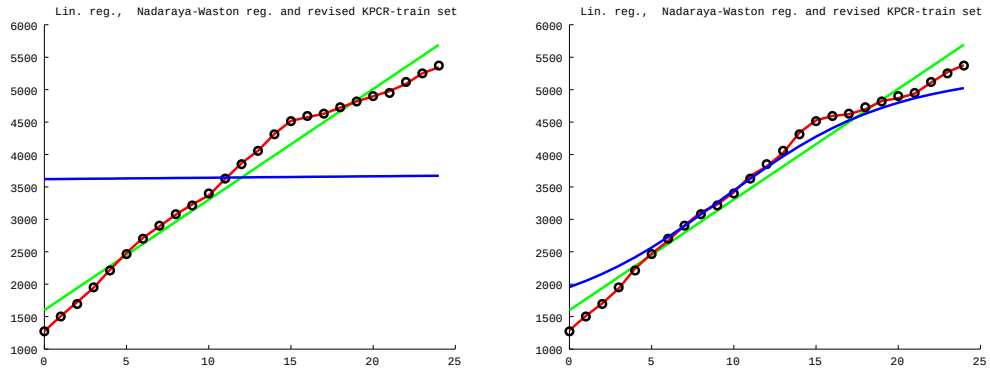
Noise σ_1 (σ_2)	MRMSE				
	OLR	N-W	Revised KPCR	Kernel	$\tilde{r}$
0	9.0152e-015	3.5066$^\sharp$	0.0526	Gaussian	34
(0.5)	(8.89042e-015)	(3.4591)$^\sharp$	(0.0529)	$\varrho = 1$	
0.02	3.4740e-003	3.5134$^\sharp$	0.0540		
(0.5)	(3.4282e-003)	(3.4659)$^\sharp$	(0.0553)		
0.2	0.0138	3.5953$^\sharp$	0.1342		
(0.5)	(0.0139)	(3.5485)$^\sharp$	(0.1366)		
0		0.2662§	0.0788	Gaussian	11
(0.5)		(0.2824)§	(0.0754)	$\varrho = 10$	
0.02		0.2704§	0.0792		
(0.5)		(0.2872)§	(0.0758)		
0.2		0.2870§	0.1034		
(0.5)		(0.3027)§	(0.1039)		

Table 6.8: The stock of cars (expressed in Thousands) in the Netherlands (period 1965-1989, x_{i1} is year - 1965 and y_i represents the stock of cars.)

x_{i1}	0	1	2	3	4	5	6	7	8
y_i	1273	1502	1696	1952	2212	2465	2702	2903	3080
x_{i1}	9	10	11	12	13	14	15	16	17
y_i	3214	3399	3629	3851	4056	4312	4515	4594	4630
x_{i1}	18	19	20	21	22	23	24		
y_i	4728	4818	4901	4950	5118	5251	5371		

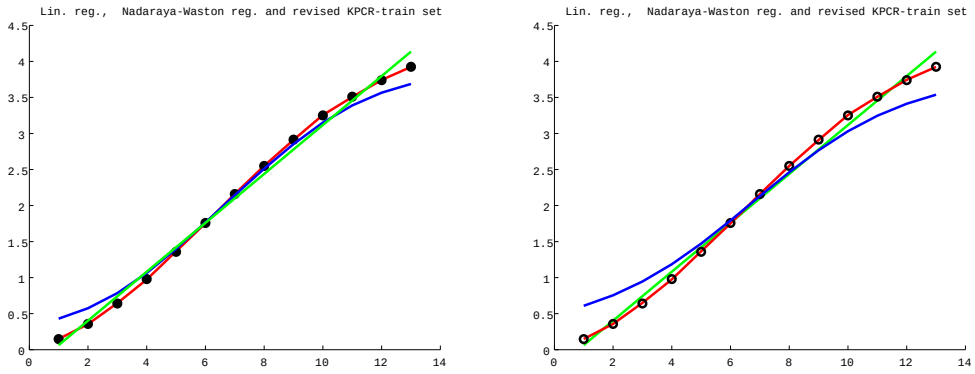
Table 6.9: The weight of a certain kind of female chickens observed once a week (x_{i1} in week and y_i in kg).

x_{i1}	1	2	3	4	5	6	7	8	9
y_i	0.147	0.357	0.641	0.980	1.358	1.758	2.159	2.549	2.915
x_{i1}	10	11	12	13					
y_i	3.251	3.510	3.740	3.925					



(a) (b)

Figure 6.7: The linear regression (green), Nadaraya-Watson Regression (blue) and the revised KPCR (red, $\varrho = 5$ and $\tilde{r} = 19$) for the stock of cars in Netherland. The black circles are the given data: (a) Nadaraya-Watson Regression with $\hat{h}_{1ba} = 62.8357$ (b) Nadaraya-Watson Regression with $\hat{h}_{1s} = 4.0981$.



(a) (b)

Figure 6.8: The linear regression (green), Nadaraya-Watson Regression (blue) and the revised KPCR (red, $\varrho = 5$ and $\tilde{r} = 10$) for the weight of female chickens. The black circles are the given data: (a) Nadaraya-Watson Regression with $\hat{h}_{1ba} = 1.7682$ (b) Nadaraya-Watson Regression with $\hat{h}_{1s} = 2.4715$.

the nonlinear regressions for these real data:

$$q(x, a, b, c) = \exp(a - b\exp(-cx)), \qquad b, c > 0, a \in \mathbb{R}, \tag{6.1.9}$$

We refer to the nonlinear regression proposed by Jukic *et al.* as the *Jukic's regression*. In this study, we compare the performance of the linear regression, the revised KPCR, KRR Nadaraya-Watson regression and the Jukic's regression for these real data.

The RMSEs by the revised KPCR for the stock of cars and the weight of female chickens are 1.9623e-012 and 6.8926e-016, respectively, when ϱ is equal to one. However, the predictions by the revised KPCR seem to be an overfitting, i.e., it provides the very small RMSE for the given data, but the testing data do not obtain so small RMSE as the given data did. According to our case studies, the overfitting can be avoided by setting ϱ to five. The comparison of the linear regression, the revised KPCR, KRR, Nadaraya-Watson regression and the Jukic's regression are shown in Table 6.10. We see that the RMSEs by the revised KPCR are smaller than that of the linear regression, KRR, Nadaraya-Watson regression and the Jukic's regression. A plot of the prediction of linear regression, Nadaraya-Watson regression and the revised KPCR for the stock of cars data is given in Figure 6.7, while a plot of the prediction of linear regression, Nadaraya-Watson regression and the revised KPCR for the weight of female chickens data is given in Figure 6.8.

6.2 Case Study for WLS-KPCR and WLS-KRR

There are several methods to estimate the weight w_i [30, 14, 27, 41]. Here, we use the method based on replication to estimate the weight w_i. First, we arrange the data x in order of increasing y_i. Then, we make several groups, say M ($< N$) groups, of the ordered data. Let the kth group, $k = 1, 2, \ldots, M$,

Table 6.10: The comparison of the linear regression, the revised KPCR, KRR, N-W regression and the Jukic's regression (N-W: Nadaraya-Watson, ♯: N-W with the Bowman's and Azzalini's method, §:N-W with the Silverman's method).

Data	Model	RMSE
The stock of cars	Linear regression	205.8677
	The revised KPCR ($\varrho = 5, \tilde{r} = 19$)	7.8016
	KRR($\varrho = 5$, $\tilde{c} = 10^3$)	29.4920
	The Jukic's regression	63.2097
	N-W regression$^{\sharp}$	1.2359e+003
	N-W regression§	246.5681
The weight of female chickens	Linear regression	0.1023
	The revised KPCR ($\varrho = 5, \tilde{r} = 10$)	0.0040
	KRR($\varrho = 5$, $\tilde{c} = 1$)	0.1230
	The Jukic's regression	0.0141
	N-W regression$^{\sharp}$	0.1452
	N-W regression§	0.2662

Table 6.11: The restaurant foods sales data ($y_i \times 100$)

Obs.	1	2	3	4	5	6	7	8
x	3.00	3.150	3.085	5.225	5.350	6.090	8.925	9.015
y	81.464	72.661	72.344	90.743	98.588	96.507	126.574	114.133

9	10	11	12	13	14	15	16
8.885	8.950 9.00	11.345	12.275	12.400	12.525	12.310	
115.814	123.181	131.434	140.564	151.352	146.426	130.963	144.630

17	18	19	20	21	22	23	24
13.700	15.000	15.175	14.995	15.050	15.200	15.150	16.800
147.041	179.021	166.200	180.732	178.187	185.304	155.931	172.579

25	26	27	28	29	30
16.500	17.830	19.500	19.200	19.000	19.350
188.851	192.424	203.112	192.482	218.715	214.317

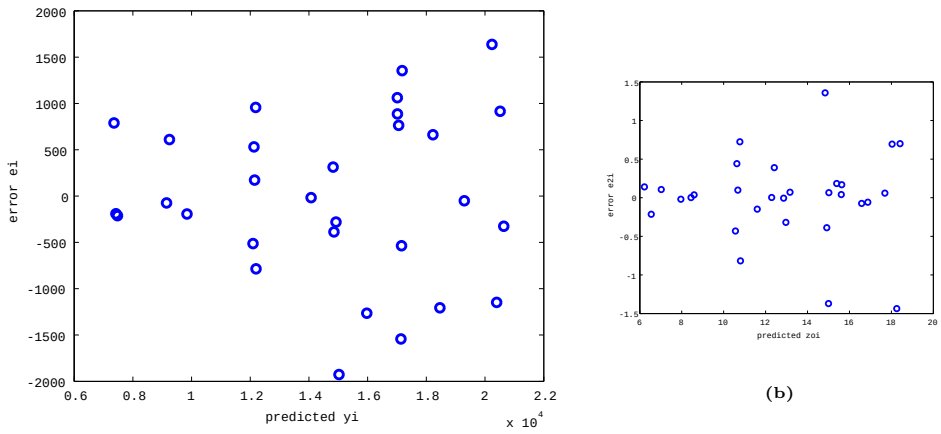


Figure 6.9: A plot of the residual and its corresponding predicted value for training data: (a) ordinary linear regression model, (b) WLS KPCR.

contains $\{(\acute{y}_{\hat{i}k}, \acute{x}_{\hat{i}k})\}$ for some $\hat{i} = 1, 2, \ldots, N$ where $\acute{y}_{\hat{i}k} \in \{y_1, y_2, \ldots, y_N\}$ and $\acute{x}_{\hat{i}k} \in \{x_1, x_2, \ldots, x_N\}$. Let $\bar{\acute{x}}_k$ and s_k^2 be the average of $\{\acute{x}_{\hat{i}k}\}$ and variance of $\{\acute{y}_{\hat{i}k}\}$, respectively. Then, we make the prediction from the set $\{(\bar{\acute{x}}_k, s_k^2)\}$, say $f_1(x) = \hat{c}_0 + \hat{c}_1 x$, where f_1 is a function from $\mathbb{R}$ to $\mathbb{R}$ and $\hat{c}_0, \hat{c}_1 \in \mathbb{R}$. Further, we calculate the estimated variance of y_i by using the predictor $f_1(x_i)$. The weight w_i is chosen inversely from $f_1(x_i)$. When $f_1(x_i)$ is equal to zero, w_i is set to be one. The procedure to obtain the WLS KPCR's weights and the WLS KRR's weights are straightforward as the explained procedure. We just replace y_i by y_{oi} where y_{oi} is the ith element of $\mathbf{y}_o$.

In this case study, we use the Gaussian kernel and the average monthly income from food sales (y) and the corresponding annual advertising expenses (x) for 30 restaurants which are given in Table 6.11 [27]. We use some of the data to test the prediction by the ordinary linear regression, WLS regression, KPCR and WLS KPCR. Note that the plot of the residual e_i and the corresponding $\hat{y}_i$ is useful to check the assumption of constant variance. The plot of e_i and $\hat{y}_i$ is shown in Figure 6.9 (a). Figure 6.9 (a) shows that the variation of the residuals increases significantly as the prediction values increase. Hence, the assumption of constant variance is not met.

Table 6.12: The RMSE of OLR, WLS-LR, KPCR, KRR, WLS-KPCR and WLS KRR for the restaurant foods sales data.

	Model	RMSE		
		M=2	M=4	M=5
Training data	ordinary linear regression	869.8845	869.8845	869.8845
	KPCR ($\varrho = 0.5$, $\tilde{r}$= 18)	601.3838	601.3838	601.3838
	KPCR ($\varrho = 1$, $\tilde{r}$= 17)	624.2270	624.2270	624.2270
	KRR ($\varrho = 0.5$, $\tilde{c} = 10^{-8}$)	29729.3585	29729.3585	29729.3585
	KRR ($\varrho = 1$, $\tilde{c} = 10^{-8}$)	33302.2060	33302.2060	33302.2060
	WLS linear regression	0.3834	0.5471	0.6958
	WLS KPCR ($\varrho = 0.5$, $\breve{r}$= 18)	0.2769	0.4178	0.5258
	WLS KPCR ($\varrho = 1$, $\breve{r}$=17)	0.2874	0.4336	0.5457
	WLS-KRR ($\varrho = 0.5$, $\tilde{c} = 10^{-8}$)	0.2976	0.4279	0.5275
	WLS-KRR ($\varrho = 1$, $\tilde{c}$= 17)	0.3095	0.4436	0.5465
Testing data	ordinary linear regression	834.9586	834.9586	834.9586
	KPCR ($\varrho = 0.5$, $\tilde{r}$=18)	689.8944	689.8944	689.8944
	KPCR ($\varrho = 1$, $\tilde{r}$=17)	721.4072	721.4072	721.4072
	KRR ($\varrho = 0.5$, $\tilde{c} = 10^{-8}$)	29617.2494	29617.2440	29617.2494
	KRR ($\varrho = 1$, $\tilde{c} = 10^{-8}$)	28415.9128	28415.9128	28415.9128
	WLS linear regression	0.5380	1.2599	0.5639
	WLS KPCR ($\varrho = 0.5$, $\breve{r}$=18)	0.3973	0.7893	0.4314
	WLS KPCR ($\varrho = 1$, $\breve{r}$=17)	0.4229	0.8984	0.4573
	WLS-KRR ($\varrho = 0.5$, $\tilde{c} = 10^{-8}$)	0.4499	0.9033	0.4432
	WLS-KRR ($\varrho = 1$, $\tilde{c} = 10^{-8}$)	0.4648	0.9739	0.4605

We can also see that the residual e_i has a relatively large number. This implies that $RMSE_{olr}$ is also a large number. For the sake of comparison, the values of M are chosen to be two, four and five. For instance $M = 2$, it means that the ordered data is divided into two groups where each group contains 50 percentage of the ordered data. The plot of residual $e_{\mathbf{2}i}$ and its corresponding prediction value $\hat{z}_{\mathbf{o}i}$ with $M = 2$ and $\varrho = 1$ is shown in Figure 6.9 (b). In comparison to the plot in Figure 6.9 (a), it is much more improved since the residual $e_{\mathbf{2}i}$ has a much smaller number than e_i. Beside that, Figure 6.9 (b) shows a residual plot with no systematic pattern around zero. It seems that the assumption of constant variance is satisfied for the data.

The results of this study are given in Table 6.12. Note that, multicollinearity exists in the regression matrix for both of the ordinary linear regression model and the WLS regression model. In Table 6.12, we can see that the WLS KPCR and WLS KRR significantly decreases the RMSEs of OLR, KPCR and KRR.

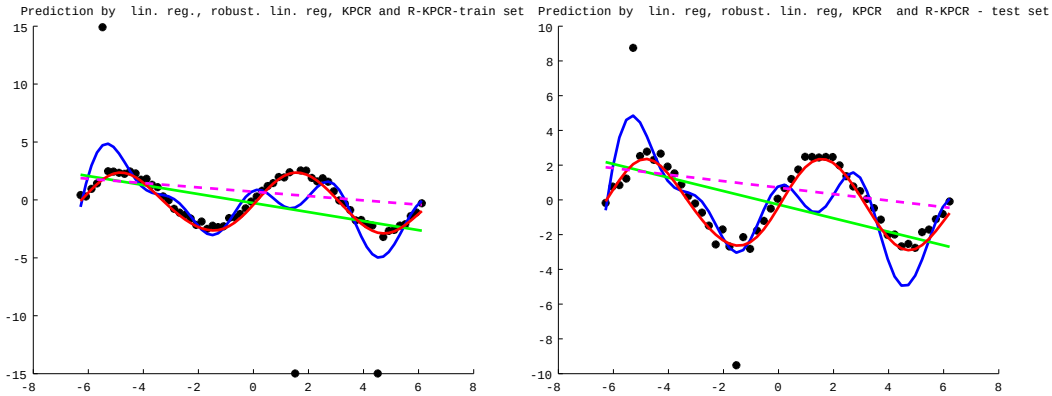
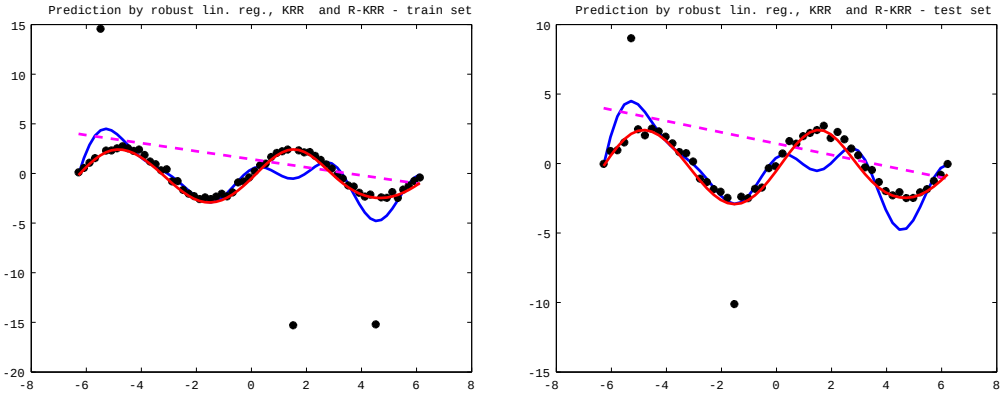


Figure 6.10: A plot of predictions for the linear regression (Green), robust linear regression (Magenta-dash line), KPCR (Blue) and R-KPCR (Red) with ϱ and $\tilde{r}$ equal to 5 and 10, respectively. The robust regression methods used the Huber function with k is equal to 2. The black dots are the toy data by adding the random noise: (a) training data, (b) testing data.



(a) (b)

Figure 6.11: A plot of predictions for the robust linear regression (Magenta-dash line), KRR (Blue) and R-KRR (Red) with ϱ and $\tilde{q}$ are equal to 2.5 and 0.1, respectively. The robust regression methods used the Huber function with k is equal to 2. The black dots are the toy data with random noise: (a) training data, (b) testing data.

6.3 Case Study for R-KPCR and R-KRR

6.3.1 The Sine Function with Outliers

In this case study, we use the Gaussian kernel and the toy data constructed by the function

$$f(x) = 2.5 \sin x, \tag{6.3.1}$$

with $x_{i1} = -2\pi + 0.2 \times i$ for $i = 0, 1, \ldots, 62$; and

$$y_i = \begin{cases} f(x_{i1}) + \acute{e}_i & \text{if } i \in \{0, 1, \ldots, 62\} \setminus \{5, 40, 55\}, \\ 15 + \acute{e}_5 & \text{if } i = 5, \\ -15 + \acute{e}_{40} & \text{if } i = 40, \\ -15 + \acute{e}_{55} & \text{if } i = 55. \end{cases} \tag{6.3.2}$$

where $\acute{e}_i, \acute{e}_5, \acute{e}_{40}$ and $\acute{e}_{55}$ are real numbers generated by a normally distributed random noise with zero mean and standard deviation $\sigma_1 \in [0, 1]$. The set of the data (y_i, x_{i1}) is used as the training data set. Here, y_5, y_{40} and y_{55} are the outliers of the training data. We also generated another set of data for the predictions by robust linear regression, KPCR, KRR, R-KPCR and R-KRR. It was also constructed by the Eq. (6.3.1) with $\check{x}_{j1} = -2\pi + 0.25 \times j$ for $j = 0, 1, \ldots, 50$; and

$$\check{y}_j = \begin{cases} f(\check{x}_{j1}) + \check{e}_j & \text{if } j \in \{0, 1, \ldots, 50\} \setminus \{5, 20\}, \\ 9 + \check{e}_5 & \text{if } j = 5, \\ -10 + \check{e}_{20} & \text{if } j = 20, \end{cases} \tag{6.3.3}$$

where $\check{e}_j, \check{e}_5$ and $\check{e}_{20}$ are also real numbers generated by a normally distributed random noise with zero mean and standard deviation $\sigma_2 \in [0, 1]$. The set of the data $(\check{y}_j, \check{x}_{j1})$ is used as the testing data set. Here, $\check{y}_5$ and $\check{y}_{20}$ are the outliers of the testing data.

Then, we generated 1000 sets of the training data and the testing data to test the performance of the five methods. For the sake of comparison, we set σ_1 and σ_2 are equal to 0.2 and 0.25, respectively. A plot of the predictions of the robust linear regression, KPCR and R-KPCR corresponding to the toy data are shown in Figure 6.10, while a plot of the predictions of the robust linear regression, KRR and R-KRR corresponding to the toy data are shown in Figure 6.11. The averages of RMSEs of the five methods are given in Table 6.13. Compared to robust linear regression, KPCR and KRR; R-KPCR and R-KRR yield the better results as shown in Table 6.13.

Table 6.13: Comparison of the robust linear regression, KPCR, KRR, R-KPCR and R-KRR.

	Model	RMSE	
		Training	Testing
Tukey biweighted function	robust linear regression	0.5331	0.5306
	KPCR ($\rho = 1, \hat{r} = 21$)	2.5968	2.0875
	KPCR ($\rho = 2.5, \hat{r} = 14$)	2.8100	1.9047
	KRR ($\varrho = 1, \tilde{q} = 0.1$)	7.1038	3.4783
	KRR ($\varrho = 2.5, \tilde{q} = 0.1$)	8.0810	3.0135
	R-KPCR ($\varrho = 1, \acute{r} = 21$)	0.1461	0.1518
	R-KPCR ($\varrho = 2.5, \acute{r} = 14$)	0.1627	0.1523
	R-KRR ($\varrho = 1, \tilde{q} = 0.1$)	0.1782	0.1617
	R-KRR ($\varrho = 1, \tilde{q} = 0.5$)	0.1801	0.1641
	R-KRR ($\varrho = 2.5, \tilde{q} = 0.1$)	0.1782	0.1617
	R-KRR ($\varrho = 2.5, \tilde{q} = 0.5$)	0.1795	0.1631
Huber function	robust linear regression	1.6909	2.0350
	KPCR ($\rho = 1, \hat{r} = 21$)	2.5960	2.0821
	KPCR ($\rho = 2.5, \hat{r} = 14$)	2.7875	1.8995
	KRR ($\varrho = 1, \tilde{q} = 0.1$)	2.6624	1.8932
	KRR ($\varrho = 2.5, \tilde{q} = 0.1$)	2.8427	1.7359
	R-KPCR ($\varrho = 1, \acute{r} = 21$)	1.1763	0.7520
	R-KPCR ($\varrho = 2.5, \acute{r} = 14$)	1.1540	0.7217
	R-KRR ($\varrho = 1, \tilde{q} = 0.1$)	1.2088	0.7246
	R-KRR ($\varrho = 1, \tilde{q} = 0.5$)	1.1692	0.7634
	R-KRR ($\varrho = 2.5, \tilde{q} = 0.1$)	1.3207	0.7250
	R-KRR ($\varrho = 2.5, \tilde{q} = 0.5$)	1.2015	0.7635

6.3.2 The Sinc Function with Outliers

In this case study, we fit the toy data constructed by the Sinc function Eq. (6.1.7) with $x_{i1} = -7 + 0.2 \times i$ for $i = 0, 1, \ldots, 70$; and

$$y_i = \begin{cases} f(x_{i1}) + \acute{e}_i & \text{if } i \in \{0, 1, \ldots, 70\} \setminus \{5, 20\}, \\ 13 + \acute{e}_5 & \text{if } i = 5, \\ -14 + \acute{e}_{20} & \text{if } i = 20. \end{cases} \tag{6.3.4}$$

where $\acute{e}_i, \acute{e}_5$ and $\acute{e}_{20}$ are real numbers generated by a normally distributed random noise with zero mean and standard deviation $\sigma_1 \in [0, 1]$. The set of the data (y_i, x_{i1}) is used as the training data set. Here, y_5 and y_{20} are the outliers of the training data. The testing set is constructed by the Eq. (6.3.1) with $\check{x}_{j1} = -5\pi + 0.25 \times j$ for $j = 0, 1, \ldots, 40$; and

$$\check{y}_j = \begin{cases} f(\check{x}_{j1}) + \check{e}_j & \text{if } j \in \{0, 1, \ldots, 40\} \setminus \{8, 30\}, \\ 15 + \check{e}_8 & \text{if } j = 8, \\ -24 + \check{e}_{30} & \text{if } j = 30, \end{cases} \tag{6.3.5}$$

where $\check{e}_j, \check{e}_8$ and $\check{e}_{30}$ are also real numbers generated by a normally distributed random noise with zero mean and standard deviation $\sigma_2 \in [0, 1]$. Here, $\check{y}_5$ and $\check{y}_{20}$ are the outliers of the testing data.

We also generated 1000 sets of the training data and the testing data to test the performance of the five methods. For the sake of comparison, we also set σ_1 and σ_2 to 0.2 and 0.25, respectively. A plot of the predictions of the robust linear regression, KPCR and R-KPCR corresponding to the toy data are shown in Figure 6.12, while a plot of the predictions of the robust linear regression, KRR and R-KRR corresponding to the toy data are shown in Figure 6.13. The averages of RMSEs of the five methods are given in Table 6.14. Compared to robust linear regression, KPCR and KRR; R-KPCR and R-KRR also yield the better results as shown in Table 6.14.

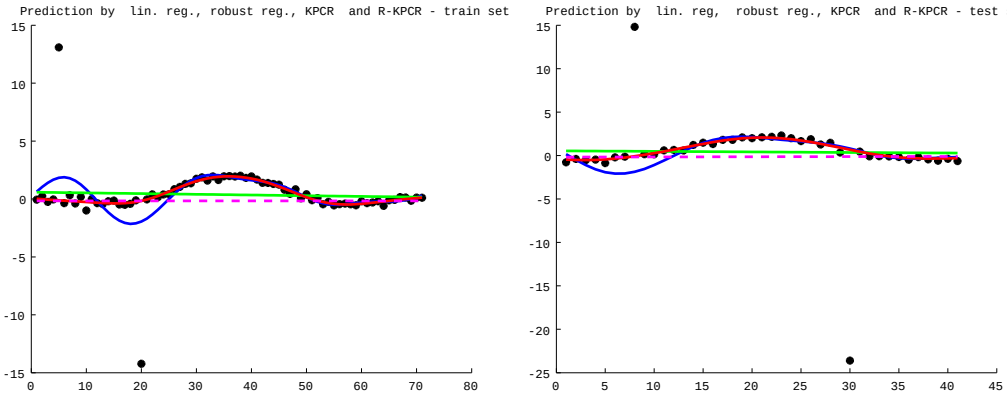


Figure 6.12: A plot of predictions for the linear regression (Green), robust linear regression (Magenta-dash line), KPCR (Blue) and R-KPCR (Red) with ϱ and $\tilde{r}$ equal to 5 and 10, respectively. The robust regression methods used the Huber function with k is equal to 2. The black dots are the toy data by adding the random noise: (a) training data, (b) testing data.

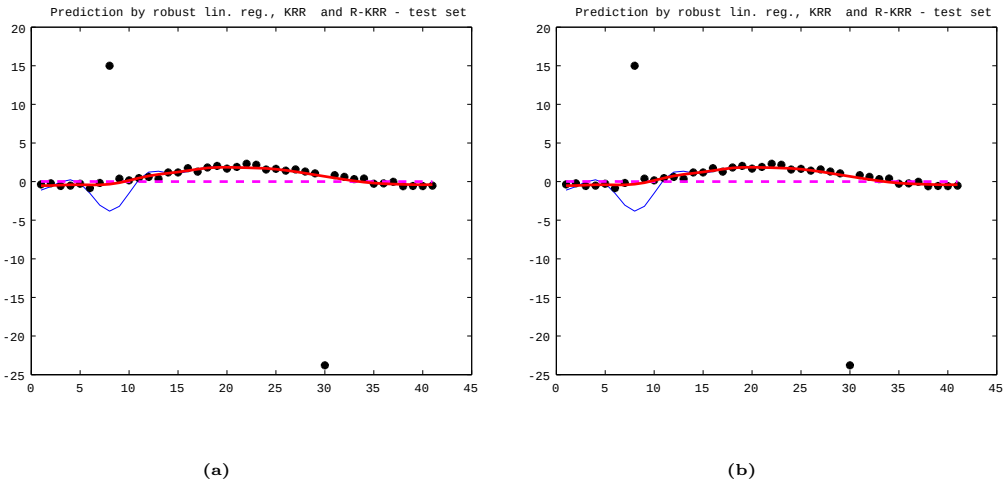


Figure 6.13: A plot of predictions for the robust linear regression (Magenta-dash line), KRR (Blue) and R-KRR (Red) with ϱ and $\tilde{q}$ are equal to 5 and 0.1, respectively. The robust regression methods used the Huber function with k is equal to 2. The black dots are the toy data with random noise: (a) training data, (b) testing data.

Table 6.14: Comparison of the robust linear regression, KPCR, KRR, R-KPCR and R-KRR.

	Model	RMSE	
		Training	Testing
Tukey biweighted function	robust linear regression	0.2383	0.2830
	KPCR ($\rho = 1, \hat{r} = 24$)	1.8669	4.9884
	KPCR ($\rho = 2.5, \hat{r} = 18$)	1.9676	4.8796
	KRR ($\varrho = 1, \tilde{q} = 0.1$)	3.6380	23.9891
	KRR ($\varrho = 2.5, \tilde{q} = 0.1$)	4.0974	22.5408
	R-KPCR ($\varrho = 1, \acute{r} = 24$)	0.0686	0.0901
	R-KPCR ($\varrho = 2.5, \acute{r} = 14$)	0.0685	0.0901
	R-KRR ($\varrho = 1, \tilde{q} = 0.1$)	0.1370	0.1803
	R-KRR ($\varrho = 1, \tilde{q} = 0.5$)	0.1370	0.1803
	R-KRR ($\varrho = 2.5, \tilde{q} = 0.1$)	0.1370	0.1803
	R-KRR ($\varrho = 2.5, \tilde{q} = 0.5$)	0.1370	0.1803
Huber function	robust linear regression	1.0202	1.4842
	KPCR ($\rho = 1, \hat{r} = 24$)	1.8642	5.0004
	KPCR ($\rho = 2.5, \hat{r} = 16$)	1.9660	4.8587
	KRR ($\varrho = 1, \tilde{q} = 0.1$)	3.6297	24.0610
	KRR ($\varrho = 2.5, \tilde{q} = 0.1$)	4.0933	22.6309
	R-KPCR ($\varrho = 1, \acute{r} = 24$)	0.8539	1.3466
	R-KPCR ($\varrho = 2.5, \acute{r} = 16$)	0.8459	1.3530
	R-KRR ($\varrho = 1, \tilde{q} = 0.1$)	1.0506	1.3533
	R-KRR ($\varrho = 1, \tilde{q} = 0.5$)	1.0246	1.3538
	R-KRR ($\varrho = 2.5, \tilde{q} = 0.1$)	1.0603	1.3523
	R-KRR ($\varrho = 2.5, \tilde{q} = 0.5$)	1.0263	1.355

Chapter 7

Conclusions

7.1 Conclusions

KPCR is a novel method to perform a nonlinear regression analysis. However, the previous KPCR still has theoretical difficulties in the procedure to derive the KPCR and in its choice rule of the retained number of PCs. In this dissertation, we revised the procedure of the previous KPCR and showed that the difficulties are eliminated by our revised KPCR. Regarding our case studies, the revised KPCR together with the Gaussian kernel provides the small enough RMSEs. The revised KPCR with the Gaussian kernel gives the better results than that of the ordinary linear regression and the Jukic's regression for the given data which are nonlinearly distributed. The revised KPCR with an appropriate parameter of the Gaussian kernel also gives better results than the Nadaraya-Watson regression.

In some cases, however, we face the regression model with variance of random errors having unequal values in diagonal elements. WLS is widely used to handle the limitations. However, applying WLS yields a linear prediction model and there is no guarantee that the effects of multicollinearity can be avoided by applying this method. Although KPCR and KRR can be used to handle the limitations of the linearity and the effect of multicollinearity, but KPCR and KRR can still be inappropriate since KPCR and KRR were constructed by the assumption that the variance of random errors having equal values in its diagonal elements. In this dissertation, we proposed WLS

KPCR and WLS KRR for the regression model with unequal variance of random errors. These methods yield nonlinear prediction model and can avoid the effects of multicollinearity. In our case study, the WLS KPCR and WLS KRR yield the better results that of the OLR, WLS-LR, KPCR and KRR.

If the outliers are contained in the observed data, the predictions of OLR, KPCR and KRR can also be inappropriate to be used. Fomengko *et al.* [13] proposed a nonlinear robust prediction based on the M-estimation where their method needs a specific nonlinear regression model in advance. In many situations, however, an appropriate nonlinear regression model for a set of data is unknown in advance. Hence, their method has limitations in applications. In this dissertation, we proposed R-KPCR and R-KRR to obtain a nonlinear robust prediction where our proposed method does not need to specify a nonlinear model in advance. Our case studies showed that both of R-KPCR and R-KRR yield the better results than that of the robust linear regression, KPCR and KRR.

Bibliography

[1] Howard Anton, *Elementary linear algebra*, John Wiley and Sons, Inc., 2000.

[2] Steven F. Arnold, *Mathematical statistics*, The Prentice Hall., 1990.

[3] David E. Booth and Kidong Lee, *Robust regression-based analysis of drug nucleic acid binding*, Analytical Biochemistry **319** (2003).

[4] Andrian W. Bowman and Adelchi Azzalini, *Applied smoothing techniques for data analysis: The kernel approach with s-plus illustrations*, Oxford Statistical Science Series, 1997.

[5] Bryan W. Brown, *Econometrics*, Departement of Economics, Rice University, http://www.ruf.rice.edu/ bwbwn/, 2008.

[6] Yi Cao, *A non-parametrical regression (smoothing) tool using gaussian kernel*, http://www.mathworks.com/matlabcentral/fileexchange/19195, 2004.

[7] R.J. Caroll and D. Ruppert, *Transformation and weighting in regression*, Chapman and Hall, 1988.

[8] J. Cho, J. Lee, S.W Choi, D. Lee, and I. Lee, *Fault identification for process monitoring using kernel principal component analysis*, Chemical Engineering Science (2005), 279–288.

[9] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machine*, Cambridge Univ. Press, 2000.

[10] K.I. Diamantaras and S.Y. Kung, *Principal component neural networks: Theory and applications*, John Wiley and Sons, Inc., 1996.

[11] D. Dong and T.J. McAvov, *Nonlinear component analysis-based on principal curves and neural networks*, Chemical Engineering Sciences **20** (1996), 65–78.

[12] W. DuMouchel and F. O'Brien, *Integrating a robust option into a multiple regression computing environment*, Computing Science and Statistics: Proceedings of the 21st Symposium on the Interface, American Statistical Association, Alexandria, VA, 1989, pp. 297.301 (1990).

[13] I. Famenko, M. Durst, and D. Balaban, *Robust regression for high throughput drug screening*, Computer Methods and Program in Biomedicine **82** (2006).

[14] Julian J. Faraway, *Linear models with r*, Chapman and Hall/CRC, 2005.

[15] Stephen .H. Friedberg, Arnold J. Insel, and Lawrence E. Spence, *Linear algebra*, Prentice-Hall, 2003.

[16] W. Hardle, M. Muller, S. Sperlich, and A. Werwatz, *Nonparametric and semiparametric models*, Springer, 2004.

[17] David A. Harville, *Matrix algebra from a statistician's perspective*, Springer, 1997.

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

[19] P. Huber, *Robust statistics*, John Wiley and Son Inc, 1981.

[20] 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** (2003), 4441–4448.

[21] Richard A. Johson and Dean W. Wichern, *Applied multivariate statistical analysis*, The Prentice Hall., 1982.

[22] I.T. Jolliffe, *Principal component analysis*, Springer, 2002.

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

[24] C. Lu, C. Zhang, T. Zhang, and W. Zhang, *Kernel based symmerical principal component analysis for face classification*, Neurocomputing **70** (2007), 904–911.

[25] William Mendenhall, Dennis D. Wackerly, and Richard L. Sheaffer, *Mathematical statistics with applications*, PWS-Kent Publishing Company, 1990.

[26] 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).

[27] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining, *Introduction to linear regression*, Wiley-Interscience, 2006.

[28] E.A. Nadaraya, *On estimating regression*, Theory of Probability and its Applications **10** (1964), 186–190.

[29] ———, *Nonparametric estimation of probabilty density and regression curve*, Kluwer Academic Publisher, 1989.

[30] Draper R. Norman and Harry Smith, *Applied regression analysis*, John Wiley and Sons, 1998.

[31] Eubank L. Randall, *Nonparametric regression and spline smootihing*, Marcel Dekker, Inc., 1999.

[32] D. Rocke, *Constructive statistics: estimators, algorithms, and asymptotics*, Comput. Sci. Stat. **30** (1998).

[33] 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 (2001), 231–243.

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

[35] 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).

[36] R. Saegusa, H. Sakano, and S. Hashimoto, *Nonlinear principal component analysis to preserve the order of principal components*, Neurocomputing **61** (2004), 57–70.

[37] C. Saunders, A. Gammerman, and V. Vovk, *Ridge regression learning algorithm in dual variables*, Proceedings of the 15th International Conference on Machine Learning, ICML'98. (1998).

[38] Michael G. Schimek, *Smoothing and regression: Approaches, computation, and application.*, John Wiley and Sons, 2000.

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

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

[41] George A.F. Seber and Alan J. Lee, *Linear regression analysis*, John Wiley and Sons, Inc., 2003.

[42] George A.F. Seber and C.J. Wild, *Nonlinear regression*, John Wiley and Sons, Inc., 1998.

[43] Larry Smith, *Linear algebra*, Springer-Verlag, New York, 1998.

[44] M.S. Srivastava, *Methods of multivariate statistics*, John Wiley and Sons, Inc., 2002.

[45] I. Stanimirova, M. Daszykowski, and B. Walczak, *Dealing with missing values and outliers in principal components analysis*, Chemometrics and Intelligent Laboratory Systems **72** (2007), 172–178.

[46] G. Strang, *Introduction to linear algebra*, Wellesley-Cambridge Press, 2003.

[47] V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag, New York, 1995.

[48] G.S. Watson, *Smooth regression analysis*, Sankhy Series, 1964.

[49] Max Welling, *Robust prediction in kernel ridge regression based on m-estimation*, Departement of Computer Science, University of Toronto, Canada.

[50] Antoni Wibowo, *An algorithm for nonlinear weighted least squares regression*, Discussion Paper Series No. 1217, Department of Social Systems and Management, Univ. of Tsukuba (2008).

[51] ———, *Robust prediction in kernel principal component regression based on m-estimation*, http://www.sk.tsukuba.ac.jp/SSM/libraries/list1201.html, Department of Social Systems and Management, Univ. of Tsukuba (2008).

[52] ———, *Robust prediction in kernel ridge regression based on m-estimation*, To be appear in Computational Mathematics and Modeling, Springer USA **21** (2010).

[53] Antoni Wibowo and Yoshitsugu Yamamoto, *The new approach for kernel principal component regression*, Discussion Paper Series No. 1195, Department of Social Systems and Management, Univ. of Tsukuba (2008).

[54] R. Wolke and H. Schwetlick, *Iteratively reweighted least squares: Algorithms, convergence analysis and numerical comparisons*, SIAM Journal of Sci. Stat. Comput. **9** (1988).

[55] W. Wu, D.L. Massart, and S. de Jong, *The kernel pca algorihms for wide data part i: theory and algorithms*, Chemometrics and Intelligent Systems **36** (1997), 165–172.

[56] Shie Mannor Yakoov Engel and Ron Meir, *The kernel recursive least-squares algorithm*, IEEE Transaction on Signal Processing **52**.

APPENDIX

A Review of Linear Algebra and Random Vectors

This appendix summarizes some basic concepts from linear and matrix algebra that are related to some important statistics concepts with emphasis on random variable, expected value and random vectors (matrices). Readers may consult other linear algebra, matrices and statistics books for the detailed discussion, see for example [1, 2, 15, 21, 25, 43, 46].

A.1 Eigenvalue and Eigenvector

Definition A.1. Let $\mathbf{A}$ be an $p \times p$ square matrix. A nonzero vector $\mathbf{x}$ in $\mathbb{R}^p$ is called an *eigenvector* of $\mathbf{A}$ if $\mathbf{Ax}$ is a scalar multiple of $\mathbf{x}$, that is

$$\mathbf{Ax} = \lambda\mathbf{x}$$

for some scalar λ. The scalar λ is called an eigenvalue of $\mathbf{A}$, and $\mathbf{x}$ is said to be an eigenvector of $\mathbf{A}$ corresponding to λ.

A.2 Orthogonal Projection

Definition A.2. Let $\mathcal{W}$ be a finite-dimensional subspace of an inner product space $\mathcal{V}$. A vector $\mathbf{u}$ in $\mathcal{V}$ is said to be *orthogonal to* $\mathcal{W}$ if it orthogonal to

every vector in $\mathcal{W}$, and the set of all vectors in $\mathcal{V}$ that are orthogonal to $\mathcal{W}$ is called the *orthogonal complement of* $\mathcal{W}$ and is denoted by $\mathcal{W}^{\perp}$.

Definition A.3. Let $\mathcal{W}$ be a finite-dimensional subspace of an inner product space $\mathcal{V}$ and $\mathbf{u}$ be a vector in $\mathcal{V}$. A vector $\mathbf{w}_1$ in $\mathcal{V}$ is said to be *orthogonal projection on of* $\mathbf{u}$ on $\mathcal{W}$ if $\mathbf{w}_1$ is in $\mathcal{W}$ and $< (\mathbf{u} - \mathbf{w}_1), \mathbf{w_2} >= 0$ for every $\mathbf{w_2}$ in $\mathcal{W}$. Then, $\mathbf{w}_1$ is denoted by $proj_{\mathcal{W}}\mathbf{u}$.

Theorem A.4. *If $\mathcal{W}$ is a finite-dimensional subspace of an inner product space $\mathcal{V}$, then every vector $\mathbf{u}$ in $\mathcal{V}$ can be expressed in exactly one way as*

$$\mathbf{u} = proj_{\mathcal{W}}\mathbf{u} + \mathbf{w}_2 \tag{A.1}$$

where $proj_{\mathcal{W}}\mathbf{u}$ is in $\mathbf{W}$ and $\mathbf{w}_2$ is in $\mathcal{W}^{\perp}$.

A.3 Best Approximation-Least Squares

Theorem A.5. *If $\mathcal{W}$ is a finite-dimensional subspace of an inner product space $\mathcal{V}$, and if $\mathbf{u}$ is a vector in $\mathcal{V}$, then $proj_{\mathcal{W}}\mathbf{u}$ is the best approximation to $\mathbf{u}$ from $\mathcal{W}$ in the sense that*

$$\|\mathbf{u} - proj_{\mathcal{W}}\mathbf{u}\| \leq \|\mathbf{u} - \mathbf{w}\| \tag{A.2}$$

for every $\mathbf{w}$ in $\mathcal{W}$ that is different from $proj_{\mathcal{W}}\mathbf{u}$.

A.4 Symmetric Matrix

Symmetric matrices have a lot of interesting and aesthetically pleasing properties with respect to eigenvalue decomposition. A sample of the most im-

portant results that form the background for our analysis is given here.

Theorem A.6. *If* $\mathbf{A}$ *is an* $p \times p$ *matrix, then the following are equivalent.*

(1) $\mathbf{A}$ *is orthogonally diagonalizable.*

(2) $\mathbf{A}$ *has an orthonormal set of* n *eigenvectors.*

(3) $\mathbf{A}$ *is symmetric.*

Theorem A.7. *If* $\mathbf{A}$ *is a symmetric matrix, then:*

(1) The eigenvalues of $\mathbf{A}$ *are all real numbers.*

(2) Eigenvectors from different eigenvalues are orthogonal.

Definition A.8. Let $\mathbf{A}$ be an $p \times p$ matrix. $\mathbf{A}$ is said to be a *positive definite* if $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^p \setminus \{0\}$, is said to be a positive semidefinite if $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^p$.

Theorem A.9. $\mathbf{A}$ *is a symmetric matrix and positive semidefinite matrix, then all of the eigenvalues of* $\mathbf{A}$ *are nonnegative real numbers.*

A.5 Random Vectors and Matrices

We define random variable, random vector and random matrix that are used in the subsequences chapters. We start with definition of experiment. Experiment is defined as any process of observation or measurement. Then, the results of an experiment are called the *outcomes* of the experiment. A sample space is a set of all possible outcomes of an experiment and denoted by Ω. A *random variable* is defined as function from Ω to $\mathbb{R}$. Let $\mathcal{X}$ be a random variable with probability density function $f(x)$. The expected value

of X, denoted by $E(X)$, is defined as

$$\mathrm{E}(X) = \begin{cases} \sum_{x \in Rg(X)} x f(x) & \text{if } X \text{ is a discrete random variable,} \\ \int_{x \in \mathbb{R}} x f(x) dx & \text{if } X \text{ is a continuous random variable,} \end{cases}$$

where $Rg(X)$ is a range of X.

A *random vector* is a vector whose elements are random variables. Similarly, a *random matrix* is a matrix whose elements are random variables. The expected value of random vector (matrix) is the matrix consisting of the expected values of each of its elements.

B Theorems and Lemmas

Lemma B.1. *Let* $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ *be a set of vectors in* $\mathbb{R}^p$, $\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_N \end{pmatrix}^T$ *and* $\mathbf{C} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T$. *Suppose* $\hat{\lambda} \neq 0$ *and* $\hat{\mathbf{v}} \in \mathbb{R}^p \backslash \{\mathbf{0}\}$. *If* $\hat{\lambda}$ *and* $\hat{\mathbf{v}}$ *satisfy* $\lambda \mathbf{v} = \mathbf{C}\mathbf{v}$, *then* $\hat{\lambda}$ *and* $\hat{\mathbf{v}}$ *also satisfy* $\lambda \mathbf{x}_k^T \mathbf{v} = \mathbf{x}_k^T \mathbf{C}\mathbf{v}$, *for* $k = 1, \ldots, N$, *and* $\mathbf{v} \in$ *span* $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$.

Proof. 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) $\hat{\lambda}\mathbf{x}_k^T\hat{\mathbf{v}} = \mathbf{x}_k^T\mathbf{C}\hat{\mathbf{v}}$, for $k = 1, \ldots, N$.
(b) $\hat{\mathbf{v}} = \frac{1}{\hat{\lambda}}\mathbf{C}\hat{\mathbf{v}}$, since $\hat{\lambda} \neq 0$.
$\Rightarrow \hat{\mathbf{v}} = \frac{1}{\hat{\lambda}}\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^T\hat{\mathbf{v}} = \sum_{i=1}^{N} \frac{1}{N\hat{\lambda}}(\mathbf{x}_i^T\hat{\mathbf{v}})\mathbf{x}_i$.
Let $\alpha_i = \frac{1}{N\hat{\lambda}}\mathbf{x}_i^T\hat{\mathbf{v}}$,
$\Rightarrow \hat{\mathbf{v}} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i$.
$\Rightarrow \hat{\mathbf{v}} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$.
$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{v}}$ satisfy $\lambda\mathbf{x}_k^T\mathbf{v} = \mathbf{x}_k^T\mathbf{C}\mathbf{v}$, for $k = 1, \ldots, N$, and $\mathbf{v} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$.
□

B.1 Proof of Theorem 2.2.1

Proof. We prove (1) ⇒ (2), (2) ⇒ (3) and (3) ⇒ (1).
(1) ⇒ (2):

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

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\psi(\mathbf{x}_k)^T\mathbf{a} = \psi(\mathbf{x}_k)^T\tilde{\mathbf{C}}\mathbf{a}$, for $k = 1, \ldots, N$,
and $\mathbf{a} \in$ span $\{\psi(\mathbf{x}_1), \psi(\mathbf{x}_2), \ldots, \psi(\mathbf{x}_N)\}$ (By Lemma B.1).

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

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\psi(\mathbf{x}_k)^T\sum_{i=1}^N b_i\psi(\mathbf{x}_i) = \psi(\mathbf{x}_k)^T\tilde{\mathbf{C}}\sum_{i=1}^N b_i\psi(\mathbf{x}_i)$,
for $k = 1, \ldots, N$, and $\mathbf{a} = \sum_{i=1}^N b_i\psi(\mathbf{x}_i)$ for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\sum_{i=1}^N b_i\psi(\mathbf{x}_k)^T\psi(\mathbf{x}_i) = \sum_{i=1}^N b_i\psi(\mathbf{x}_k)^T\tilde{\mathbf{C}}\psi(\mathbf{x}_i)$,
for $k = 1, \ldots, N$, and $\mathbf{a} = \sum_{i=1}^N b_i\psi(\mathbf{x}_i)$ for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\sum_{i=1}^N b_i\psi(\mathbf{x}_k)^T\psi(\mathbf{x}_i) = \sum_{i=1}^N b_i\psi(\mathbf{x}_k)^T\sum_{j=1}^N \psi(\mathbf{x}_j)\psi(\mathbf{x}_j)^T\psi(\mathbf{x}_i)$,
for $k = 1, \ldots, N$, and $\mathbf{a} = \sum_{i=1}^N b_i\psi(\mathbf{x}_i)$ for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\sum_{i=1}^N b_i\psi(\mathbf{x}_k)^T\psi(\mathbf{x}_i) = (\mathbf{K}\boldsymbol{b})_k$ and
$\sum_{i=1}^N b_i\psi(\mathbf{x}_k)^T\sum_{j=1}^N \psi(\mathbf{x}_j)\psi(\mathbf{x}_j)^T\psi(\mathbf{x}_i) = (\mathbf{K}^2\boldsymbol{b})_k$ for $k = 1, \ldots, N$,
where $(\mathbf{K}\boldsymbol{b})_k$ is the kth element of $\mathbf{K}\boldsymbol{b}$ and $(\mathbf{K}^2\boldsymbol{b})_k$
is the kth element of $\mathbf{K}^2\boldsymbol{b}$.

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N(\mathbf{K}\boldsymbol{b})_k = (\mathbf{K}^2\boldsymbol{b})_k$, for $k = 1, \ldots, N$,
and $\mathbf{a} = \sum_{i=1}^N b_i\psi(\mathbf{x}_i)$ for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}$ 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 $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\mathbf{a} = \boldsymbol{\Psi}^T\mathbf{b}$ for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

(2) $\Rightarrow$ (3):

Suppose $\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\mathbf{a} = \boldsymbol{\Psi}^T\mathbf{b}$ for some
$\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \ldots & b_N \end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\hat{\mathbf{a}} = \boldsymbol{\Psi}^T\mathbf{b}$ for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \exists_{\mathbf{b}=(b_1 \quad b_2 \quad \ldots \quad b_N)^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}} \hat{\lambda}N\mathbf{K}\boldsymbol{b} = \mathbf{K}^2\boldsymbol{b}$ and $\hat{\mathbf{a}} = \boldsymbol{\Psi}^T\mathbf{b}$.

Since $\mathbf{K}$ is symmetric,

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

Let λ_i be the eigenvalue of $\mathbf{K}$ belonging to $\mathbf{p}_i$, for $i = 1, \ldots, N$.

$\Rightarrow \lambda_i\mathbf{p}_i = \mathbf{C}\mathbf{p}_i$, for $i = 1, \ldots, N$.

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

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

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

$\Rightarrow \hat{\lambda} N \mathbf{K} \sum_{i=1}^{N} \alpha_i \mathbf{p}_i = \mathbf{K}^2 \sum_{i=1}^{N} \alpha_i \mathbf{p}_i$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^{N} \alpha_i \mathbf{K} \mathbf{p}_i = \sum_{i=1}^{N} \alpha_i \mathbf{K}^2 \mathbf{p}_i$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^{N} \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^{N} \alpha_i \lambda_i^2 \mathbf{p}_i$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \sum_{i=1}^{N} (\hat{\lambda} N \alpha_i \lambda_i - \alpha_i \lambda_i^2) \mathbf{p}_i = \mathbf{0}$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$ for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ is linearly independent,

$\Rightarrow (\hat{\lambda} N \alpha_i \lambda_i - \alpha_i \lambda_i^2) = 0$, $(i = 1, \ldots, N)$, and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \lambda_i (\hat{\lambda} N \alpha_i - \alpha_i \lambda_i) = 0$, $(i = 1, \ldots, N)$, and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\lambda_i > 0$ for $i = 1, \ldots, N$,

$\Rightarrow (\hat{\lambda} N \alpha_i - \alpha_i \lambda_i) = 0$, $(i = 1, \ldots, N)$, and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \alpha_i = \alpha_i \lambda_i$, $(i = 1, \ldots, N)$, and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \alpha_i \mathbf{p}_i = \alpha_i \lambda_i \mathbf{p}_i$, $(i = 1, \ldots, N)$, and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N \sum_{i=1}^{N} \alpha_i \mathbf{p}_i = \sum_{i=1}^{N} \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^{N} \alpha_i \mathbf{K} \mathbf{p}_i$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\mathbf{b} = \sum_{i=1}^{N} \alpha_i \mathbf{p}_i$,

$\Rightarrow \hat{\lambda} N \mathbf{b} = \mathbf{K} \mathbf{b}$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$ for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N \tilde{\boldsymbol{b}} = \mathbf{K} \tilde{\boldsymbol{b}}$ and $\mathbf{a} = \mathbf{\Psi}^T \tilde{\mathbf{b}}$, for some $\tilde{\mathbf{b}} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

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

$\Rightarrow \hat{\lambda} N \mathbf{K} \sum_{i=1}^{N} \alpha_i \mathbf{p}_i = \mathbf{K}^2 \sum_{i=1}^{N} \alpha_i \mathbf{p}_i$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{b}$
for some $\mathbf{b} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

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

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

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

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 $k = 1, 2, \dots, N$.

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

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

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

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

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

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

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

$\Rightarrow \hat{\lambda}N\mathbf{K}(\mathbf{v}_1 + \mathbf{v}_2) = \mathbf{K}^2(\mathbf{v}_1 + \mathbf{v}_2)$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T\mathbf{b} = \sum_{i=1}^{N}(v_{1i} + v_{2i})\psi(\mathbf{x}_i)$ 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 by the fact $\sum_{i=1}^{N} v_{2i}\psi(\mathbf{x}_i) = \mathbf{0}$,

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

$\Rightarrow \hat{\lambda}N\mathbf{K} \sum_{i=1}^{r} \alpha_i \mathbf{p}_i = \mathbf{K}^2 \sum_{i=1}^{r} \alpha_i \mathbf{p}_i$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}N \sum_{i=1}^{r} \alpha_i \mathbf{K}\mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \mathbf{K}^2\mathbf{p}_i$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda}N \sum_{i=1}^{r} \alpha_i \lambda_i \mathbf{p}_i = \sum_{i=1}^{r} \alpha_i \lambda_i^2 \mathbf{p}_i$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \sum_{i=1}^{r}(\hat{\lambda}N\alpha_i\lambda_i - \alpha_i\lambda_i^2)\mathbf{p}_i = \mathbf{0}$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

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

$\Rightarrow (\hat{\lambda}N\alpha_i\lambda_i - \alpha_i\lambda_i^2) = 0$, $(i = 1, \dots, r)$, and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \lambda_i(\hat{\lambda}N\alpha_i - \alpha_i\lambda_i) = 0$, $(i = 1, \dots, r)$, and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

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

$\Rightarrow (\hat{\lambda}N\alpha_i - \alpha_i\lambda_i) = 0$, $(i = 1, \dots, r)$, and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$ for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

$\Rightarrow \hat{\lambda} N\alpha_i = \alpha_i\lambda_i$, $(i = 1, \ldots, r)$, and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$
for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow \hat{\lambda} N\alpha_i\mathbf{p}_i = \alpha_i\lambda_i\mathbf{p}_i$, $(i = 1, \ldots, r)$, and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$
for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow \hat{\lambda} N \sum_{i=1}^{r} \alpha_i\mathbf{p}_i = \sum_{i=1}^{r} \alpha_i\lambda_i\mathbf{p}_i = \sum_{i=1}^{r} \alpha_i\mathbf{K}\mathbf{p}_i$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$
for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
Since $\mathbf{v}_1 = \sum_{i=1}^{r} \alpha_i\mathbf{p}_i$,
$\Rightarrow \hat{\lambda} N\mathbf{v}_1 = \mathbf{K}\mathbf{v}_1$ and $\hat{\mathbf{a}} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$
for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{v}_1 = \mathbf{K}\mathbf{v}_1$ and $\mathbf{a} = \sum_{i=1}^{N} v_{1i}\psi(\mathbf{x}_i)$
for some $\mathbf{v}_1 \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\tilde{\mathbf{b}} = \mathbf{K}\tilde{\mathbf{b}}$ and $\mathbf{a} = \sum_{i=1}^{N} \tilde{b}_i\psi(\mathbf{x}_i)$
for some $\tilde{\mathbf{b}} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.
$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\tilde{\mathbf{b}} = \mathbf{K}\tilde{\mathbf{b}}$ and $\mathbf{a} = \mathbf{\Psi}^T\tilde{\mathbf{b}}$ for some $\tilde{\mathbf{b}} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Case 3: $\lambda_1 = \lambda_2 = \ldots = \lambda_r = \lambda_{r+1} = \ldots = \lambda_N = 0$.
$\Rightarrow \mathbf{K}\mathbf{b} = \sum_{i=1}^{N} \alpha_i\mathbf{K}\mathbf{p}_i = \mathbf{0}$.
$\Rightarrow \sum_{i=1}^{N} b_i(\mathbf{K})_{ki} = 0$ for $k = 1, 2, \ldots, N$.
$\Rightarrow \sum_{i=1}^{N} b_i\psi(\mathbf{x}_k)^T\psi(\mathbf{x}_i) = 0$ for $k = 1, 2, \ldots, N$.
$\Rightarrow \psi(\mathbf{x}_k)^T \sum_{i=1}^{N} b_i\psi(\mathbf{x}_i) = 0$ for $k = 1, 2, \ldots, N$.
$\Rightarrow \hat{\mathbf{a}} = \sum_{i=1}^{N} b_i\psi(\mathbf{x}_i) = \mathbf{0}$ (Why?).
Suppose $\sum_{i=1}^{N} b_i\psi(\mathbf{x}_i) \neq \mathbf{0}$.
$\Rightarrow (\sum_{i=1}^{N} b_i\psi(\mathbf{x}_i))^T(\sum_{j=1}^{N} b_j\psi(\mathbf{x}_j)) \neq 0$.
$\Rightarrow b_1\psi^T(\mathbf{x}_1) \sum_{j=1}^{N} b_j\psi(\mathbf{x}_j) + b_2\psi^T(\mathbf{x}_2) \sum_{j=1}^{N} b_j\psi(\mathbf{x}_j) + \ldots$
$b_N\psi^T(\mathbf{x}_N) \sum_{j=1}^{N} b_j\psi(\mathbf{x}_j) \neq 0$
$\Rightarrow 0 \neq 0$.
(Contradiction)

(3) $\Rightarrow$ (1):
$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\tilde{\boldsymbol{b}} = \mathbf{K}\tilde{\boldsymbol{b}}$ and $\mathbf{a} = \mathbf{\Psi}^T\tilde{\mathbf{b}}$, for some $\tilde{\mathbf{b}} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$,
$\Rightarrow \hat{\lambda} N\tilde{\boldsymbol{b}} = \mathbf{K}\tilde{\boldsymbol{b}}$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T\tilde{\mathbf{b}}$, for some $\tilde{\mathbf{b}} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$,
$\Rightarrow \hat{\lambda} N\mathbf{\Psi}^T\tilde{\boldsymbol{b}} = \mathbf{\Psi}^T\mathbf{K}\tilde{\boldsymbol{b}}$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T\tilde{\mathbf{b}}$, for some $\tilde{\mathbf{b}} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$,
$\Rightarrow \hat{\lambda} N\mathbf{\Psi}^T\tilde{\boldsymbol{b}} = \mathbf{\Psi}^T\mathbf{\Psi}\mathbf{\Psi}^T\tilde{\boldsymbol{b}}$ and $\hat{\mathbf{a}} = \mathbf{\Psi}^T\tilde{\mathbf{b}}$, for some $\tilde{\mathbf{b}} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$,
$\Rightarrow \hat{\lambda} N\hat{\mathbf{a}} = \mathbf{\Psi}^T\mathbf{\Psi}\hat{\mathbf{a}}$,

$\Rightarrow \hat{\lambda}\hat{\mathbf{a}} = \tilde{\mathbf{C}}\hat{\mathbf{a}}$,
$\Rightarrow \hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda\mathbf{a} = \tilde{\mathbf{C}}\mathbf{a}$. □

B.2 Proof of Lemma 3.1.1

Proof. In 3.1, 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 $\mathbf{y}_o = \mathbf{B}\mathbf{y}$. This implies

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

Since

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

we obtain $\mathbf{U}_{(\hat{p}-r)} = \mathbf{Z}\mathbf{A}_{(\hat{p}-r)}$. This implies,

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

□

B.3 Proof of Lemma 4.1.1

Proof. As mentioned in 3.2, we assume that $\sum_{i=1}^{N} \psi(\mathbf{x}_i) = \mathbf{\Psi}^T \mathbf{1}_N = \mathbf{0}$. This implies $\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{\Psi} = \mathbf{O}$. Hence, 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

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

Since

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

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

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

□

B.4 Proof of Lemma 5.1.1

Proof. Note that if $\mathbf{E}$ is invertible matrix, we have the following statements.

- If $\mathbf{EA} = \mathbf{EB}$, then $\mathbf{A} = \mathbf{B}$.
- If $\mathbf{AE} = \mathbf{BE}$, then $\mathbf{A} = \mathbf{B}$.

It is evident that $(\mathbf{\Psi}^T\mathbf{\Psi} + \tilde{c}\mathbf{I}_{p_F})$ invertible matrix. Then,

$$(\mathbf{\Psi}^T\mathbf{\Psi} + \tilde{c}\mathbf{I}_{p_F})(\mathbf{\Psi}^T\mathbf{\Psi} + \tilde{c}\mathbf{I}_{p_F})^{-1}\mathbf{\Psi}^T\mathbf{y} = \mathbf{\Psi}^T\mathbf{y} \tag{B.1}$$

$$\begin{aligned}(\mathbf{\Psi}^T\mathbf{\Psi} + \tilde{c}\mathbf{I}_{p_F})\mathbf{\Psi}^T(\mathbf{\Psi}\mathbf{\Psi}^T + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y} &= (\mathbf{\Psi}^T\mathbf{\Psi}\mathbf{\Psi}^T + \tilde{c}\mathbf{I}_{p_F}\mathbf{\Psi}^T)(\mathbf{\Psi}\mathbf{\Psi}^T + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y} \\ &= (\mathbf{\Psi}^T\mathbf{\Psi}\mathbf{\Psi}^T + \tilde{c}\mathbf{\Psi}^T)(\mathbf{\Psi}\mathbf{\Psi}^T + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y} \\ &= \mathbf{\Psi}^T(\mathbf{\Psi}\mathbf{\Psi}^T + \tilde{c}\mathbf{I}_N)(\mathbf{\Psi}\mathbf{\Psi}^T + \tilde{c}\mathbf{I}_N)^{-1}\mathbf{y} \\ &= \mathbf{\Psi}^T\mathbf{y}. \end{aligned} \tag{B.2}$$

By letting $\mathbf{E} = (\mathbf{\Psi}^T\mathbf{\Psi}+\tilde{c}\mathbf{I}_{p_F})$, $\mathbf{A} = (\mathbf{\Psi}^T\mathbf{\Psi}+\tilde{c}\mathbf{I}_{p_F})^{-1}\mathbf{\Psi}^T\mathbf{y}$ and $\mathbf{B} = \mathbf{\Psi}^T(\mathbf{\Psi}\mathbf{\Psi}^T+\tilde{c}\mathbf{I}_N)^{-1}\mathbf{y}$, we have proven Lemma 5.1.1. □

C AIC for KPCR

Let consider the KPCR model (4.1.14) again

$$\mathbf{Y}_o = \tilde{\mathbf{U}}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})} + \tilde{\tilde{\boldsymbol{\epsilon}}}, \tag{C.1}$$

where $\tilde{\tilde{\boldsymbol{\epsilon}}}$ is normally and independently distributed with mean $\mathbf{0}$ and constant variance σ^2, or $\tilde{\tilde{\boldsymbol{\epsilon}}}$ is distributed as $N(\mathbf{0}, \sigma^2\mathbf{I}_N)$. The normal density for the errors is

$$s(\tilde{\tilde{\epsilon}}_i) = \frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{1}{2\sigma^2}\tilde{\tilde{\epsilon}}_i) \qquad \text{for i=1,2,..., N} \tag{C.2}$$

The likelihood function of $\tilde{\tilde{\epsilon}}_1, \tilde{\tilde{\epsilon}}_2, \ldots, \tilde{\tilde{\epsilon}}_N$ is

$$L(\tilde{\tilde{\boldsymbol{\epsilon}}}, \boldsymbol{\vartheta}_{(\tilde{r})}, \sigma^2) = \Pi_{i=1}^N s(\tilde{\tilde{\epsilon}}_i) = \frac{1}{(2\pi)^{N/2}\sigma^N}\exp(-\frac{1}{2\sigma^2}\tilde{\tilde{\boldsymbol{\epsilon}}}^T\tilde{\tilde{\boldsymbol{\epsilon}}}). \tag{C.3}$$

Since $\tilde{\tilde{\boldsymbol{\epsilon}}} = \mathbf{Y}_o - \tilde{\mathbf{U}}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})}$, the likelihood can be written as

$$L(\mathbf{Y}_o, \tilde{\mathbf{U}}_{(\tilde{r})}, \boldsymbol{\vartheta}_{(\tilde{r})}, \sigma^2) = \frac{1}{(2\pi)^{N/2}\sigma^N} \exp(-\frac{1}{2\sigma^2}(\mathbf{Y}_o - \tilde{\mathbf{U}}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})})^T(\mathbf{Y}_o - \tilde{\mathbf{U}}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})})).$$

The log of the likelihood function is

$$\ln L(\mathbf{Y}_o, \tilde{\mathbf{U}}_{(\tilde{r})}, \boldsymbol{\vartheta}_{(\tilde{r})}, \sigma^2) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{Y}_o - \tilde{\mathbf{U}}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})})^T(\mathbf{Y}_o - \tilde{\mathbf{U}}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})}).$$

It is evident that for a fixed value of σ the log-likehood is maximized when the term

$$(\mathbf{Y}_o - \tilde{\mathbf{U}}_{(\hat{p}_F)}\boldsymbol{\vartheta}_{(\tilde{r})})^T(\mathbf{Y}_o - \tilde{\mathbf{U}}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})})$$

is minimized. Therefore, the maximum-likelihood estimator of $\boldsymbol{\vartheta}_{(\tilde{r})}$ is

$$\hat{\boldsymbol{\vartheta}}_{(\tilde{r})} = (\tilde{\mathbf{U}}_{(\tilde{r})}^T\tilde{\mathbf{U}}_{(\tilde{r})})^{-1}\tilde{\mathbf{U}}_{(\tilde{r})}^T\mathbf{Y}. \tag{C.4}$$

and the maximum-likelihood estimator of σ^2 is

$$\hat{\sigma}^2 = \frac{(\mathbf{Y}_o - \tilde{\mathbf{U}}_{(\tilde{r})}\hat{\boldsymbol{\vartheta}}_{(\tilde{r})})^T(\mathbf{Y}_o - \tilde{\mathbf{U}}_{(\tilde{r})}\hat{\boldsymbol{\vartheta}}_{(\tilde{r})})}{N}. \tag{C.5}$$

Furthermore, the Akaike Information Criteria (AIC) is defined by

$$AIC = -2\ln(L_{max}) + 2\tilde{t} \tag{C.6}$$

where $\ln(L_{max})$ is the maximized value of the corresponding likelihood function and $\tilde{t}$ is the number of parameters in the statistical model. Hence, the AIC for KPCR is

$$AIC_{kpcr} = N\ln(2\pi\hat{\sigma}^2) + N + 2\tilde{r}. \tag{C.7}$$