Department of Social Systems and Management

Discussion Paper Series

No. 1218

Robust Prediction in Kernel Principal Component Regression Based on M-Estimation

by

Antoni Wibowo

October 2008

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

Robust Prediction in Kernel Principal Component Regression Based on M-Estimation

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

October 14, 2008

Abstract

Robust regression is an important tool for analyzing data that are contaminated with outliers. Fomengko *et al.* [5] proposed a nonlinear robust prediction based on the M-estimation; their method, however, needs a specific nonlinear regression model in advance. In this paper, we propose a method to obtain a nonlinear robust prediction without specifying a nonlinear model in advance. We combine M-estimation and kernel principal component regression to obtain the nonlinear prediction. Then, we compare the proposed method with some other methods.

Keywords: Robust, nonlinear robust regression, kernel principal component analysis, kernel principal component regression, robust kernel principal component regression.

1 Introduction

Regression analysis is one of the most widely used techniques for analyzing data. The *ordinary multiple linear regression model* is given by

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.1}$$

where $\mathbf{Y} = (Y_1 \quad Y_2 \quad \dots \quad Y_N)^T$, $\tilde{\mathbf{X}} = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_N)^T$, $\mathbf{X} = \begin{pmatrix}\mathbf{1}_N & \tilde{\mathbf{X}}\end{pmatrix}$, $\mathbf{x}_i = \begin{pmatrix}x_{i1} & x_{i2} & \dots & x_{ip}\end{pmatrix}^T \in \mathbb{R}^p$, $\boldsymbol{\beta} = (\beta_0 \quad \beta_1 \quad \dots \quad \beta_p)^T$ is a vector of regression coefficients, $\boldsymbol{\epsilon} = (\epsilon_1 \quad \epsilon_2 \quad \dots \quad \epsilon_N)^T$ is a vector of random errors, I_N denotes the $N \times N$ identity matrix, $x_{ij} \in \mathbb{R}$ for $i = 1, 2, \dots, N; j = 1, 2, \dots, p$; and $\sigma^2 \in \mathbb{R}$ where $\mathbb{R}$ is the set of real numbers. The 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 $\mathbf{1}_N = (1 \quad 1 \quad \dots \quad 1)^T_{N \times 1}$. The vector $\mathbf{x}_i^T$ denotes the transpose of the vector $\mathbf{x}_i$. Matrix $\mathbf{X}$ is called the *regression matrix*.

Let $\mathbf{y} = (y_1 \quad y_2 \quad \dots \quad y_N)^T \in \mathbb{R}^N$ be the observed data corresponding to $\mathbf{Y}$. Hence, we have

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{e}, \tag{1.2}$$

*Email address: wibowo@sk.tsukuba.ac.jp.

where $\boldsymbol{e} \in \mathbb{R}^N$ is a vector of residuals. The aim of regression analysis is to find the estimator of $\boldsymbol{\beta}$, say $\hat{\boldsymbol{\beta}} = \begin{pmatrix} \hat{\beta}_0 & \hat{\beta}_1 & \dots & \hat{\beta}_p \end{pmatrix}^T \in \mathbb{R}^{p+1}$, such that the *least-squares function*,

$$\begin{aligned} \mathcal{S}(\boldsymbol{\beta}) &= \boldsymbol{e}^T\boldsymbol{e} \\ &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \end{aligned} \tag{1.3}$$

is minimized. The solution can be found by solving the following linear equation

$$\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{y}. \tag{1.4}$$

Eq. (1.4) is called the *least squares normal equations* and $\hat{\boldsymbol{\beta}}$ is called the *ordinary least-squares (OLS) estimator* of $\boldsymbol{\beta}$. The procedure to obtain $\hat{\boldsymbol{\beta}}$ by solving Eq. (1.3) is called the *OLS method*.

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

$$\hat{\mathbf{y}} = \begin{pmatrix} \hat{y}_1 & \hat{y}_2 & \dots & \hat{y}_N \end{pmatrix}^T := \mathbf{X}\hat{\boldsymbol{\beta}}, \tag{1.5}$$

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

$$\hat{\mathbf{e}} = \begin{pmatrix} \hat{e}_1 & \hat{e}_2 & \dots & \hat{e}_N \end{pmatrix}^T := \mathbf{y} - \hat{\mathbf{y}}. \tag{1.6}$$

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

$$RMSE_{ols} := \sqrt{\frac{\hat{\mathbf{e}}^T\hat{\mathbf{e}}}{N}}, \tag{1.7}$$

and the *prediction by ordinary linear regression* is given by

$$f(\boldsymbol{x}) := \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j, \tag{1.8}$$

where f is a function from $\mathbb{R}^p$ to $\mathbb{R}$ and $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_p \end{pmatrix}^T \in \mathbb{R}^p$.

The main disadvantage of the OLS method is its sensitivity to outliers, i.e., residuals of the observed data are large numbers. Outlier have a large influence the prediction value because squaring residuals magnifies the effect of the outliers. If the outliers are contained in the observed data, the predictions of ordinary linear regression and *kernel principal component regression* (*KPCR*) become inappropriate to be used. Since those methods were constructed based on OLS method. The KPCR was studied by Rosipal *et al.* [14, 15, 16], Hoegaerts *et al.* [7], Jade *et al.* [9], Wibowo *et al.* [21] and Wibowo [20].

Andrews, Carol and Ruppert; Hogg, Hubber, Krasker and Welsch; Rousseeuw, and Rousseeuw and Leroy proposed *robust regression* methods to eliminate the influence of the outliers [12]. The *M-estimation*, which was first introduced by Huber in 1964, is one of the most widely used methods for the robust regression in which the method usually yields a linear prediction. We notice that Fomengko [5] proposed a nonlinear robust prediction based on M-estimation with specifying a nonlinear regression model in advance. In many situations, however, an appropriate nonlinear regression model for a set of data is unknown in advance. Hence, the proposed method has limitations in applications.

In this paper, we propose a technique to obtain a nonlinear robust prediction without specifying a nonlinear model in advance. In our proposed technique, we combine M-estimation and KPCR to obtain the nonlinear robust prediction. The procedure to derive the nonlinear prediction of the proposed method is straightforward as the procedure of the M-estimation in linear regression, except that some mathematical techniques are done to obtain the nonlinear robust prediction. We refer the proposed technique as the *Robust KPCR (R-KPCR)*.

This manuscript is organized as follows: Section 2, we review the robust linear regression based on M-estimation. In Section 3, the R-KPCR and its algorithm will be discussed. In Section 4, we compare the capabilities of R-KPCR with other methods. Finally, conclusions are given in Section 5.

2 Linear Robust Regression based on M-Estimation

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. (1.3) can be written as

$$\sum_{i=1}^{N} e_i^2, \tag{2.1}$$

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

$$\sum_{i=1}^{N} \rho(e_i) = \sum_{i=1}^{N} \rho(y_i - \acute{\mathbf{x}}_i^T \boldsymbol{\beta}), \tag{2.2}$$

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 ($\rho(|e(i_1)|) > \rho(|e(i_2)|)$ if $|e(i_1)| > |e(i_2)|$), and a convex function on $\mathbb{R}$. For example, for the OLS estimation, $\rho(z) = z^2$. The most commonly used choice of ρ is the Huber function [2]

$$\rho(z) = \begin{cases} 1/2z^2 & |z| \leq k, \\ k|z| - 1/2k^2 & |z| > k, \end{cases} \tag{2.3}$$

where $k \in \mathbb{R}$. Fomenko et al. [5] used the Tukey biweighted function

$$\rho(z) = \begin{cases} 1/6[(1-(1-z^2)^3] & |z| \leq 1, \\ 1/6 & |z| > 1. \end{cases} \tag{2.4}$$

To minimize Eq. (2.2), 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

$$\sum_{i=1}^{N} \rho'(e_i) \acute{\mathbf{x}}_i^T = \mathbf{0}^T, \tag{2.5}$$

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

$$w(z) = \begin{cases} \rho'(z)/z & \text{if } z \neq 0, \\ 1 & \text{if } z = 0, \end{cases} \tag{2.6}$$

Then, Eq. (2.5) can be written as

$$\sum_{i=1}^{N} (\frac{\rho'(e_i)}{e_i}) e_i \acute{\mathbf{x}}_i^T = \sum_{i=1}^{N} w_i e_i \acute{\mathbf{x}}_i^T = \mathbf{0}^T, \tag{2.7}$$

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

$$\sum_{i=1}^{N} w_i y_i \acute{\mathbf{x}}_i^T = \sum_{i=1}^{N} w_i \acute{\mathbf{x}}_i^T \boldsymbol{\beta} \acute{\mathbf{x}}_i^T. \tag{2.8}$$

In matrix form, Eq. (2.8) becomes

$$\mathbf{X}^T \mathbf{W} \mathbf{X} \boldsymbol{\beta} = \mathbf{X}^T \mathbf{W} \mathbf{y}, \tag{2.9}$$

where $\mathbf{W} = \text{diag}(w_1, w_2, \ldots, w_N)$ and called the *weigthed least squares equations*. Let $\hat{\boldsymbol{\beta}}^*$ be the solution of Eq. (2.9). Hence, we have

$$\mathbf{X}^T \mathbf{W} \mathbf{X} \hat{\boldsymbol{\beta}}^* = \mathbf{X}^T \mathbf{W} \mathbf{y}. \tag{2.10}$$

The estimator $\hat{\boldsymbol{\beta}}^*$ 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's algorithm is given in the following steps:

1. Select the initial estimator of $\boldsymbol{\beta}$, say $\hat{\boldsymbol{\beta}}^{*(\mathbf{0})}$, by OLS.
2. At each iteration t, calculate residual $e_i^{(t-1)} = y_i - \acute{\mathbf{x}}_i^T \boldsymbol{\beta}^{*(t-1)}$, $w_i^{(t-1)} = w(e_i^{(t-1)})$ 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

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

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

The convergence proof of IRLS's algorithm is not possible without unsupported distribution assumption. The convergence proof of IRLS can be found in [22]. However, IRLS works well in practice [3, 8] and is frequently used in the computational statistic community [4, 13].

3 Robust Kernel Principal Component Regression

3.1 Regression in Feature Space

Assume we have a function $\psi : \mathbb{R}^p \to \mathcal{F}$, where $\mathcal{F}$ is the feature space which is an Euclidean space with dimension p_F ($p_F \geq p$). Then, we define $\boldsymbol{\Psi} = \begin{pmatrix}\psi(\mathbf{x}_1) & \dots & \psi(\mathbf{x}_N)\end{pmatrix}^T$, $\mathbf{C} := \frac{1}{N}\boldsymbol{\Psi}^T\boldsymbol{\Psi} = \frac{1}{N}\sum_{i=1}^N \psi(\mathbf{x}_i)\psi(\mathbf{x}_i)^T$ and $\mathbf{K} = \boldsymbol{\Psi}\boldsymbol{\Psi}^T$ where sizes of $\boldsymbol{\Psi}$, $\mathbf{C}$ 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 $\mathbf{C}$ [18]. The eigenvalues and eigenvectors of the matrices $\mathbf{C}$ and $\mathbf{K}$ are related by the following theorem.

Theorem 3.1. *[21] 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 $\lambda\mathbf{a} = \mathbf{C}\mathbf{a}$.*
2. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\mathbf{K}\hat{\boldsymbol{b}} = \mathbf{K}^2\hat{\boldsymbol{b}}$ and $\mathbf{a} = \sum_{i=1}^N \hat{b}_i\psi(\mathbf{x}_i)$, for some $\hat{\mathbf{b}} = \begin{pmatrix}\hat{b}_1 & \hat{b}_2 & \dots & \hat{b}_N\end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.*
3. *$\hat{\lambda}$ and $\hat{\mathbf{a}}$ satisfy $\lambda N\tilde{\boldsymbol{b}} = \mathbf{K}\tilde{\boldsymbol{b}}$ and $\mathbf{a} = \sum_{i=1}^N \tilde{b}_i\psi(\mathbf{x}_i)$, for some $\tilde{\mathbf{b}} = \begin{pmatrix}\tilde{b}_1 & \tilde{b}_2 & \dots & \tilde{b}_N\end{pmatrix}^T \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.*

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

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

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 $\mathbf{Y}_o = (I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\boldsymbol{Y}$. Let $\mathbf{y}_o = (y_{o1} \quad y_{o2} \quad \dots \quad y_{oN})^T \in \mathbb{R}^N$ be the observed data corresponding to $\mathbf{Y}_o$. Hence, we have

$$\mathbf{y}_o = \boldsymbol{\Psi}\boldsymbol{\gamma} + \tilde{\boldsymbol{e}}, \tag{3.2}$$

where $\tilde{\boldsymbol{e}} \in \mathbb{R}^N$ is a vector of residuals. Here, we cannot use the generalized inverse matrix to obtain the estimator of $\boldsymbol{\gamma}$ since $\boldsymbol{\Psi}$ is not known explicitly.

Let $\hat{p}_F$ be the rank of $\boldsymbol{\Psi}$ where $\hat{p}_F \leq \min(N, p_F)$. Since the rank($\boldsymbol{\Psi}$) is equal to rank($\mathbf{K}$) and rank ($\boldsymbol{\Psi}^T\boldsymbol{\Psi}$), then rank($\mathbf{K}$) and rank($\boldsymbol{\Psi}^T\boldsymbol{\Psi}$) are equal to $\hat{p}_F$. Since the matrix $\mathbf{K}$ is symmetric and positive semidefinite, the eigenvalues of $\mathbf{K}$ are nonnegative real numbers [1]. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{\tilde{r}} \geq \lambda_{\tilde{r}+1} \geq \ldots \geq \lambda_{\hat{p}_F} > \lambda_{\hat{p}_F+1} = \ldots = \lambda_N = 0$ be the eigenvalues of $\mathbf{K}$ and $\mathbf{B} = (\mathbf{b}_1 \quad \mathbf{b}_2 \quad \dots \quad \mathbf{b}_N)$ be the matrix of the corresponding normalized eigenvectors $\mathbf{b}_l$ of $\mathbf{K}$. Then, let $\boldsymbol{\alpha}_l = \frac{\mathbf{b}_l}{\sqrt{\lambda_l}}$ and $\mathbf{a}_l = \boldsymbol{\Psi}^T\boldsymbol{\alpha}_l$ for $l = 1, 2, \ldots, \hat{p}_F$. By Theorem 3.1 we obtain

$$\frac{\lambda_l}{N}\mathbf{a}_l = \mathbf{C}\mathbf{a}_l \qquad \text{for } l = 1, 2, \ldots, \hat{p}_F$$
$$\mathbf{a}_i^T\mathbf{a}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for } i, j = 1, 2, \ldots, \hat{p}_F,$$

or equivalent to

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

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 λ_k, $(k = \hat{p}_F + 1, \hat{p}_F + 2, \ldots, p_F)$, be the zero eigenvalues of $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ and $\mathbf{a}_k$ be the normalized eigenvectors of $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ corresponding to λ_k. Hence, we have

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

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

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

where

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{(\hat{p}_F)} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{pmatrix},$$
$$\mathbf{D}_{(\hat{p_F})} = \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \lambda_{\hat{p}_F} \end{pmatrix}.$$

By using $\mathbf{A}\mathbf{A}^T = \mathbf{I}_{p_F}$, we can rewrite the model (3.1) as

$$\mathbf{y}_o = \mathbf{U}\boldsymbol{\vartheta} + \tilde{\boldsymbol{e}}, \tag{3.3}$$

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

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

where sizes of $\mathbf{U}_{(\hat{p}_F)}$, $\mathbf{U}_{(p_F - \hat{p}_F)}$, $\boldsymbol{\vartheta}_{(\hat{p}_F)}$, 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 (3.3) can be written as

$$\mathbf{y}_o = \mathbf{U}_{(\hat{p}_F)}\boldsymbol{\vartheta}_{(\hat{p}_F)} + \mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)} + \tilde{\boldsymbol{e}}. \tag{3.4}$$

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

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

and

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

Since $(\mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)})^T\mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)} = 0$, we see that $\mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)}$ is equal to $\mathbf{0}$. Consequently, the model (3.4) reduces to

$$\mathbf{y}_o = \mathbf{U}_{(\hat{p}_F)}\boldsymbol{\vartheta}_{(\hat{p}_F)} + \tilde{\boldsymbol{e}}. \tag{3.5}$$

Since

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

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

$$\mathbf{U}_{(\hat{p}_F)} = \mathbf{\Psi}\mathbf{\Psi}^T\mathbf{\Gamma}_{(\hat{p}_F)} = \mathbf{K}\mathbf{\Gamma}_{(\hat{p}_F)}, \tag{3.6}$$

where $\mathbf{\Gamma}_{(\hat{p}_F)} = \begin{pmatrix}\boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \dots & \boldsymbol{\alpha}_{\hat{p}_F}\end{pmatrix}$. However, we do not know $\mathbf{U}_{(\hat{p}_F)}$ explicitly yet. Let us consider the following theorem:

Theorem 3.2. *(Mercer [11, 17]) For any symmetric, continuous and positive semi-definite 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*

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

By using Theorem 3.2, 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. This implies that $\mathbf{U}_{(\hat{p}_F)}$ is also explicitly known and model (3.5) is well defined now.

Let

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

Hence model (3.5) can be written as

$$\mathbf{y}_o = \mathbf{U}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})} + \mathbf{U}_{(\hat{p}_F-\tilde{r})}\boldsymbol{\vartheta}_{(\hat{p}_F-\tilde{r})} + \tilde{\boldsymbol{e}}. \tag{3.7}$$

Note that, $\mathbf{U}_{(\hat{p}_F)} = \mathbf{K}\mathbf{\Gamma}_{(\hat{p}_F)}$ and $\mathbf{\Gamma}_{(\hat{p}_F)} = \begin{pmatrix}\boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \dots & \boldsymbol{\alpha}_{\hat{p}_F}\end{pmatrix}$. If we only use the first $\tilde{r}$ vectors of $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_{\hat{p}_F}$, model (3.7) becomes

$$\mathbf{y}_o = \mathbf{U}_{(\tilde{r})}\boldsymbol{\vartheta}_{(\tilde{r})} + \tilde{\mathbf{e}}_{\mathbf{1}}, \tag{3.8}$$

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 $\mathbf{U}_{(\hat{p}_F-\tilde{r})}\boldsymbol{\vartheta}_{(\hat{p}_F-\tilde{r})}$ in model (3.7). Let $\mathbf{U}_{(\tilde{r})} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_N)^T$. Now, we apply M-estimation method for model (3.8) which minimize

$$\sum_{i=1}^{N}\rho(\tilde{e}_{\mathbf{1}i}) = \sum_{i=1}^{N}\rho(y_{oi} - \mathbf{u}_i^T\boldsymbol{\vartheta}_{(\tilde{r})}), \tag{3.9}$$

with respect to $\boldsymbol{\vartheta}_{(\tilde{r})}$. To minimize Eq. (3.9), equate the first partial derivatives of ρ with respect to β_j $(j = 0, 1, \dots, p)$ to zero. This gives the system of $\hat{p}_F$ equations

$$\sum_{i=1}^{N}\rho'(\tilde{e}_{\mathbf{1}i})\mathbf{u}_i^T = \mathbf{0}^T, \tag{3.10}$$

Then, we define the weight function

$$\tilde{w}(z) = \begin{cases} \rho'(z)/z & \text{if } z \neq 0, \\ 1 & \text{if } z = 0. \end{cases} \tag{3.11}$$

Then, Eq. (3.19) can be written as

$$\sum_{i=1}^{N} (\frac{\rho'(\tilde{e}_{\mathbf{1}i})}{\tilde{e}_{\mathbf{1}i}}) \tilde{e}_{\mathbf{1}i} \mathbf{u}_i^T = \sum_{i=1}^{N} \tilde{w}_i \tilde{e}_{\mathbf{1}i} \mathbf{u}_i^T = \mathbf{0}^T, \tag{3.12}$$

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

$$\sum_{i=1}^{N} \tilde{w}_i y_{oi} \mathbf{u}_i^T = \sum_{i=1}^{N} \tilde{w}_i \mathbf{u}_i^T \boldsymbol{\vartheta}_{(\tilde{r})} \mathbf{u}_i^T. \tag{3.13}$$

In matrix form, Eq. (3.13) becomes

$$\mathbf{U}_{(\tilde{r})}^T \tilde{\mathbf{W}} \mathbf{U}_{(\tilde{r})} \boldsymbol{\vartheta}_{(\tilde{r})} = \mathbf{U}_{(\tilde{r})}^T \tilde{\mathbf{W}} \mathbf{y_o}, \tag{3.14}$$

where $\tilde{\mathbf{W}} = \text{diag}(\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_N)$. Let $\hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})} = (\hat{\vartheta}_1^* \quad \hat{\vartheta}_2^* \quad \ldots \quad \hat{\vartheta}_{\tilde{r}}^*)^T$ be the solution of Eq. (3.15). Hence, we have

$$\mathbf{U}_{(\tilde{r})}^T \tilde{\mathbf{W}} \mathbf{U}_{(\tilde{r})} \hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})} = \mathbf{U}_{(\tilde{r})}^T \tilde{\mathbf{W}} \mathbf{y_o}. \tag{3.15}$$

As mentioned in Section 2, 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. Therefore, $\hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})}$ is obtained by IRLS's algorithm.

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

$$\tilde{\mathbf{y}} := \bar{y} \mathbf{1}_N + \mathbf{K} \boldsymbol{\Gamma}_{(\tilde{r})} \hat{\boldsymbol{\vartheta}}^*_{(\tilde{r})}. \tag{3.16}$$

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

$$\hat{\tilde{\mathbf{e}}}_{\mathbf{1}} := \mathbf{y} - \tilde{\mathbf{y}}, \tag{3.17}$$

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

$$g_{(\tilde{r})}(\mathbf{x}) := \bar{y} + \sum_{i=1}^{N} \tilde{c}_i \kappa(\mathbf{x}, \mathbf{x}_i), \tag{3.18}$$

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

3.2 The R-KPCR's Algorithm

We summarize the procedure in Subsection 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 $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{\tilde{r}} \geq \lambda_{\tilde{r}+1} \geq \ldots \geq \lambda_{\hat{p}_F} > \lambda_{\hat{p}_F+1} = \ldots = \lambda_N = 0$ 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 $\boldsymbol{\alpha}_l = \frac{\mathbf{b}_l}{\sqrt{\lambda_l}}$ for $l = 1, 2, \ldots, \tilde{r}$ and $\boldsymbol{\Gamma}_{(\tilde{r})} = \begin{pmatrix}\boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \ldots & \boldsymbol{\alpha}_{\tilde{r}}\end{pmatrix}$ where $\tilde{r} \in \{1, 2, \ldots, \hat{p}_F\}$.

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

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

   (a) Select the initial estimator of $\boldsymbol{\vartheta}_{(\hat{p}_F)}$, say $\hat{\boldsymbol{\vartheta}}^{*(0)}_{(\hat{p}_F)}$, by OLS.

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

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

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

   (c) Solve the new weighted least squares equations

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

   Step (b) and (c) are repeated until the estimated regression coefficients convergence. Let the estimated regression coefficients convergence at $\hat{\boldsymbol{\vartheta}}^{*(\hat{t})}_{(\tilde{r})} = \begin{pmatrix}\hat{\vartheta}_1^{*(\hat{t})} & \hat{\vartheta}_2^{*(\hat{t})} & \ldots & \hat{\vartheta}_{\tilde{r}}^{*(\hat{t})}\end{pmatrix}^T$

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

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

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

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 $\mathbf{K}_N := \mathbf{K} - \mathbf{EK} - \mathbf{KE} + \mathbf{EKE}$ 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.

The *cross validation* (CV) technique can be used to determine the appropriate $\tilde{r}$ in model (3.18). The CV has a large literature, see for example [6, 10, 12, 19]. In the CV, the original data are partitioned into L disjoint subsets 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 $\boldsymbol{\vartheta}_{(\tilde{r})}$. The CV technique uses the *prediction error sum of squares (PRESS)* to obtain the appropriate $\tilde{r}$, say $\tilde{r}^*$. The *PRESS* of G_k is given by

$$PRESS(G_k)_{(\tilde{r})} = \sum_{s=1}^{m_k} \rho(y_s^k - g_{(\tilde{r})}(\mathbf{x}_s^k)), \tag{3.19}$$

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, say,

$$PRESS_{(\tilde{r})} = \sum_{k=1}^{L} PRESS(G_k)_{(\tilde{r})}. \tag{3.20}$$

The number $\tilde{r}^*$ is chosen such that $PRESS_{(\tilde{r}^*)} \leq PRESS_{(\tilde{r})}$ for $\tilde{r} = 1, 2, \ldots, \hat{p}_F$.

4 Case Study

In the case study, we fix the number of regressors to one and use the Gaussian kernel $\kappa(\mathbf{x}, \mathbf{y}) = \exp(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{\varrho})$ where ϱ is the parameter of the kernel. The toy data were constructed by the function

$$f(x) = 2.5\sin(x), \tag{4.1}$$

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

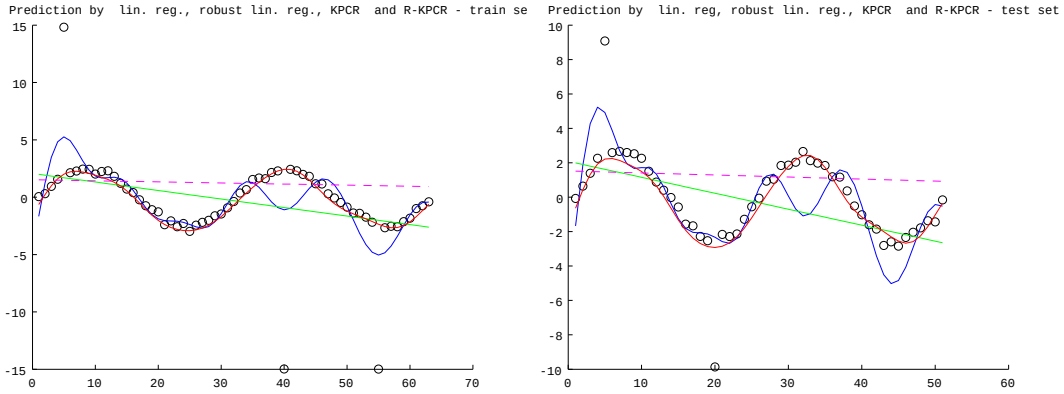
$$y_i = \begin{cases} f(x_{i1}) + \acute{e}_i & \text{if } i \in \{0, 1, \ldots, 62\} \setminus \{5, 40, 55\}, \\ 15 + \acute{e}_5 & \text{if } i = 5, \\ -15 + \acute{e}_{40} & \text{if } i = 40, \\ -15 + \acute{e}_{55} & \text{if } i = 55. \end{cases} \tag{4.2}$$

where $\acute{e}_i, \acute{e}_5, \acute{e}_{40}$ and $\acute{e}_{55}$ are real numbers generated by a normally distributed random with zero mean and standard deviation $\sigma_1 \in [0, 1]$. The set of the data (y_i, x_{i1}) is called the *training data set*. We also generated another set of data for the predictions by ordinary linear regression, robust linear regression, KPCR and R-KPCR. It was also constructed by the Eq. (4.1) with $\hat{x}_{j1} = -2\pi + 0.25 \times j$ for $j = 0, 1, \ldots, 50$; and

$$\hat{y}_j = \begin{cases} f(\hat{x}_{j1}) + \hat{e}_j & \text{if } j \in \{0, 1, \ldots, 50\} \setminus \{5, 20\}, \\ 9 + \hat{e}_5 & \text{if } j = 5, \\ -10 + \hat{e}_{20} & \text{if } j = 20, \end{cases} \tag{4.3}$$

where $\hat{e}_j, \hat{e}_5$ and $\hat{e}_{20}$ 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 $(\hat{y}_j, \hat{x}_{j1})$ is called the *testing data set*. For shake of comparisons, we set σ_1 and σ_2 are equal to 0.2 and 0.25, respectively.

To test the performance of the four methods, we generated 10000 sets of the training data and the testing data. For shake of comparisons, the value of $\tilde{r}$ is chosen to be equal the retained number of nonlinear PCs for the KPCR, say $\hat{r}$. A plot of the predictions of the four methods corresponding to the toy data are given in Figure 1. The average of RMSEs of the four methods are shown in Table 1. In comparing to ordinary linear regression, robust linear regression and KPCR, the R-KPCR yields the better results as shown in Table 1.



(a) (b)

Figure 1: 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 stars are the toy data by adding the random noise: (a) training data, (b) testing data.

Table 1: The comparison of ordinary linear regression, robust linear regression, KPCR and R-KPCR.

	Model	RMSE	
		Training	Testing
Tukey biweighted function	ordinary linear regression	3.36965128586967	2.48160865268650
	robust linear regression	0.16562014395649	0.33437227269687
	KPCR ($\rho = 2.5, \hat{r} = 14$)	2.81004003365153	1.90467636266656
	KPCR ($\rho = 5, \hat{r} = 10$)	2.84127583904498	1.75054139256838
	R-KPCR ($\rho = 2.5, \tilde{r} = 14$)	0.16532672128538	0.17193946920348
	R-KPCR ($\rho = 5, \tilde{r} = 10$)	0.16412845968267	0.16705693873425
Huber function	ordinary linear regression	3.40241441626446	2.48915962165389
	robust linear regression	1.66326272704349	1.56821960764758
	KPCR ($\rho = 2.5, \hat{r} = 14$)	2.787507237923666	1.899546619671557
	KPCR ($\rho = 5, \hat{r} = 10$)	2.79447892213644	1.88776460800414
	R-KPCR ($\rho = 2.5, \tilde{r} = 14$)	1.152228759181353	0.725512925314396
	R-KPCR ($\rho = 5, \tilde{r} = 10$)	1.14679840289139	0.72096269525088

5 Conclusion

Robust regression is a technique in regression analysis to eliminate the effects of outliers. If the outliers are contained in the observed data, both prediction of ordinary linear regression and KPCR can be inappropriate to be used. We noticed that Fomengko *et al.* [5] 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, the proposed method has limitations in applications.

In this paper, we proposed the R-KPCR to obtain a nonlinear robust prediction where our proposed method does not need to specify a nonlinear model in advance. Our case study showed that R-KPCR yields the better result that of ordinary linear regression, robust linear regression and KPCR.

Acknowlegdement
The author thank Professor Yoshitsugu Yamamoto, University of Tsukuba, for comments and suggestions. The author also thank the Ministry of Education, Culture, Sports, Science and Technology Japan.

References

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

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

[3] R.J. Caroll and D. Ruppert. *Transformation and Weighting in Regression*. Chapman and Hall, 1988.

[4] 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.

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

[6] Julian J. Faraway. *Linear Models with R*. Chapman and Hall/CRC, 2005.

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

[8] P. Huber. *Robust Statistics*. John Wiley and Son Inc, 1981.

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

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

[11] 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.

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

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

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

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

[16] 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.

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

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

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

[20] 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.

[21] 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.

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