Nonlinear Robust Regression Using Kernel Principal Component Analysis and R-Estimators

Antoni Wibowo and Mohammad Ishak Desa

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor Malaysia

Abstract

In recent years, many algorithms based on kernel principal component analysis (KPCA) have been proposed including kernel principal component regression (KPCR). KPCR can be viewed as a non-linearization of principal component regression (PCR) which uses the ordinary least squares (OLS) for estimating its regression coefficients. We use PCR to dispose the negative effects of multicollinearity in regression models. However, it is well known that the main disadvantage of OLS is its sensitiveness to the presence of outliers. Therefore, KPCR can be inappropriate to be used for data set containing outliers. In this paper, we propose a novel nonlinear robust technique using hybridization of KPCA and R-estimators. The proposed technique is compared to KPCR and gives better results than KPCR.

Keywords: *Kernel principal component analysis, kernel principal component regression, robustness, nonlinear robust regression, R-estimators.*

1. Introduction

Kernel principal component analysis (*KPCA*) has been proposed to be used for nonlinear systems by mapping an original input space into a higher-dimensional feature space, see [3, 9, 17, 18] for the detailed discussion, and becomes an attractive algorithm because it does not involve nonlinear optimization, it is as simple as the *principal component analysis* (*PCA*) and it does not need to specify the number of principal components prior to modeling compared to other nonlinear methods. In recent years, many nonlinear algorithms based on KPCA have been proposed including *kernel principal component regression* (*KPCR*) which can be viewed as a non-linearization of *principal component regression* (*PCR*) and dispose the effects of multicollinearity in regression models [6, 8, 14, 15, 16, 22]. However, KPCR was constructed based on the *ordinary least squares* (*OLS*) for estimating its regression coefficients which was sensitive to the presence of outliers. An observation is called outlier if it does not follow the OLS based linear regression model. When we use OLS to estimate regression coefficients then outliers have a large influence to the prediction values since squaring residuals magnifies the effect of the outliers. Therefore, KPCR can be inappropriate to be used when outliers are present.

In previous years, several techniques have been developed to overcome the negative effects of outliers such as *R-estimators* which was a robust method based on the ranks of the residuals [7, 11]. However, the previous works applied it for tackling the effect of outliers in the linear regression model. We should notice that the estimate of regression coefficients using R-estimators is obtained through solving a nonlinear optimization problem. To obtain the estimate of regression coefficients, we can use several techniques for solving this nonlinear optimization problems such as genetic algorithms (GAs), simulated annealing and particle swarm optimization [2, 4, 5, 12, 13, 21, 23]. However, applying R-estimators in the ordinary regression still yields linear models which have limitations in applications.

In this paper, we propose a novel robust technique using hybridization of KPCA and R-estimators to overcome the limitation of KPCR and R-estimators in the linear regression. We use KPCA to dispose the effects of multicollinearity in regression and to construct nonlinearity of prediction model by transforming original data into a higher-dimensional feature space and perform a kernel trick to have a multiple linear regression in this space. Then, we perform R-estimators in this linear regression and solve the optimization problems of the R-estimators for obtaining a nonlinear robust regression. We refer the proposed method as the *robust kernel principal component R regression* (*R-KPCRR*).

We organize the rest of manuscript as follows: Section 2, we review theories and methods of R-estimators and

KPCA, followed by R-KPCRR and its algorithm. In Section 3, we compare the performance of the proposed method using several data sets. Finally, conclusions are given in Section 4.

2. Theories and Methods

2.1 R-Estimators

The ordinary multiple linear regression model is given by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{2.1}$$

where
$\mathbf{y} = \begin{pmatrix} y_1 & y_2 & \cdots & y_N \end{pmatrix}^T \in \mathbf{R}^N$, $\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \end{pmatrix} \in \mathbf{R}^{N \times p}$,
$\mathbf{x}_i = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{iN} \end{pmatrix}^T \in \mathbf{R}^p$, $\mathbf{X} = \begin{pmatrix} \mathbf{1}_N & \tilde{\mathbf{X}} \end{pmatrix} \in \mathbf{R}^{N \times (p+1)}$
with $\mathbf{1}_N$ is $N \times 1$ vector with all elements equal to one, $\boldsymbol{\beta} = \begin{pmatrix} \beta_0 & \beta_1 & \cdots & \beta_p \end{pmatrix}^T \in \mathbf{R}^{p+1}$ is a vector of regression coefficients, $\mathbf{e} = \begin{pmatrix} e_1 & e_2 & \cdots & e_N \end{pmatrix}^T \in \mathbf{R}^N$ is vector of residuals and $\mathbf{R}$ is the set of real numbers and i = 1, 2, …, N.

When we use OLS to find the estimate of $\boldsymbol{\beta}$, say $\hat{\boldsymbol{\beta}}$, then the estimate is found by minimizing

$$\sum_{i=1}^{N} e_i^2, \tag{2.2}$$

where $e_i = y_i - \dot{\mathbf{x}}_i^T \boldsymbol{\beta}$ and $\dot{\mathbf{x}}_i^T = \begin{pmatrix} 1 & \mathbf{x}_i^T \end{pmatrix}$. The solution can be found by solving the following linear equation

$$\mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}} = \mathbf{X}^T \mathbf{y} \tag{2.3}$$

However, it is well known that the prediction of the OLS based regression will be distorted when outliers are present. To overcome the presence of outliers, we can use R-estimators which minimize

$$\sum_{i=1}^{N} a_N(R_i) e_i \tag{2.4}$$

where R_i is the rank of e_i and $a_N(i)$ is a score function which is monotone and satisfies $\sum_{i=1}^{N} a_N(i) = 0$. The common choice of $a_N(i)$ are

$$a_N(i) = i - (N+1)/2 \tag{2.5a}$$

and

$$a_N(i) = \Theta^{-1}\left(i/(N+1)\right) \tag{2.5b}$$

which are called the Wilcoxon and Van der Waerden scores, respectively, with Θ^{-1} is the inverse of cumulative normal distribution function.

2.2 KPCA

Assume that we have a function $\psi : \mathbf{R}^p \to \mathbf{F}$, where $\mathbf{F}$ is the feature space which it is an Euclidean space with dimension p_F ($p_F \geq p$). Then, we define the matrices $\mathbf{C} = (1/N)\boldsymbol{\Psi}^T\boldsymbol{\Psi} \in \mathbf{R}^{p_F \times p_F}$ and $\mathbf{K} = \boldsymbol{\Psi}\boldsymbol{\Psi}^T \in \mathbf{R}^{N \times N}$ where $\boldsymbol{\Psi} = \begin{pmatrix} \psi(\mathbf{x}_1) & \psi(\mathbf{x}_2) & \ldots & \psi(\mathbf{x}_N) \end{pmatrix}^T \in \mathbf{R}^{N \times p_F}$ and assume that $\sum_{i=1}^{N} \psi(\mathbf{x}_i) = \mathbf{0}$. The relation of eigenvalues and eigenvectors of the matrices $\mathbf{C}$ and $\mathbf{K}$ were studied by Scholkopf *et al.* [18] .

Let $\hat{\rho}_F$ be the rank of $\boldsymbol{\Psi}$ where $\hat{\rho}_F \leq \min(N, p_F)$ which implies that both rank($\mathbf{K}$) and rank($\boldsymbol{\Psi}^T\boldsymbol{\Psi}$) are equal to $\hat{\rho}_F$. It is evident that the eigenvalues of $\mathbf{K}$ are nonnegative real numbers since the matrix $\mathbf{K}$ is symmetric and positive semidefinite [1]. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{\tilde{r}} \geq \lambda_{\tilde{r}+1} \geq \cdots \geq \lambda_{\hat{\rho}_F} > \lambda_{\hat{\rho}_F+1} = \cdots = \lambda_N = 0$ be the eigenvalues of $\mathbf{K}$ and $\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \ldots & \mathbf{b}_N \end{pmatrix}$ be the matrix of the corresponding normalized eigenvectors $\mathbf{b}_s$ (s =1, 2, …, N) of $\mathbf{K}$. Then, let $\boldsymbol{\alpha}_l = \mathbf{b}_l / \sqrt{\lambda_l}$ and $\mathbf{a}_l = \boldsymbol{\Psi}^T \boldsymbol{\alpha}_l$ for l = 1, 2, …, $\hat{\rho}_F$. The eigenvectors $\mathbf{a}_l$, 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 a *kernel trick*. The l-th principal component of $\psi(\mathbf{x})$ (l = 1, 2, …, $\hat{\rho}_F$) as given as follows:

$$\psi(\mathbf{x})^T \mathbf{a}_l = \sum_{i=}^{N} \alpha_{li} \psi(\mathbf{x})^T \psi(\mathbf{x}_i) \tag{2.6a}$$

where α_{li} is the i-th element of $\boldsymbol{\alpha}_l$. According to Mercer Theorem, if we choose a continuous, symmetric and positive semidefinite kernel $\kappa : \mathbf{R}^P \times \mathbf{R}^P \to \mathbf{R}$ then there exists $\varphi : \mathbf{R}^P \to \mathbf{F}$ such that $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$ [10, 17]. Instead of choosing ψ explicitly, we choose a kernel κ and employ the

corresponding function φ as ψ. Let $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ then $\mathbf{K}$ and $\boldsymbol{\alpha}_l$ (l = 1, 2, …, $\hat{p}_F$) are explicitly known now. Therefore, Eq. (2.6a) is also explicitly known and can be written as

$$\psi(\mathbf{x})^T \mathbf{a}_l = \sum_{i=}^{N} \alpha_{li} \kappa(\mathbf{x}, \mathbf{x}_i) \qquad (2.6b)$$

2.3 Nonlinear Robust Regression Using KPCA and R-Estimators

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

$$\mathbf{y}_0 = \boldsymbol{\Psi}\boldsymbol{\gamma} + \tilde{\mathbf{e}} \qquad (2.7)$$

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, $\tilde{\mathbf{e}}$ is a vector of random errors and $\mathbf{y}_0 = \left(\mathbf{I}_N - (1/N)\mathbf{1}_N\mathbf{1}_N^T\right)\mathbf{y}$ where $\mathbf{I}_N$ is the $N \times N$ identity matrix.

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. Let λ_k $(k = \hat{p}_F + 1, \hat{p}_F + 2, ..., p_F)$ 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. Furthermore, we define $\mathbf{A} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_{p_r} \end{pmatrix}$. 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

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

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

and $\mathbf{O}$ is a zero matrix. By using $\mathbf{A}\mathbf{A}^T = \mathbf{I}_{p_r}$, we can rewrite the model (2.7) as

$$\mathbf{y}_0 = \mathbf{U}\boldsymbol{\vartheta} + \tilde{\mathbf{e}} \qquad (2.8)$$

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

$$\mathbf{y}_0 = \mathbf{U}_{(\hat{p}_F)}\boldsymbol{\vartheta}_{(\hat{p}_F)} + \mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)} + \tilde{\mathbf{e}}. \qquad (2.9)$$

It is easy to verify that $(\mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)})^T\mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)} = 0$ which implies $\mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)}$ is equal to $\mathbf{0}$. Consequently, the model (2.9) reduces to

$$\mathbf{y}_0 = \mathbf{U}_{(\hat{p}_F)}\boldsymbol{\vartheta}_{(\hat{p}_F)} + \tilde{\mathbf{e}}. \qquad (2.10)$$

where $\mathbf{U}_{(\hat{p}_F)} = \boldsymbol{\Psi}\mathbf{A}_{(\hat{p}_F)} = \mathbf{K}\boldsymbol{\Gamma}_{(\hat{p}_F)}$ and $\boldsymbol{\Gamma}_{(\hat{p}_F)} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_{p_F})$. It is evident that the elements of $\mathbf{U}_{(\hat{p}_F)}$ are the principal components of $\psi(\mathbf{x}_i)$ for i = 1, 2, …, N. Then, if we only use the first $\hat{r}$ $(\leq \hat{p}_F)$ vectors of $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$, $\dots, \boldsymbol{\alpha}_{\hat{p}_F}$, model (2.10) becomes

$$\mathbf{y}_0 = \mathbf{U}_{(\hat{r})}\boldsymbol{\vartheta}_{(\hat{r})} + \boldsymbol{\varepsilon}, \qquad (2.11)$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_N)^T$ is a vector of residuals influenced by dropping the term $\mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)}$ in model (2.11), respectively. We usually dispose of the term $\mathbf{U}_{(p_F - \hat{p}_F)}\boldsymbol{\vartheta}_{(p_F - \hat{p}_F)}$ for tackling the effects of multicollinearity on the PCA based regressions where the number $\hat{r}$ is called the retained number of nonlinear principal components (PCs) for the KPCR. We can use the ratio λ_l/λ_1 (l = 1, 2, …, $\hat{p}_F$) for detecting the presence of multicollinearity on $\mathbf{U}_{(\hat{r})}$. If λ_l/λ_1 is smaller than, say < 1/1000, then we consider that multicollinearity exists on $\mathbf{U}_{(\hat{r})}$ [11].

Let us consider model (2.11) again. We can see that model (2.11) has the same structure with model (2.1) which implies that we can directly apply R-estimators in model (2.11). For this purpose, we define $\mathbf{U}_{(\hat{r})} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_N)^T \in \mathbf{R}^{N \times \hat{r}}$ and obtain $\varepsilon_i = y_{oi} - \mathbf{u}_i^T\boldsymbol{\vartheta}_{(\hat{r})}$. Then, we minimize

$$\sum_{i=1}^{N} a_N\left(\tilde{R}_i\right)\varepsilon_i, \qquad (2.15)$$

To find the estimators of $\boldsymbol{\vartheta}_{(\hat{r})}$, where $\tilde{R}_i$ is the rank of ε_i, by using a nonlinear optimization solver .

Let $\hat{\boldsymbol{\vartheta}}^{*}_{(\hat{r})}$ be the estimator of $\boldsymbol{\vartheta}_{(\hat{r})}$ using the above R-estimators. Then, the prediction value of $\mathbf{y}$ *with the first* $\hat{r}$ *vectors of* $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_{\hat{\rho}_r}$ *using R-estimators*, say $\tilde{\mathbf{y}}$, is given by

$$\tilde{\mathbf{y}} = \bar{y}\mathbf{1}_N + \mathbf{K}\boldsymbol{\Gamma}_{(\hat{r})}\hat{\boldsymbol{\vartheta}}^{*}_{(\hat{r})} \tag{2.16}$$

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

$$\tilde{\boldsymbol{\varepsilon}} = \mathbf{y} - \tilde{\mathbf{y}}. \tag{2.17}$$

The *prediction of the R-KPCRR with the first* $\hat{r}$ *vectors of* $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_{\hat{\rho}_r}$ is given by

$$g_{(\hat{r})}(\mathbf{x}) = \bar{y} + \sum_{i=1}^{N} d_i \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_i) \tag{2.18}$$

where $(d_1 \;\; d_2 \;\; \cdots \;\; d_N)^T = \boldsymbol{\Gamma}_{(\hat{r})}\hat{\boldsymbol{\vartheta}}^{*}_{(\hat{r})}$ and $g_{(\hat{r})}$ is a function from $\mathbf{R}^N$ into $\mathbf{R}$.

We summarize the above procedures of the R-KPCRR as follows:

1 . Given $(y_i \;\; x_{i1} \;\; x_{i2} \;\; \cdots \;\; x_{iN})$ for i =1, 2, . . . , N.

2. Calculate $\bar{y} = (1/N)\mathbf{1}_N^T\mathbf{y}$ and $\mathbf{y}_0 = (\mathbf{I}_N - (1/N)\mathbf{1}_N\mathbf{1}_N^T)\mathbf{y}$.

3. Choose a kernel $\kappa : \mathbf{R}^P \times \mathbf{R}^P \to \mathbf{R}$ and a function $a_N : \mathbf{R} \to \mathbf{R}$.

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

5. Diagonalize $\mathbf{K}$.
Let rank($\mathbf{K}$) = $\hat{\rho}_F$ and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{\hat{r}} \geq \lambda_{\hat{r}+1} \geq \cdots \geq \lambda_{\hat{\rho}_r} > \lambda_{\hat{\rho}_r+1} = \ldots = \lambda_n = 0$ be the eigenvalues of $\mathbf{K}$ and $\mathbf{b}_1 \;\; \mathbf{b}_2 \;\; \ldots \;\; \mathbf{b}_N$ be the corresponding normalized eigenvectors $\mathbf{b}_s$ (s =1, 2, …,N) of $\mathbf{K}$.

6. Choose $\hat{r}$ $(\leq \hat{\rho}_F)$ and construct $\boldsymbol{\alpha}_l = \mathbf{b}_l / \sqrt{\lambda_l}$ for l = 1, 2, …, $\hat{r}$. Then, define $\boldsymbol{\Gamma}_{(\hat{r})} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_{\hat{r}})$.

7. Calculate $\mathbf{U}_{(\hat{r})} = \mathbf{K}\boldsymbol{\Gamma}_{(\hat{r})}$ and let $\varepsilon_i = y_{oi} - \mathbf{u}_i^T\boldsymbol{\vartheta}_{(\hat{r})}$.

8. Let $\tilde{R}_i$ be the rank of ε_i.

9. Solve problem (2.15) using a nonlinear optimization solver and let $\hat{\boldsymbol{\vartheta}}^{*}_{(\hat{r})}$ be solution of (2.15).

10. Calculate $(d_1 \;\; d_2 \;\; \cdots \;\; d_N)^T = \boldsymbol{\Gamma}_{(\hat{r})}\hat{\boldsymbol{\vartheta}}^{*}_{(\hat{r})}$.

11. Given a vector $\mathbf{x} \in \mathbf{R}^P$, the prediction of the R-KPCRR with the first $\hat{r}$ vectors of $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_{\hat{\rho}_r}$ is given by

$$g_{(\hat{r})}(\mathbf{x}) = \bar{y} + \sum_{i=1}^{N} d_i \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_i).$$

We should note that this 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 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 1/N.

Table 1: 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		

3. Case Studies

3.1 Data Sets

We generated data sets from a trigonometric function and sinc function to test the performances of KPCR and R-KPCRR. The generated data from the trigonometric function and sinc function are given as follows:

$$f(x) = 4.5\sin(2x) + 2.5\cos(x), \tag{3.1}$$

with $x_i \in [-2\pi : 0.15 : 2\pi]$ and $x_{tj} \in [-2\pi : 0.2 : 2\pi]$;

$$f(x)=\begin{cases}5\sin(|x|)/|x| & \text{if } x\neq 0\\ 5 & \text{otherwise.}\end{cases} \quad (3.2)$$

with $x_i \in [-8:0.25:8]$ and $x_{tj} \in [-6:0.3:6]$, respectively. The notation $[-z:l:z]$ stands for $[-z, z+l, z+2l, \cdots, z]$ where l is a real number.

Generally, the generated data from the those functions can be written as $y_i = f(x_i) + e_i$ i where i = 1, 2,· · ·, N. We also generate $y_{tj} = f(x_{tj}) + e_{tj}$ where j = 1, 2, · · ·,N_t; where N_t is a positif integer. The random noises e_i and e_{tj} are real numbers generated by a normally distributed random with zero mean and standard deviation σ_1 and σ_2, respectively, with $\sigma_1, \sigma_2 \in [0,1]$. For shake of comparisons, we set σ_1 and σ_2 equal to 0.2 and 0.3, and call the set of $\{(x_i, y_i)\}$ and $\{(x_{tj}, y_{tj})\}$ *the training data set* and *the testing data set*, respectively. In addition, we also used a subset of the famous set of observation taken on the height of the son of the Count de Montbeillard between 1959 and 1977 [19]. Only the first ten years of data were used in this analysis. The growth of son data are given in the Table 1. In these data, we artificially generate the testing data by the relation $x_{jt} = 0.1 \times rand(1) + \mathrm{x_j}$ and $y_{jt} = 0.25 \times rand(1) + \mathrm{y_j}$ where $rand(1)$ generates a random number which is uniformly distributed in the interval (0,1).

Then, we compare the performance of the above methods using the three data sets with and without outliers. For this purpose, we generated 200 sets of the training data and 200 sets of the testing data. Furthermore, we use the *mean absolute error* (*MAE*) to estimate the prediction error for the training data set which is given by

$$\mathrm{MAE} = (1/N)\sum_{i=1}^{N} |y_i - \tilde{y}_i|. \quad (3.3)$$

The MAE is also used to prediction error of the testing data sets and denoted by MAEt.

In this case studies, outliers are created artificially by moving some (x_i, y_i)s and (x_{tj}, y_{tj})s away from designated locations. We generate eight potential outliers for each of the first, second and third data sets where the positions of outliers in x-direction and x_t-direction are chosen randomly in the domain of x_i and domain of x_{tj}, respectively. The positions of outliers in y-direction and y_t-direction are randomly selected in interval [−20, 20] from the correct positions of y_i and y_{tj}, respectively.

3.2 Results

In these case studies, we used the Wilcoxon and Van der Waerden scores for R-estimators and the standard genetic algorithm (GA) for solving the optimization problem of R-estimators. Then, we used the Gaussian kernel $\kappa(\mathbf{x},\mathbf{z}) = \exp(-\|\mathbf{x}-\mathbf{z}\|/\rho)$ with parameter ρ is equal to five for both KPCR and R-KPCRR. We involved the estimate of $\vartheta_{(\hat{r})}$ by using KPCR, say $\hat{\vartheta}_{(\hat{r})}$, in the initial population of GA. In the intial population, the i-th gene of the other chromosomes (or candidate solutions of $\vartheta_{(\hat{r})}$) is randomly chosen by the formulae

$$\left(\hat{\vartheta}_{(\hat{r})}\right)_i + 30 \times rand(1) - 15 \quad (3.4)$$

where $\left(\hat{\vartheta}_{(\hat{r})}\right)_i$ is the i-th element of $\hat{\vartheta}_{(\hat{r})}$ and i = 1, 2, …, $\hat{r}$. For the sake of comparisons, the numbers of population, maximum iterations, mutation rate and selection rate are 50, 1000, 0.2 and 0.5, respectively. For each chromosome in any population we sort ε_i in the descending order, say $\varepsilon_{[i]}$ and $\varepsilon_{[1]} \geq \varepsilon_{[2]} \geq \cdots \geq \varepsilon_{[N]}$, to determine its rank. Then, we define $c_k = \varepsilon_{[\lfloor 0.2 \times k \times N \rfloor]}$ where k=1, 2, 3, 4 and rank ε_i is given by

$$\tilde{R}_i = \begin{cases} 5 & \text{if } c_1 \leq \varepsilon_i \\ 4 & \text{if } c_2 \leq \varepsilon_i < c_1 \\ 3 & \text{if } c_3 \leq \varepsilon_i < c_2 \\ 2 & \text{if } c_4 \leq \varepsilon_i < c_3 \\ 1 & \text{if } \varepsilon_i < c_4. \end{cases} \quad (3.5)$$

As the results, the three plots of the predictions of KPCR and R-KPCRR corresponding to the three data sets are presented in Figure 1, Figure 2 and Figure 3, respectively. We can see that the predictions of R-KPCRR are less distorted by the presence of outliers compared to KPCR. Table 2 illustrates the prediction errors of KPCR and R-KPCRR. In the case of data with outliers, R-KPCRR with Wilcoxon and Van der Waeden scores give lower MAEs and MAEts compared to KPCR. The MAEs' R-KPCRR with Wilcoxon scores for the trigometric, sinc and growth of son are 1.4527, 1.4250 and 4.8730 whereas the corresponding MAEts are 1.1363, 1.6742 and 6.5350,

respectively. The MAEs' R-KPCRR with Van der Waeden score for the trigometric, sinc and growth of son are 1.4774, 1.4609 and 4.9876 whereas the corresponding MAEts are 1.1257, 1.7709 and 6.5776, respectively. Table 2 summarizes MAEs and MAEts of the three data sets without outliers. In this case, we can see that both KPCR and R-KPCRR perform well.

Table 2: MAE and MAEt for KPCR and R-KPCRR with outliers (Wilc=Wilcoxon, VDW=Van der Waerden).

Data	**Method**	**MAE**	**MAEt**
Trigono-Metric ($\hat{r}=10$)	KPCR	1.8804	1.6970
	R-KPCRR Wilc	1.4527	1.1363
	R-KPCRR VDW	1.4774	1.1257
Sinc ($\hat{r}=13$)	KPCR	2.3831	3.3284
	R-KPCRR Wilc	1.4250	1.6742
	R-KPCRR VDW	1.4609	1.7709
Growth of Son ($\hat{r}=14$)	KPCR	5.0125	7.0113
	R-KPCRR Wilc	4.8730	6.5350
	R-KPCRR VDW	4.9876	6.5776

Table 3: MAE and MAEt for KPCR and R-KPCRR without outliers (Wilc=Wilcoxon, VDW=Van der Waerden).

Data	**Method**	**MAE**	**MAEt**
Trigono-Metric ($\hat{r}=10$)	KPCR	0.0822	0.0823
	R-KPCRR Wilc	0.0821	0.0829
	R-KPCRR VDW	0.0814	0.0823
Sinc ($\hat{r}=13$)	KPCR	0.0725	0.0610
	R-KPCRR Wilc	0.0733	0.0613
	R-KPCRR VDW	0.0729	0.0614
Growth of Son ($\hat{r}=14$)	KPCR	0.9663	0.9840
	R-KPCRR Wilc	0.9172	0.9305
	R-KPCRR VDW	0.9229	0.9379

4. Conclusions

We have proposed a novel robust regression using the hybridization of KPCA and R-estimators. Our method yields a nonlinear robust prediction and can dispose the effects of multicollinearity in regression model. The proposed method was performed by transforming original data into a higher dimensional feature space and creating a multiple linear regression in the feature space. After that, we performed a kernel trick to have an explicit multiple linear regression and used R-estimators on this linear model to have a robust regression. Then, we solved the optimization problem of R-estimators using GA for obtaining the estimate of regression coefficients. In this paper, we used Wilcoxon and Van der Waerden scores on R-estimators.

We summarized several important points relating to our cases studies. First, the predictions of R-KPCRR are less distorted and give smaller MAEs and MAEts compared to KPCR when outliers are present in thedata. Second, without outliers, both R-KPCRR and KPCR perform equally well.

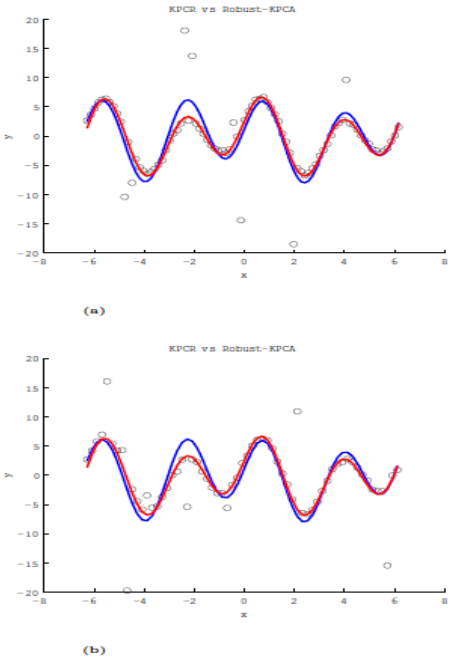

Figure 1: KPCR (Black) and R-KPCRR (red) using Wilcoxon scores with ρ and $\hat{r}$ equal to 5 and 10, respectively. The black circles are trigonometric data with random noises: (a) training data, (b) testing data.

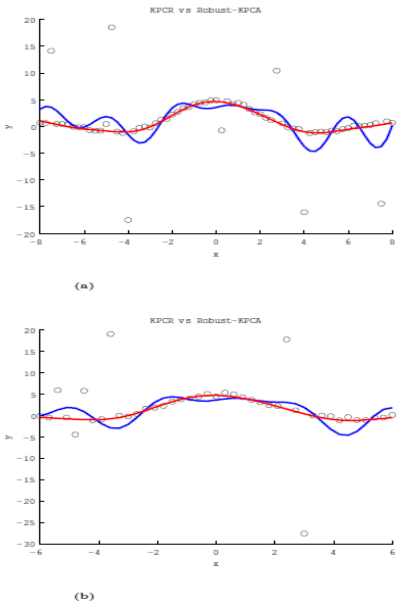

Figure 2: KPCR (Black) and R-KPCRR (red) using Van der Waerden scores with ρ and $\hat{r}$ equal to 5 and 13, respectively. The black circles are sinc data with random noises: (a) training data, (b) testing data.

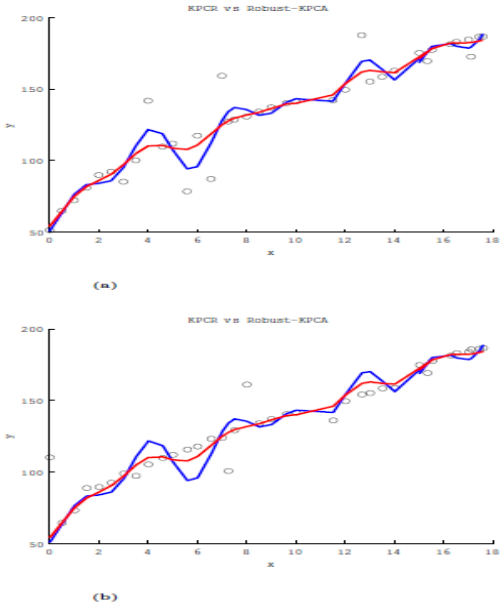

Figure 3: KPCR (Black) and R-KPCRR (red) using Wilcoxon scores with ρ and $\hat{r}$ equal to 5 and 13, respectively. The black circles are the growth of son data with random noises: (a) training data, (b) testing data.

Acknowledgments

The authors sincerely thank to Universiti Teknologi Malaysia and Ministry of Higher Education (MOHE) Malaysia for Research University Grant (RUG) with vot number Q.J130000.7128**.** We also thank to The Research Management Center (RMC) – UTM for supporting this research project.

References

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

[2] M.B. Aryanezhad and M. Hemati, "A new genetic algorithm for solving nonconvex nonlinear programming problems", Applied Mathematics and Computation, 86:186–194, 2008.

[3] 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, pages 279–288, 2005.

[4] M. Gen, R. Cheng, and L. Lin, Network Models and Optimization Multiobjective Genetic ALgorithm Approach, Spinger, 2008.

[5] R.L. Haupt and S.E. Haupt, Practical Genetic Algorithms, John Wiley and Sons, 2004.

[6] 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.

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

[8] 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.

[9] C. Lu, C. Zhang, T. Zhang, and W. Zhang, "Kernel based symmetrical principal component analysis for face classification", Neurocomputing, 70:904–911, 2007.

[10] H. Q. Minh, P. Niyogi, and Y. Yao, "Mercer's theorem, feature maps, and smoothing", Lecture Notes in Computer Science, Springer Berling, 4005/2006, 2009.

[11] D. C. Montgomery, E. A. Peck, and G. G. Vining, Introduction to Linear Regression, Wiley-Interscience, 2006.

[12] M.S. Osman, Mahmoud A. Abo-Sinn, and A.A. Mousa, "A combined genetic algorithm-fuzzy logic controller (ga-fls) in nonlinear programming", Applied Mathematics and Computation, 170:821–840, 2005.

[13] C. H. Park, W. I. Lee, W. Suck, and A. Vautrin, "Improved genetic algorithm for multidisciplinary optimization of composite laminates", Chemometrics and Intelligent Laboratory Systems, 68:1894–1903, 2008.

[14] R. Rosipal, M. Girolami, L. J. Trejo, and A. Cichoki, "Kernel pca for feature extraction and de-noising in nonlinear regression", Neural Computing and Applications, pages 231–243, 2001.

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

[16] R. Rosipal, L. J. Trejo, and A. 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] B. Scholkopf and A.J. Smola, Learning with kernels, The MIT Press., 2002.
[19] G.A.F. Seber and C.J. Wild, Nonlinear Regression, John Wiley and Sons, Inc., 1998.
[20] S.N.Sivanandam and S.N.Deepa, Introduction to Genetic Algorithms, Springer, 2008.
[21] S. Sumathi, T. Hamsapriya, and P. Surekha, Evolutionary Intelligence, Springer, 2008.
[22] A. Wibowo and Y. Yamamoto, "A note of kernel principal component regression", To appear in Computational Mathematics and Modeling, Springer, 2011.
[23] X. Yu and M. Gen, Introduction to Evolutionary Algorithms, Springer, 2010.

Antoni Wibowo is currently working as a senior lecturer in the faculty of computer science and information systems, UTM. He received B.Sc in Math Engineering from University of Sebelas Maret (UNS) Indonesia and M.Sc in Computer Science from University of Indonesia. He also received M. Eng and Dr. Eng in System and Information Engineering from University of Tsukuba Japan. His interests are in the field of computational intelligence , operations research and data analysis.

Mohamad Ishak Desa is a professor in the faculty of computer science and information systems, UTM. He received his B.Sc. in Mathematics from UKM in Malaysia, a postgraduate diploma in system analysis from Aston University, UK. He also received a M.A. in Mathematics from University of Illinois at Springfield, USA and then, a PhD in operations research from Salford University, UK. He is currently the Head of the Operations and Business Intelligences Research Group in UTM. His interests are operations research, optimization, logistic and supply chain, and computational intelligence.