Journal of Computational Information Systems 10: 17 (2014) 7489–7497
Available at http://www.Jofcis.com

A Fixed-point Iteration Algorithm for Robust Kernel Principal Component Analysis ⋆

Shengkai YANG, Haibin SHEN*, Jianyi MENG, Zhijian CHEN

Institute of VLSI Design, Zhejiang University, Zhejiang 310027, China

Abstract

Kernel principal component analysis (KPCA) has been widely used in machine learning and pattern recognition. However, the classical KPCA is quite sensitive to outliers. In this paper a robust KPCA algorithm is proposed. We follow the observation in the classical KPCA that the eigenvectors and central point can be expressed as linear combination of training samples. Borrowing the idea from robust statistics, a robust loss function is induced. Finally the solution can be obtained in an iterative way. The proposed algorithm can make robust estimation of both the central point and the eigenvectors. The form of the solution to the proposed algorithm is of the same as that to the classical KPCA algorithm, thus it is easy to integrate into existing systems. Experimental results show that the proposed algorithm is robust and achieves better performance than the classical KPCA.

Keywords: Kernel Principal Component Analysis (KPCA); Robust KPCA; Outliers; Nonlinear Denoising

1 Introduction

Principal component analysis (PCA) is a multivariate statistical method which has been widely used in pattern recognition and machine learning. The main factors can be parsed out from extensive data, thus the problems to be solved are simplified. There are two main drawbacks in classical PCA: it is only applicable for linearly dependent data, and it lacks robustness. To solve the former drawback, KPCA was proposed to introduce nonlinearity by projecting the original data into a high-dimensional Hilbert space [1]. However, the robustness issue is still involved in KPCA. It is sensitive to outliers, which may lead to offset principal components. Many robust KPCA methods have been proposed, and they will be discussed in Section 2.

KPCA has been widely used in many occasions since it is proposed. The most frequent application is feature extraction for classification [2, 3]. Because of the nonlinear characteristic of the algorithm, the underlying key components and structures can be extracted for further classification. Another scenario is signal denoising [4, 5, 6], where KPCA is used to remove nonlinear

⋆Project supported by National Science and Technology Major Project (No. 2009ZX01031-001-007-2).
*Corresponding author.
Email address: haibinshen66@gmail.com (Haibin SHEN).

1553–9105 / Copyright © 2014 Binary Information Press
DOI: 10.12733/jcis11487
September 1, 2014

noise. Other applications include image reconstruction [7], statistical shape analysis [8], novelty detection [9], etc. Note that in all the above applications, if there are outliers in the training dataset, it may lead to unsatisfactory results, since the classical KPCA is not robust and outliers will result in undesirable deviation of principal components.

In this paper, a robust KPCA algorithm is proposed. In this algorithm we follow the observation in classical KPCA that both the central point and eigenvectors lie in the subspace spanned by training samples in feature space [1]. Thus the central points and eigenvectors can be represented by training samples. By introducing a more robust cost function, the problem can be solved in an iterative way.

The proposed algorithm has two appealing advantages: (1) it is robust to outliers. The central point and eigenvectors are both obtained under robust estimation; (2) the form of the solutions to this algorithm is the same as that of the classical KPCA. This means the proposed robust KPCA algorithm can be merged easily to existing systems involving classical KPCA algorithms without any modification.

The rest of this paper is organized as follows. In Section 2 a brief introduction to the classical KPCA and previous studies on robust KPCA are presented. The proposed robust KPCA is described in Section 3. Experimental results are shown in Section 4, along with the discussion. Conclusions are given in Section 5.

2 Related Studies

2.1 The classical KPCA algorithm

As the kernel extension of PCA, KPCA has showed good performance in many applications. A brief introduction is given here, and for detailed description, the readers could refer to [1].

Denote the d-dimensional training samples in input space as $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. The mapping used in KPCA is denoted as ϕ. Thus the samples in feature space is $\boldsymbol{\Phi} = \{\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \ldots, \phi(\boldsymbol{x}_n)\}$. After that we need to compute the principal components in feature space. The central point in feature space can be calculated as $\boldsymbol{\psi} = \frac{1}{n}\boldsymbol{\Phi}\boldsymbol{1}$. The k-th orthogonal eigenvector $\boldsymbol{\nu}_k$ of the centered covariance matrix could be expressed by linear combinations of training samples, i.e. $\boldsymbol{\nu}_k = \bar{\boldsymbol{\Phi}}\boldsymbol{\gamma}_i$, where $\bar{\boldsymbol{\Phi}}$ is the centered sample matrix. The kernel trick is induced to avoid dealing with the high dimensional or even infinite dimensional data. By applying SVD on the centered kernel matrix $\bar{\boldsymbol{K}}$, $\boldsymbol{\gamma}_i$ can be calculated as $\boldsymbol{\gamma}_i = \frac{\boldsymbol{u}_i}{\sqrt{\lambda_i}}$, where $\boldsymbol{u}_i$ and λ_i are the i-th eigenvector and eigenvalue of $\bar{\boldsymbol{K}}$.

When a new test sample $\boldsymbol{x_t}$ comes, after implicitly being mapped to feature space by computing $\boldsymbol{K_t} = [\phi(\boldsymbol{x_t})^T\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x_t})^T\phi(\boldsymbol{x}_n)]^T$, its projection on to the first l eigenvectors could be expressed as linear combinations of training samples, i.e. $\boldsymbol{P}\phi(\boldsymbol{x_t}) = \boldsymbol{\Phi}\boldsymbol{\beta}$, in which $\boldsymbol{\beta} = \boldsymbol{HMH}(\boldsymbol{K_t} - \frac{1}{n}\boldsymbol{K1}) + \frac{1}{n}\boldsymbol{1}$ and $\boldsymbol{M} = \sum_{i=1}^{l}\boldsymbol{\gamma}_i\boldsymbol{\gamma}_i^T$.

Throughout the algorithm of KPCA, all the computations can be carried out with the "kernel trick". Another observation is that the eigenvectors and the central point can all be represented by linear combinations of $\phi(\boldsymbol{x}_i)$. This conclusion is important and it is utilized to derive our algorithm for robust KPCA.

2.2 Previous robust KPCA algorithms

The classical KPCA can be reformulated as the following minimization problem [10]:

$$\arg\min_{\boldsymbol{\psi},\boldsymbol{V}} = \sum_{i=1}^{n}(\|\phi(\boldsymbol{x}_i) - \boldsymbol{\psi}\|^2 - \|\boldsymbol{V}^T(\phi(\boldsymbol{x}_i) - \boldsymbol{\psi})\|^2) \text{ subject to } \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I} \tag{1}$$

It is easily seen that this optimization problem is not robust since the L_2 loss function is included [5]. Thus many improved algorithms have been proposed based on the knowledge from robust statistics. One of the most representative study was presented in [10], where a monotonic increasing concave function was introduced, acting on each term of the summation in expression (1). The solution process was iterative and small weights would be assigned to outliers in the iteration process. Thus it led to a robust solution. However, in this algorithm a finite basis set was needed to approximate the data in feature space. This resulted in a form of solution different from one obtained from the classical KPCA algorithm, thus can not be merged into the existing systems conveniently.

An outlier detection algorithm for KPCA was proposed in [11]. A sphere with a determined radius was calculated, and samples outside the sphere were determined to be outliers and discarded. However, the premise that outliers and normal samples could be separated by a sphere was unproved. In [12] the influence index of each sample to the eigenvectors were estimated, and the samples with the largest influence indexes were identified as outliers, then they were removed from the dataset. A similar idea to [10] was proposed in [5], where a robust loss function was applied to the dual objective function of the original objective function in KPCA algorithm. However, the solving process was complex, and only one eigenvector could be obtained at a time.

3 The Proposed Robust KPCA Algorithm

Define $y_i = \|\phi(\boldsymbol{x}_i) - \boldsymbol{\psi}\|^2 - \|\boldsymbol{V}^T(\phi(\boldsymbol{x}_i) - \boldsymbol{\psi})\|^2$. Similar to [10], by introducing a monotonic increasing concave function f, we consider the following constrained minimization problem:

$$\arg\min_{\boldsymbol{\psi},\boldsymbol{V}} = \sum_{i=1}^{n} f(y_i) \text{ subject to } \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I} \tag{2}$$

In our robust KPCA algorithm, we follow the observation in classical KPCA that the solution $(\boldsymbol{\psi},\ \boldsymbol{V})$ to problem (2) lies in the subspace spanned by training samples $\phi(\boldsymbol{x}_i)$ in feature space. Define $\boldsymbol{\psi} = \boldsymbol{\Phi}\boldsymbol{\mu},\ \boldsymbol{V} = \boldsymbol{\Phi}\boldsymbol{\Gamma}$. Thus the problem (2) can be rewritten as:

$$\arg\min_{\boldsymbol{\mu},\boldsymbol{\Gamma}} = \sum_{i=1}^{n} f(z_i) \text{ subject to } \boldsymbol{\Gamma}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Gamma} = \boldsymbol{I}, \tag{3}$$

in which $z_i = \|\phi(\boldsymbol{x}_i) - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 - \|\boldsymbol{\Gamma}^T\boldsymbol{\Phi}^T(\phi(\boldsymbol{x}_i) - \boldsymbol{\Phi}\boldsymbol{\mu})\|^2$. Denote the i-th column of kernel matrix $\boldsymbol{K}$ as $\boldsymbol{k}_i$, and the i-th diagonal element as k_{ii}. Thus

$$z_i = k_{ii} + (\boldsymbol{\mu}^T\boldsymbol{K} - 2\boldsymbol{k}_i^T)(\boldsymbol{I} - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T\boldsymbol{K})\boldsymbol{\mu} - \boldsymbol{k}_i^T\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T\boldsymbol{k}_i \tag{4}$$

Theorem 1 *The solution* $(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ *to problem (3) should satisfy the following stationary equation:*

$$\boldsymbol{\mu} = \frac{\boldsymbol{\zeta}_{f'(z_i)}}{\sum_{i=1}^{n} f'(z_i)} \tag{5}$$

$$\boldsymbol{\Gamma} = \mathbf{A}^{-1}\hat{\mathbf{U}}, \tag{6}$$

in which

$$\boldsymbol{\zeta}_{f'(z_i)} = [f'(z_1), f'(z_2), \ldots, f'(z_n)]^T \text{ with } f'(z_i) = \mathrm{d}f(z)/\mathrm{d}z$$

$$\mathbf{A} \text{ is an invertible matrix satisfying } \mathbf{A}^T\mathbf{A} = \mathbf{K}$$

$$\hat{\mathbf{U}} = eigen_l(\mathbf{A}(I - \boldsymbol{\mu}\mathbf{1}^T)\mathbf{G}(I - \mathbf{1}\boldsymbol{\mu}^T)\mathbf{A}^T) \text{ with } \mathbf{G} = diag(f'(z_1), f'(z_2), \ldots, f'(z_n))$$

.

Proof By introducing the Lagrange multipliers, the problem (3) can be reformulated as

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Gamma}; \boldsymbol{\Lambda}) = \sum_{i=1}^{n} f(z_i) + \sum_{i=1}^{l} \lambda_{ii}(\boldsymbol{\gamma}_i^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{\gamma}_i - 1) + \sum_{i=2}^{k}\sum_{j=1}^{i-1} \lambda_{ij} \boldsymbol{\gamma}_i^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{\gamma}_j, \tag{7}$$

where $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_k]$, and $\boldsymbol{\Lambda}$ is a symmetric matrix with $\boldsymbol{\Lambda}_{ii} = \lambda_{ii}$ and $\boldsymbol{\Lambda}_{ij} = \boldsymbol{\Lambda}_{ji} = \lambda_{ij}$.

The Karush-Kuhn-Tucker (KKT) optimality conditions of problem (7) is

$$\begin{cases} \partial\mathcal{L}/\partial\boldsymbol{\mu} = \boldsymbol{0} & \Rightarrow \quad \boldsymbol{\mu} = \frac{\boldsymbol{\zeta}_{f'(z_i)}}{\sum_{i=1}^{n} f'(z_i)} \\ \partial\mathcal{L}/\partial\boldsymbol{\Gamma} = \boldsymbol{0} & \Rightarrow \quad \boldsymbol{K}(I - \boldsymbol{\mu}\boldsymbol{1}^T)\boldsymbol{G}(I - \boldsymbol{1}\boldsymbol{\mu}^T)\boldsymbol{K}\boldsymbol{\Gamma} = \boldsymbol{K}\boldsymbol{\Gamma}\boldsymbol{\Lambda} \\ \partial\mathcal{L}/\partial\boldsymbol{\Lambda} = \boldsymbol{0} & \Rightarrow \quad \boldsymbol{\Gamma}^T\boldsymbol{K}\boldsymbol{\Gamma} = \boldsymbol{I} \end{cases} \tag{8}$$

By introducing a matrix $\boldsymbol{A}$ satisfying that $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{K}$, and define $\hat{\boldsymbol{U}} = \boldsymbol{A}\boldsymbol{\Gamma}$, the above equations can be expressed as:

$$\boldsymbol{A}(I - \boldsymbol{\mu}\boldsymbol{1}^T)\boldsymbol{G}(I - \boldsymbol{1}\boldsymbol{\mu}^T)\boldsymbol{A}^T\hat{\boldsymbol{U}} = \hat{\boldsymbol{U}}\boldsymbol{\Lambda} \text{ with } \hat{\boldsymbol{U}}^T\hat{\boldsymbol{U}} = \boldsymbol{I} \tag{9}$$

Obviously the solution to equation (9) is given by $\hat{\boldsymbol{U}} = eigen_l(\boldsymbol{A}(I - \boldsymbol{\mu}\boldsymbol{1}^T)\boldsymbol{G}(I - \boldsymbol{1}\boldsymbol{\mu}^T)\boldsymbol{A}^T)$. Thus $\boldsymbol{\Gamma}$ can be calculated as $\boldsymbol{\Gamma} = \boldsymbol{A}^{-1}\hat{\boldsymbol{U}}$. Also note that $\boldsymbol{A}$ could be obtained by SVD of $\boldsymbol{K}$. Since $\boldsymbol{K}$ is symmetric, SVD of $\boldsymbol{K}$ is $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$. Thus $A = \sqrt{\boldsymbol{S}}\boldsymbol{U}^T$.

The algorithm of proposed Robust KPCA is summarized as follows.

Input Training set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, a kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j) : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, a monotonic increasing function f

Output $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times l}$

Step 1 Compute the kernel matrix $\boldsymbol{K}$ and the matrix $\boldsymbol{A}$ satisfying $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{K}$.

Step 2 Set the initial values of $\boldsymbol{\mu}$, $\boldsymbol{\Gamma}$ and *iter*, $\boldsymbol{\mu} = \boldsymbol{\mu}^0$, $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^0$, $iter = 0$.

Step 3 $iter = iter + 1$.

Step 4 Compute z_i^{iter} for $i = 1 : n$ using equation (4).

Step 5 Compute $\boldsymbol{\mu}^{iter}$ using equation (5).

Step 6 Compute $\hat{\boldsymbol{U}}^{jter}$ using equation (9).

Step 7 *if convergence* == TRUE

$$\boldsymbol{\mu} = \boldsymbol{\mu}^{iter},\ \hat{\boldsymbol{U}} = \hat{\boldsymbol{U}}^{jter}.$$

else

Goto **Step 3**.

Step 8 Compute $\boldsymbol{\Gamma}$ using equation (6).

In our algorithm, convergence is TRUE while the difference of the values of the cost function in expression (3) between two adjacent iteration steps are smaller than ϵ_L, i.e. *convergence* == TRUE if and only if $|\sum_{i=1}^{n} f(z_i^{iter}) - \sum_{i=1}^{n} f(z_i^{iter-1})| \leqslant \epsilon_L$. Note that the initial value of $\boldsymbol{\mu}$ and $\boldsymbol{\Gamma}$ can be set to random values.

4 Experiments and Discussion

Two experiments were carried out in this section. The toy example showed the robustness of the proposed algorithm, and the actual performance was measured in a practical application. The Gaussian kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\tau}$ was used in the experiments, where $\tau = \frac{1}{n^2}\sum_{i,j=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$. The German-McClure function $f(z) = \frac{z}{1+z/c}$, which had been widely used in robust statistics [13], was used as the robust estimator. The convergence parameter $\epsilon_L = 1 \times 10^{-6}$.

4.1 A Numerical Example

In this subsection, a two-dimensional synthetic dataset $\mathcal{X} : \{(x_i, y_i)\}_{i=1}^{n}$ with $n = 101$ was used. x_i was generated uniformly over $[0, 1]$, and $y_i = 0.5\sin(2\pi x_i) + 0.5 + 0.1r_i$, where r_i was the noise with standard normal distribution. Outliers were generated from normal distribution $N \sim (5, 1)$. The corrupted dataset, which was denoted as $\mathcal{XO}$, was generated by replacing the points in the dataset $\mathcal{X}$ by outliers randomly. The left panel of Fig. 1 gives the scatter diagram of the synthetic dataset with 30 outliers. The right panel of Fig. 1 shows the value of $f'(z_i)$ in our proposed robust KPCA. Note that actually $f'(z_i)$ is the weight of the point in our algorithm. It is clearly shown that smaller weights are assigned to outliers, thus the influence of outliers are suppressed.

To measure the robustness of the proposed method, the true central points and eigenvectors were calculated from dataset $\mathcal{X}$ using the classical KPCA, and they were used as the baseline. Then the classical KPCA and the proposed robust KPCA were applied to the corrupted dataset $\mathcal{XO}$ respectively. The same kernel parameter τ was used in these experiments. The parameter c in the German-McClure function was set to 0.2. For central points, the Euclidean distance was calculated for comparison, and for principal components, the absolute cosine similarity was used, shown as follows:

$$distance(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2) = \|\boldsymbol{\psi}_1 - \boldsymbol{\psi}_2\|^2 = \mu_1^T \boldsymbol{\Phi}_1^T \boldsymbol{\Phi}_1 \mu_1 + \mu_2^T \boldsymbol{\Phi}_2^T \boldsymbol{\Phi}_2 \mu_2 - 2\mu_1^T \boldsymbol{\Phi}_1^T \boldsymbol{\Phi}_2 \mu_2$$

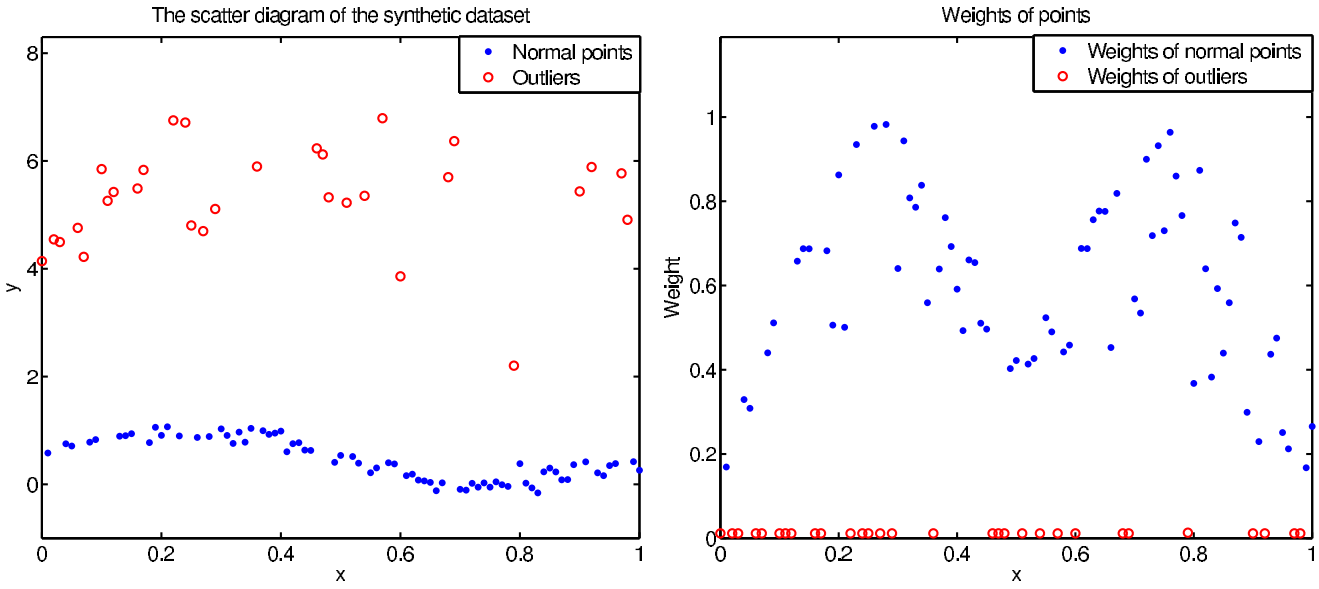

Fig. 1: The synthetic dataset and the weights of the points

$$similarity(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2) = |\boldsymbol{\nu}_1^T \boldsymbol{\nu}_2| = |\boldsymbol{\gamma}_1^T \boldsymbol{\Phi}_1^T \boldsymbol{\Phi}_2 \boldsymbol{\gamma}_2|$$

where $\boldsymbol{\psi} = \boldsymbol{\Phi}\mu$ and $\boldsymbol{\nu} = \boldsymbol{\Phi}\boldsymbol{\gamma}$. The comparison was made using different corrupted datasets containing various numbers of outliers, and only the first principal components were calculated and compared. The results are shown in Fig. 2. When there was only a small number of outliers, the first eigenvectors obtained by the classical KPCA and the proposed robust KPCA were both similar to the true eigenvector. However, with the increase in the number of outliers, the eigenvector obtained by the classical KPCA deviated from the true eigenvector sharply, while the eigenvector obtained by our robust KPCA still remained the similarity. Note that when the number of outliers was greater than 42, the similarity of the true eigenvector and the eigenvector obtained by our robust KPCA reduced rapidly. It was possibly because more than 40% of the points were outliers, such that it was hard to distinguish the normal points and the outliers. The distance metric shows the similar results. The central point obtained from our robust KPCA was close to the true one when the number of outliers was smaller than 42. However, with the increase of the number of outliers, the central point obtained from the classical KPCA gradually moved away from the true central point. The mean iterations was 15. This means the proposed algorithm could converge to the minimum quickly.

4.2 Image denoising

In this subsection the proposed algorithm was applied under a more realistic scenario: image denoising. The extended Yale Face Database B [14, 15] was used, and 200 samples were chosen as the dataset and then they were resized to 32×28. The Gaussian noise $N \sim (0, \sigma^2)$ with $\sigma = 0.1$ was added to the original samples. 20 samples were randomly chosen and corrupted by a 16×16 block consisting of random white and black dots.

The classical KPCA and the proposed robust KPCA were applied for denoising respectively. The pre-image was calculated using the algorithm proposed in [4]. Fig. 3 shows the mean MSE (mean square error) using the two methods with different numbers of eigenvectors. Using the classical KPCA, the minimum value of the mean MSE (4.73) was acquired when the number of

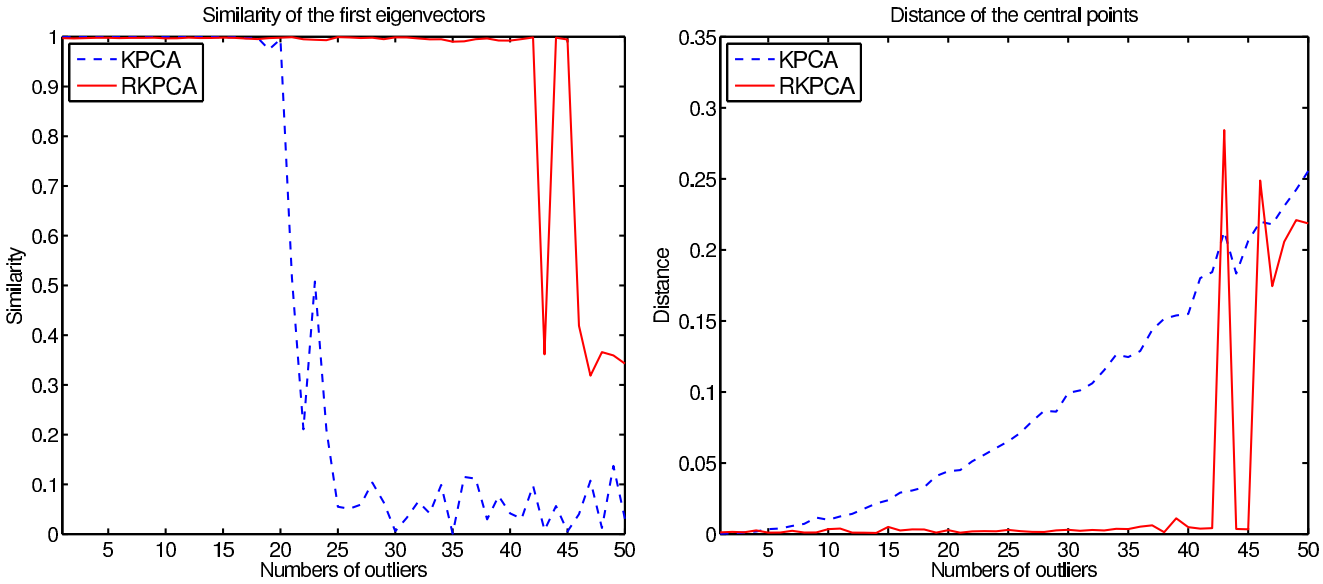

Fig. 2: Comparison of the eigenvectors and the central points using classical KPCA and the proposed robust KPCA

eigenvectors was 15. With the further increase of the number of eigenvectors, the mean MSE also increases. A possible reason is that the subsequent eigenvectors may contain incorrect principal components, which are induced by outliers. On the other hand, the robust KPCA shows a lower mean MSE. The minimum value of the mean MSE (3.35) was acquired when the number of eigenvectors was 35. Note that when the number of eigenvectors is relatively large, the mean MSE obtained using the robust KPCA is relatively large too. Anyhow, the deviation of eigenvectors caused by outliers is substantially reduced, especially for the first few leading eigenvectors.

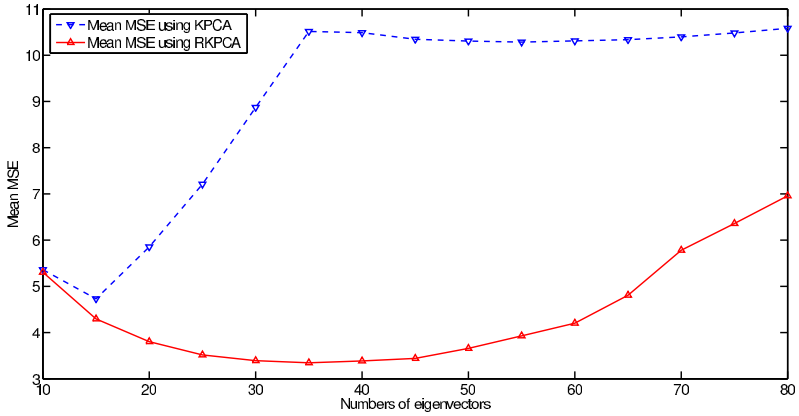

Fig. 3: The mean MSE obtained using the classical KPCA and the proposed robust KPCA

A more intuitive interpretation is shown in Fig. 4, where some clean face images, noisy face images and the denoised face images are shown. The number of eigenvectors were set to their optimal values respectively. The first row shows the original clean images; the second row shows the noisy images; the third row shows the denoised images using the classical KPCA; the last row shows the denoised images using our robust KPCA. It can be clearly seen that the proposed robust KPCA achieves better performance. The blocks with random white and black dots can be removed using the proposed algorithm, while the classical KPCA can not accomplish this goal

very well.

Fig. 4: Visualization of the denoising effect

5 Conclusions

A robust KPCA algorithm was proposed in this paper. Borrowing the idea from robust statistics, a robust loss function was introduced in the original objective function, and we followed the observation that the central point and eigenvectors could be expressed by linear combination of training samples. Thus an improved objective function could be obtained, and we could get an iterative solution, which was of the same form as the classical KPCA. The iteration rule was obtained by calculating the KKT optimality conditions of the improved objective function. And experimental results show that the proposed algorithm can converge relatively fast. The robustness of the proposed algorithm was verified and proved with experiments on a numerical example and a face dataset. In the future, we will apply the proposed algorithm to other kernel applications, like feature extraction, change detection, classification and so on.

References

[1] B. Schölkopf, A. Smola, K. R. Müller, Kernel principal component analysis, in: International Conference on Artificial Neural Networks, pp. 583-588, 1997.

[2] K. I. Kim, K. Jung, H, J, Kim, Face recognition using kernel principal component analysis, IEEE Signal Processing Letters, pp. 40-42, 2002.

[3] J. Tian, Q. Hu, X. Ma, M. Han, An improved KPCA/GA-SVM classification model for plant leaf disease recognition, Journal of Computational Information Systems, pp. 7737-7745. 2012.

[4] J. T. Y. Kwok, I. W. H. Tsang, The pre-image problem in kernel methods, IEEE Transactions on Neural Networks, pp. 1517-1525, 2004.

[5] C. Alzate, J. A. K. Suykens, Kernel component analysis using an epsilon-insensitive robust loss function, IEEE Transactions on Neural Networks, pp. 1583-1598, 2008.

[6] S. K. Yang, J. Y. Meng, H. B. Shen, Preservation of local linearity by neighborhood subspace scaling for solving the pre-image problem, Journal of Zhejiang University SCIENCE C, pp. 254-264, 2014.

[7] Q. S. Liu, J. Cheng, H. Q. Lu, S. D. Ma, Distance based kernel PCA image reconstruction, in: Proc. the 17th International Conference on Pattern Recognition, pp. 670-673, 2004.

[8] Y. Rathi, S. Dambreville, A. Tannenbaum, Statistical shape analysis using kernel PCA, in: Proc. SPIE 6064, pp. 60641B-60641B-8, 2006.

[9] H. Hoffmann, Kernel PCA for novelty detection, Pattern Recognition, pp. 863-874, 2007.

[10] S. Y. Huang, Y. R. Yeh, S. Eguchi, Robust kernel principal component analysis, Neural computation, pp. 3179-3213, 2009.

[11] C. D. Lu, T. Y. Zhang, X. Z. Du, C. P. Li, A robust kernel PCA algorithm, in: Proc. 2004 International Conference on Machine Learning and Cybernetics, pp. 3084-3087, 2004.

[12] M. T. Ding, Z. Tian, Z. Jin, M. Xu, C. X. Cao, Registration using robust kernel principal component for object-based change detection, IEEE Geoscience and Remote Sensing Letters, pp. 761-765, 2010.

[13] S. Baker, I. Matthews, Lucas-kanade 20 years on: A unifying framework, International Journal of Computer Vision, pp. 221-255, 2004.

[14] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 643-660, 2001.

[15] K. C. Lee, J. Ho, D. J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 684-698, 2005.