A Note on Robust Kernel Principal Component Analysis

Xinwei Deng, Ming Yuan, and Agus Sudjianto

Abstract. Extending the classical principal component analysis (PCA), the kernel PCA (Schölkopf, Smola and Müller, 1998) effectively extracts nonlinear structures of high dimensional data. But similar to PCA, the kernel PCA can be sensitive to outliers. Various approaches have been proposed in the literature to robustify the classical PCA. However, it is not immediately clear how these approaches can be "kernelized" in practice. In this paper, we propose a robust kernel PCA procedure. We show that the proposed method can be easily computed. Simulations and a real example in the financial service also demonstrate the competitive performance of our approach when there are outlying observations.

1. Introduction

Principal component analysis (PCA) is a linear transformation that seeks a coordinate system for a set of multivariate observations such that the greatest variance by any projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on. The new coordinates are referred to as the principal components. By keeping only the first few principal components, PCA achieves dimension reduction while retaining characteristics of the dataset that contribute most to its variation (Jolliffe, 1986).

PCA extracts linear features of high dimensional data. In many applications, however, this can be restrictive and it may be more appropriate to consider nonlinear structures of the data. In recent years, several nonlinear extensions of PCA have been proposed in the literature (Oja, 1982; Hastie and Stuetzle, 1989; Oja, 1991;

Address for correspondence: Ming Yuan, School of Industrial and Systems Engineering, Georgia Institute of Technology, 755 Ferst Drive NW, Atlanta, GA 30332-0205. Email: myuan@isye.gatech.edu. The authors wish to thank Professor Jeff Wu for the encouragement and inspiring discussions.

Bregler and Omohundro, 1994; Schölkopf et al., 1998). In particular, Schölkopf et al. (1998) introduced the kernel PCA. To allow nonlinear features, the kernel PCA performs the classical PCA in a feature space that are nonlinear transformations of the original input variables. Clearly this notion only has conceptual value because the feature space can be of infinite dimension to allow flexible nonlinear features. Nevertheless, Schölkopf et al. (1998) showed that the computation of the kernel PCA only involves the inner product in the feature space. Since the inner product in the feature space can be evaluated through a kernel operator, the kernel PCA can be computed efficiently thanks to the so-called "kernel trick" (Schölkopf and Smola, 2002). The kernel PCA has seen the explosion of its popularity since its introduction and has proven to be highly successful in various applications such as image analysis, gene expression data analysis among many others.

It is widely recognized that PCA and the kernel PCA can be extremely sensitive to outlying observations, and conclusions drawn based on contaminated principal components can be misleading. Several ways of robustifying the classical PCA have been proposed in the literature (Jackson, 1991). Among many others, these approaches include employing robust estimate of the covariance matrix (Croux and Haesbroeck, 2000) or measure of variation that is more robust than the variance (Ibazizen and Dauxois, 2003). Despite their success in the case of PCA, it is not immediately clear how these approaches can be extended to the kernel PCA.

To fill in this void, we propose a robust kernel PCA in this paper. Similar to the case of PCA, we use the mean absolute deviation (MAD) to measure the variation by a projection of the data, which is known to be more robust than the variance. We consider applying this robust PCA in the feature space. At the first glance, such a procedure can not be "kernelized" since operations other than inner product are involved in computing MAD. To overcome this problem, we re-formulate our robust kernel PCA using only the inner product in the feature space thanks to the duality property of matrix norms. We also introduce a natural measure to examine the robustness of the original kernel PCA and the proposed robust kernel PCA. We show that this robustness measure can be evaluated using the kernel operator and therefore readily computable for both methods. We use this new measure of influence to show that the robust kernel PCA is much less sensitive to the outlying observations than the original kernel PCA.

The rest of paper is organized as follows. The methodology of the robust kernel PCA is introduced in the next section. In Section 3, we compare the original kernel PCA and the robust kernel PCA based on a perturbation analysis and show that an outlying observation may have arbitrarily large influence on the original kernel PCA whereas its influence on the robust kernel PCA is always bounded by a constant smaller than one. Section 4 presents a simulation study to demonstrate the competitive performance of the robust kernel PCA. To further illustrate the method, we analyze a real data in financial service area using the proposed method in Section 5. We conclude with some discussions in Section 6.

2. Robust Kernel PCA

Given a set of centered observations $\mathbf{x}_k = (x_{k1}, \ldots, x_{kp})'$, $k = 1, \ldots, n$, PCA seeks directions that maximize the variance of the projection of the data. For example, the first principal component is given by

$$\arg\max_{\beta} \sum_{k=1}^{n} (\mathbf{x}_k' \beta)^2 \tag{2.1}$$

It is well known that the variance is extremely sensitive to outliers. To robustify PCA, one can use a more robust measure of variation. In this paper, we consider using MAD and define our first principal component as

$$\arg\max_{\|\beta\|_2=1} \sum_{k=1}^{n} |\mathbf{x}_k' \beta| \tag{2.2}$$

To consider nonlinear features of $\mathbf{x}$ that come from a functional space $\mathcal{F}$, we can apply this robust procedure to the basis functions of $\mathcal{F}$, $\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \ldots$. Without loss of generality, assume that $\sum_k \psi_i(\mathbf{x}_k) = 0$ for any i. We look for a vector β of the same dimension as the basis functions such that

$$\beta = \arg\max_{\|\beta\|_2=1} \sum_{k=1}^{n} |\Psi(\mathbf{x}_k)' \beta| \tag{2.3}$$

where $\Psi(\mathbf{x}) = (\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \ldots)'$.

The functional space $\mathcal{F}$ is often taken to be a reproducing kernel Hilbert space (Wahba, 1990). In such situations, (2.3) may not be computable since $\mathcal{F}$ can have infinite dimension in a genuine nonparametric setup. A powerful technique to get around this problem is by the so-called "kernel trick" (Schölkopf and Smola, 2002). Although there are infinitely many basis functions, the inner product in the feature space can always be computed through a kernel operator. The key step therefore is to express the objective in a formulation using only inner products, which is clearly not the case for (2.3).

To accomplish this goal, we note the duality between the matrix ℓ_p norm and ℓ_q norm given that $1/p + 1/q = 1$. Simple derivation leads to the following matrix transposition invariant property (Choulakian, 2005). Let A be a $m \times n$ matrix, define

$$\|A\|_{pr} = \max_{\|x\|_r = 1 : \mathbf{x} \in R^n} \|A\mathbf{x}\|_p, \tag{2.4}$$

where $\| \cdot \|_p$ is a vector p-norm and $p,\ r > 0$. The transposition invariant property states that

$$\|A\|_{pr} = \|A'\|_{sq}, \tag{2.5}$$

where

$$\frac{1}{p} + \frac{1}{q} = 1, \qquad \frac{1}{r} + \frac{1}{s} = 1. \tag{2.6}$$

Now define $A_{ij} = \psi_j(\mathbf{x}_i)$. Then an application of (2.5) implies that

$$\max_{\|\beta\|_2=1} \sum_{k=1}^{n} |\Psi(\mathbf{x}_k)'\beta| = \|A\|_{12} = \|A'\|_{2\infty} = \max_{\|\alpha\|_\infty = 1} \sqrt{\alpha' A A' \alpha} \tag{2.7}$$

Note that the (i, j) entry of AA' is $\langle \Psi(\mathbf{x}_i), \Psi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$. To evaluate the right hand side of (2.7), it is sufficient to know the kernel operator $K(\cdot, \cdot)$. This kernel representation allows us to compute the value of the inner product in $\mathcal{F}$ without having to carry out the map Ψ. This method was previously used by Boser et al. (1992) to extend the Generalized Portrait hyperplane classifier of Vapnik and Chervonenkis (1974) to nonlinear support vector machines by substituting a prespecified kernel function $K(\cdot, \cdot)$ for all occurrences of inner products. The readers are referred to Schölkopf and Smola (2002) for a detailed account of this so-called "kernel trick". It is also known that there is a one-to-one correspondence between a reproducing kernel Hilbert space and a positive definite kernel operator $K(\cdot, \cdot)$. For this reason, it is often times convenient to directly specify the kernel operator instead of the functional space itself. Kernels that are commonly used in practice include the polynomial kernels and Gaussian kernels.

The polynomial kernel of degree d is given by

$$K(\mathbf{x}, \mathbf{y}) = \left(\langle \mathbf{x}, \mathbf{y} \rangle + 1\right)^d. \tag{2.8}$$

Besides the polynomial kernel, Gaussian kernel

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{\sigma^2}\right) \tag{2.9}$$

is also very popular.

With slight abuse of notation, denote K a $n \times n$ matrix whose (i,j) entry is $K(\mathbf{x}_i, \mathbf{x}_j)$. Then we can rewrite the right hand side of (2.7) as

$$\widehat{\alpha}^{(1)} = \arg \max_{\|\alpha\|_\infty = 1 : \alpha \in R^n} \sqrt{\alpha' K \alpha} = \arg \max_{\|\alpha\|_\infty = 1 : \alpha \in R^n} \alpha' K \alpha, \tag{2.10}$$

where the superscript is used to indicate that it corresponds to the first principal component.

Once $\alpha^{(1)}$ is obtained, again by the transposition invariant property, the maximizer of the left hand side of (2.7) is given by $\widehat{\beta}^{(1)} = A'\alpha^{(1)}/\sqrt{(\alpha^{(1)})' K \alpha^{(1)}}$, which again requires the knowledge of map Ψ. Fortunately, we are only interested in the projection of a data point $\mathbf{x}$ into the principal components, which can be computed as

$$\Psi(\mathbf{x})'\beta^{(1)} = \frac{\Psi(\mathbf{x})' A' \alpha^{(1)}}{\sqrt{(\alpha^{(1)})' K \alpha^{(1)}}} = \frac{\sum_{k=1}^n \alpha_k^{(1)} \langle \Psi(\mathbf{x}), \Psi(\mathbf{x}_k) \rangle}{\sqrt{(\alpha^{(1)})' K \alpha^{(1)}}} = \frac{\sum_{k=1}^n \alpha_k^{(1)} K(\mathbf{x}, \mathbf{x}_k)}{\sqrt{(\alpha^{(1)})' K \alpha^{(1)}}}. \tag{2.11}$$

After the first principal component is obtained, we then target at the second principal component which is orthogonal to the first one. We first project the data from the feature space $\mathcal{F}$ into its linear subspace that is orthogonal to the first principal component. Note that the second principal component is now the first principal component of the projected data. The aforementioned procedure for the first principal component can then be applied if we know how to compute the kernel operator in the linear subspace. Let $\Psi(\mathbf{x})$ be a point in $\mathcal{F}$, then $\Psi(\mathbf{x}) - \widehat{\beta}^{(1)}(\widehat{\beta}^{(1)})'\Psi(\mathbf{x})$ is its projection into the linear subspace that is orthogonal to $\widehat{\beta}^{(1)}$. The inner product of the linear subspace can be calculated:

$$\begin{aligned} K^{(2)}(\mathbf{x}, \mathbf{y}) &= \langle \Psi(\mathbf{x}) - \widehat{\beta}^{(1)}(\widehat{\beta}^{(1)})'\Psi(\mathbf{x}), \Psi(\mathbf{y}) - \widehat{\beta}^{(1)}(\widehat{\beta}^{(1)})'\Psi(\mathbf{y}) \rangle \\ &= \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle - 2\langle \Psi(\mathbf{x}), \widehat{\beta}^{(1)}(\widehat{\beta}^{(1)})'\Psi(\mathbf{y}) \rangle \\ &\quad + \langle \widehat{\beta}^{(1)}(\widehat{\beta}^{(1)})'\Psi(\mathbf{x}), \widehat{\beta}^{(1)}(\widehat{\beta}^{(1)})'\Psi(\mathbf{y}) \rangle \\ &= \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle - \langle \Psi(\mathbf{x}), \widehat{\beta}^{(1)}(\widehat{\beta}^{(1)})'\Psi(\mathbf{y}) \rangle \qquad (2.12) \end{aligned}$$

Note that $\widehat{\beta}^{(1)} = A'\alpha^{(1)}/\sqrt{(\alpha^{(1)})' K \alpha^{(1)}}$

$$\langle \Psi(\mathbf{x}), \widehat{\beta}^{(1)}(\widehat{\beta}^{(1)})'\Psi(\mathbf{y}) \rangle = \frac{\Psi(\mathbf{x})' A' \alpha^{(1)} (\alpha^{(1)})' A \Psi(\mathbf{y})}{(\alpha^{(1)})' K \alpha^{(1)}} = \frac{\sum_{i,j=1}^n \alpha_i^{(1)} \alpha_j^{(1)} K(\mathbf{x}, \mathbf{x}_i) K(\mathbf{x}_j, \mathbf{y})}{(\alpha^{(1)})' K \alpha^{(1)}}. \tag{2.13}$$

Therefore,

$$K^{(2)}(\mathbf{x}, \mathbf{y}) = K(\mathbf{x}, \mathbf{y}) - \frac{\sum_{i,j=1}^n \alpha_i^{(1)} \alpha_j^{(1)} K(\mathbf{x}, \mathbf{x}_i) K(\mathbf{x}_j, \mathbf{y})}{(\alpha^{(1)})' K \alpha^{(1)}} \tag{2.14}$$

which can be computed without knowing Ψ.

The rest of the principle components can be computed in a similar fashion. In general, the kernel operator needed for the rth principal component is

$$K^{(r)}(\mathbf{x},\mathbf{y}) = K(\mathbf{x},\mathbf{y}) - \langle \Psi(\mathbf{x}), W\Psi(\mathbf{y})\rangle \tag{2.15}$$

where $W = (\beta^{(1)}, \ldots, \beta^{(r-1)})(\beta^{(1)}, \ldots, \beta^{(r-1)})'$.

To sum up, our proposed robust kernel PCA method can be computed using the following recipe:

Algorithm 1 Compute First R Robust Kernel Principal Components

```
Step 1. Compute $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ for all $i, j = 1, \ldots, n$.
Step 2. Center the kernel matrix $\bar{K} = K - \mathbf{1}K/n - K\mathbf{1}/n + \mathbf{1}K\mathbf{1}/n^2$, where $\mathbf{1}$ is
    a $n \times n$ matrix with ones.
Step 3. Compute the first principal component through $\alpha$ using (2.10) and kernel
    $\bar{K}$.
Step 4. For $r = 2$ to $R$
    (a) Compute the kernel matrix $(K^{(r)}(\mathbf{x}_i, \mathbf{x}_j))$ using (2.15)
    (b) Center the kernel matrix as in Step 2.
    (c) Compute the $r$th principal component using (2.10) with the kernel matrix obtained
```

3. Perturbation Analysis

The influence function is a commonly used measure of the robustness for a statistical procedure. The influence function of a statistical functional $T_0(F)$ is defined as

$$IC_{T_0,F}(z) = \lim_{\epsilon \to 0} \frac{T_0(F_\epsilon) - T_0(F)}{\epsilon} = \left.\frac{\partial T_0(F_\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} \tag{3.1}$$

where F is a distribution function, $F_\epsilon = (1-\epsilon)F + \epsilon\delta_z$ and δ_z is a point mass at z. Of particular interest is the choice of $z = \mathbf{x}_i$, $\epsilon = 1/(n-1)$ and F being the empirical distribution function, which amounts to measuring the influence of deleting the ith case (Cook and Weisberg, 1982). Instead of deleting cases one at a time, some authors have suggested to perturb a single case and the influence of the corresponding case is investigated through the deriviative of the perturbation. To formalize this approach, assign each case a weight w_i $(i = 1, \ldots, n)$. Denote $T_{\mathbf{w}}$

the statistic with weights $\mathbf{w} = (w_1, \ldots, w_n)'$. The influence of the ith case is given as

$$\left.\frac{\partial T_{\mathbf{w}}}{\partial w_i}\right|_{\mathbf{w}=(1,\ldots,1)'} \tag{3.2}$$

In the case of the kernel PCA, let β be a principal component in $\mathcal{F}$. The influence of the ith observation on the projection of a future data point $\mathbf{x}_0$ is

$$\Psi(\mathbf{x}_0)' \left.\frac{\partial \beta}{\partial w_i}\right|_{\mathbf{w}=(1,\ldots,1)'} . \tag{3.3}$$

It is therefore natural to measure the robustness of β using

$$\mathrm{IF}_i(\beta) = \left|\left| \left.\frac{\partial \beta}{\partial w_i}\right|_{\mathbf{w}=(1,\ldots,1)'} \right|\right|_2^2 \tag{3.4}$$

To fix ideas, we consider only the first principle component $\widehat{\beta}^{(1)}$ in the following discussion. We begin with the original kernel PCA of Schölkopf et al. (1998). Note that $\widehat{\beta}^{(1)}$ is the linear principal component in $\mathcal{F}$, Critchley (1985) has shown that

$$\mathrm{IF}_i(\widehat{\beta}^{(1)}) = \left(\frac{2}{n}\right)^2 \left((\Psi(\mathbf{x}_i)'\widehat{\beta}^{(1)}\right)^2 \sum_{r>1} \frac{\left(\Psi(\mathbf{x}_i)'\widehat{\beta}^{(r)}\right)^2}{(\widehat{\lambda}_1 - \widehat{\lambda}_r)^2}, \tag{3.5}$$

where $\widehat{\lambda}_r'$s are the eigenvalues corresponding to $\widehat{\alpha}^{(r)}$. Clearly, the influence function is unbounded for certain outlying observations.

To evaluate this influence function, we need to compute $\Psi(\mathbf{x}_i)'\widehat{\beta}^{(r)}$, the projection of the ith observation on the rth principal component. To this end, we apply the transposition invariant property to (2.1),

$$\max_{\beta} \sum_{k=1}^{n} (\mathbf{x}_k'\beta)^2 = \max_{\alpha} \alpha' K \alpha. \tag{3.6}$$

Therefore, $\widehat{\beta}^{(1)} = A'\widehat{\alpha}^{(1)}/\sqrt{(\widehat{\alpha}^{(1)})'K\widehat{\alpha}^{(1)}}$ where $\widehat{\alpha}^{(1)}$ is the first principal component of K. Now the influence function can be re-written in terms of $\widehat{\alpha}^{(1)}$:

$$\mathrm{IF}_i(\widehat{\beta}^{(1)}) = \left(\frac{2}{n}\right)^2 \frac{\left(K_i\widehat{\alpha}^{(1)}\right)^2}{(\widehat{\alpha}^{(1)})'K\widehat{\alpha}^{(1)}} \sum_{r>1} \frac{\left(K_i\widehat{\alpha}^{(r)}\right)^2}{(\widehat{\alpha}^{(r)})'K\widehat{\alpha}^{(r)}(\widehat{\lambda}_1 - \widehat{\lambda}_r)^2} \tag{3.7}$$

where K_i is the ith row of K. Note that (3.7) can be computed without knowing the map Ψ.

In the case of robust kernel PCA, after introducing the weights $w_1, \ldots, w_n$, we can rewrite (2.10) as

$$\widehat{\alpha}^{(1)*} = \arg \max_{\|\alpha\|_\infty = 1 : \alpha \in R^n} \alpha' \Omega K \Omega \alpha, \tag{3.8}$$

where Ω is a diagonal matrix whose (i,i) entry is w_i. Because of the discrete nature of the feasible set, $\widehat{\alpha}^{(1)*} = \widehat{\alpha}^{(1)}$ given that w_i's are sufficiently close to 1. Therefore, after perturbation,

$$\widehat{\beta}^{(1)*} = \frac{A'\Omega\alpha^{(1)*}}{\sqrt{(\alpha^{(1)*})'\Omega K\Omega\alpha^{(1)*}}} = \frac{A'\Omega\alpha^{(1)}}{\sqrt{(\alpha^{(1)})'\Omega K\Omega\alpha^{(1)}}} \tag{3.9}$$

Let $w_i = 1-\epsilon$ and $w_j = 1$ for all $j \neq i$

$$\widehat{\beta}^{(1)*} = \widehat{\beta}^{(1)} - \frac{\epsilon}{\sqrt{(\alpha^{(1)})'K\alpha^{(1)}}}\left(A'H_i\alpha^{(1)} - \frac{(\alpha^{(1)})'KH_i\alpha^{(1)}}{(\alpha^{(1)})'K\alpha^{(1)}}A'\alpha^{(1)}\right) + O(\epsilon^2), \tag{3.10}$$

where H_i is a $n \times n$ matrix with zeros except that its (i,i)th entry is one. It is natural to define the influence of perturbing the ith observation as

$$\begin{aligned}
\text{IF}_i(\widehat{\beta}^{(1)}) &= \left\|\frac{1}{\sqrt{(\alpha^{(1)})'K\alpha^{(1)}}}\left(A'H_i\alpha^{(1)} - \frac{(\alpha^{(1)})'KH_i\alpha^{(1)}}{(\alpha^{(1)})'K\alpha^{(1)}}A'\alpha^{(1)}\right)\right\|_2^2 &(3.11)\\
&= \frac{(\alpha^{(1)})'H_iKH_i\alpha^{(1)}}{(\alpha^{(1)})'K\alpha^{(1)}} - \frac{\left((\alpha^{(1)})'KH_i\alpha^{(1)}\right)^2}{\left((\alpha^{(1)})'K\alpha^{(1)}\right)^2} &(3.12)
\end{aligned}$$

In contrast to the original kernel PCA, the influence function of the robust kernel PCA is bounded by the first term. To be specific, note that $\|H_i\alpha^{(1)}\|_\infty = \alpha_i^{(1)} \leq \|\alpha^{(1)}\|_\infty = 1$. We have

$$\text{IF}_i(\widehat{\beta}^{(1)}) \leq \frac{(\alpha^{(1)})'H_iKH_i\alpha^{(1)}}{(\alpha^{(1)})'K\alpha^{(1)}} < 1, \tag{3.13}$$

from the definition of $\alpha^{(1)}$.

4. Simulation

To illustrate the methodology, we first consider a toy example. We use this example to demonstrate the robustness of the proposed approach. We first randomly generate fifty data points around a circle in the two dimensional space. Each point is generated in the following fashion. First an angle is sampled from a uniform distribution between 0 and 2π. The radius is then randomly generated from $N(3, 0.05^2)$. In the top panels of Figure 1 we plot the data points together with the first original kernel principal component and first robust kernel principal component. We use the polynomial kernel with degree two for both methods. The two methods perform very similarly in this case. Now we add an outlier to the data. The outlying observation is located at $(5,5)$. We plot in the bottom panels of Figure 1 the first original kernel principal component and the first robust kernel principal component of the contaminated data. The result suggests that the influence of the outlier on the original kernel principal component is quite significant, but marginal for the robust kernel principal component.

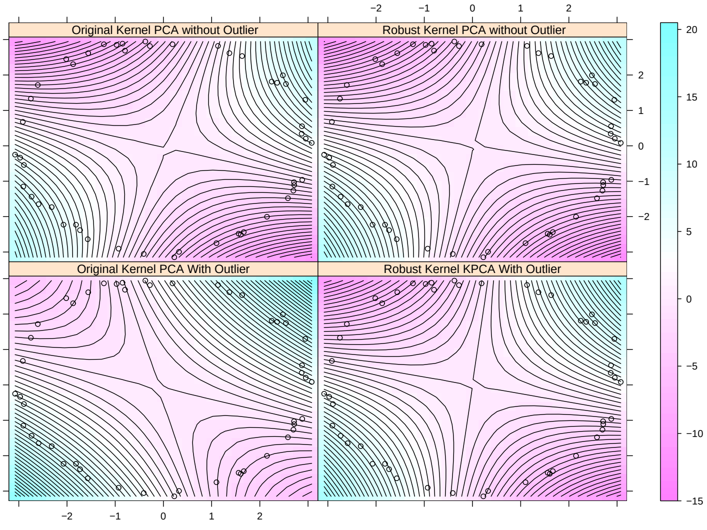


FIGURE 1. First kernel principal component for the two-dimensional circle example

Next, we examine the robustness property from a different angle by looking at the perturbation analysis from Section 3. For each method, we compute the influence of each observation. To make the influence measure comparable in magnitude for the two different methods, we scale the influence values so that the sum of the influence over all observations is one. We compare the normalized influence of the outlier for the two methods. The comparison is based on 1000 datasets simulated in the aforementioned fashion. The pairwise comparison of the influence is given in Figure 2, from which we see a significant reduction of the influence of the outlier for our robust kernel PCA.

5. Real Example

We now apply our method to a real application in financial service. For the purpose of surveillance, it is of great importance to characterize the normal transaction behavior in contrast with the suspicious ones. The banking experts often times look over several important aspects of an account history such as the number

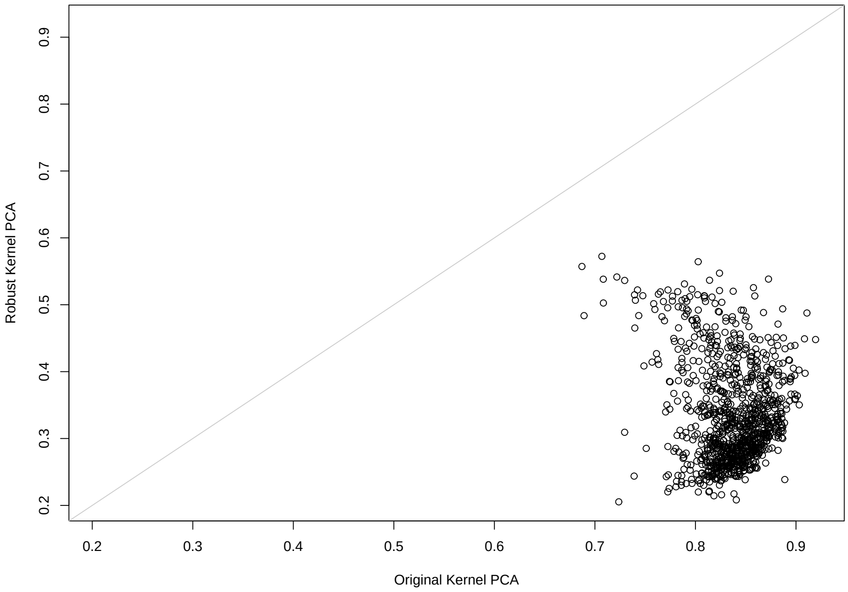
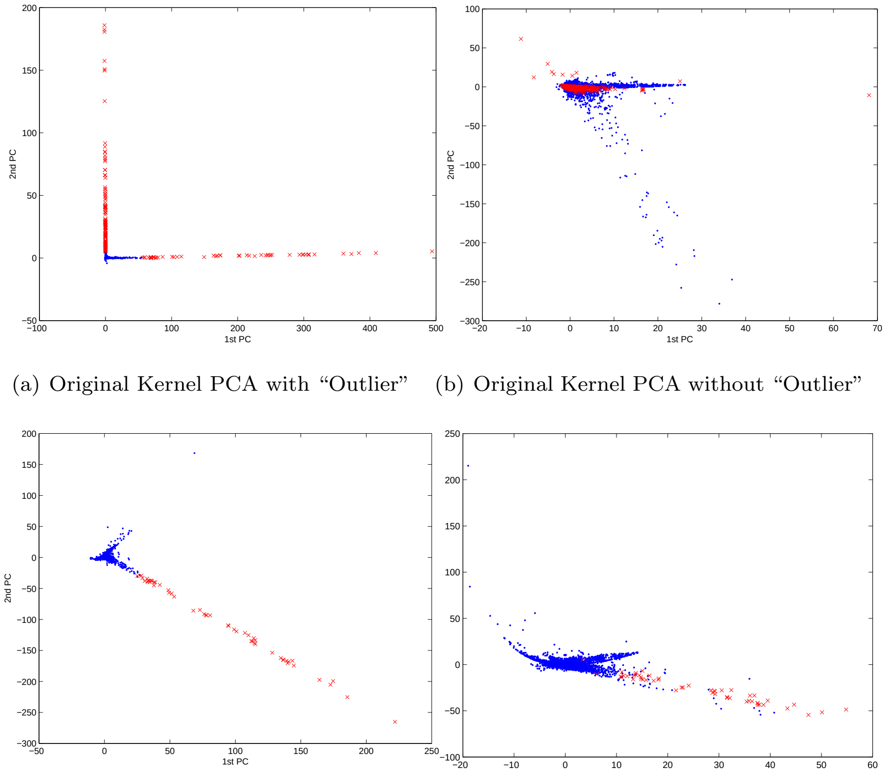


FIGURE 2. Influence measure of the outlier for the two-dimensional circle example

of transactions, the total amount of transactions among others in order to reveal transaction patterns. In a particular example, the experts suspect that there might be suspicious cases among a sample of 6321 accounts. The account history in a one-month period is summarized by eight statistical measurements. For confidentiality reason, we do not disclose more details about the data we are using here. It is clearly very time-consuming for the expert to look over all the cases. Efficient dimension reduction and visualization tool such as the kernel PCA would prove extremely helpful in this aspect. We apply the original kernel PCA and robust kernel PCA on the data to extract the first two kernel principal components and project all cases in a two dimensional space spanned by these components. We immediately see from both plots that there might be outlying cases, or in other words abnormal behavior in this dataset. The question next is which method more accurately characterizes the normal pattern and identifies suspicious activities. To this end, we re-run both kernel PCA methods with the corresponding outlying observations removed and project all cases in the new principal component space. The projections are given in the following figure with the red dots representing the outliers identified by each method. For the original kernel PCA, majority of the outlying observations found in the original analysis do not appear to be abnormal anymore. One plausible explanation is that the original kernel principal components found on the original data were influenced by the truly abnormal cases and the two-dimensional projection fails to capture the real pattern of the normal activities. In contrast the outlying observations found by robust kernel PCA still appear to be abnormal.



(a) Original Kernel PCA with "Outlier" (b) Original Kernel PCA without "Outlier"

(c) Robust Kernel PCA with "Outlier" (d) Robust Kernel PCA without "Outlier"

6. Conclusion

It is known that the kernel PCA may suffer from the presence of outlying observations. Taking advantage of the dual matrix norms, we propose a robust kernel PCA procedure in this paper. We demonstrate by a simulation study and a real application that the proposed method is more robust to outliers than the original kernel PCA.

A more general class of principal direction can be given in the feature space as

$$\arg\max_{\|\beta\|_2=1} \|A'\beta\|_p, \tag{6.1}$$

for some $1 \leq p \leq 2$. The original kernel PCA takes $p = 2$ whereas our robust kernel PCA chooses $p = 1$. Although we have focused on using the mean absolute deviation in this note, it is worth noting that all these kernel PCA can be "kernelized" in the same fashion as our robust kernel PCA. In particular, they are also determined by a n dimensional vector

$$\arg\max_{\|\alpha\|_q=1} \alpha' K\alpha, \tag{6.2}$$

where q is such that $1/p + 1/q = 1$. We choose $p = 1$ because of the robustness it brings about. Other choices may also have their own merits. We leave this for future studies.

References

[1] B. Boser, I. Guyon, and V. Vapnik, *A training algorithm for optimal margin classifiers*, in: Proceedings Fifth ACM Workshop on Computational Learning Theory, 1992, pp. 144–152.

[2] C. Bregler and M. Omohundro, *Surface Learning with Applications to Lipreading*, in J. D. Cowan, G. Tesauro, and J. Alspector (eds.), Advances in Neural Information Precessing Systems 6, Morgan Kaufmann, San Francisco, 1994.

[3] V. Choulakian, *Transposition invariant principal component analysis in L1 for long tail data*, Statistics and Probability Letters, **71** (2005), 23-31.

[4] R. Cook and S. Weisberg, *Residuals and Influence in Regression*, Chapman and Hall, London, 1984.

[5] F. Critchley, *Influence in principal components analysis*, Biometrika, (3) **72** (1985), 627-636.

[6] B. Schölkopf, A. Smola, and K. Müller, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural Computation, (5) **10** (1998), 1299-1319.

[7] C. Croux and G. Haesbroeck, *Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies*, Biometrika, **87** (2000), 603–618.

[8] T. Hastie and W. Stuetzle, *Principal curve*, Journal of The American Statistical Association, **84** (1989), 502-516.

[9] M. Ibazizen and J. Dauxois, *A robust principal component analysis*, Statistics, **37** (2003), 73–83.

[10] J. Jackson, *A Users Guide to Principal Components*, Wiley, New York, 1991.

[11] I. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, 1986.

[12] E. Oja, *A simplified neuron model as a principal component analyzer*, Journal of Mathematical Biology, **15** (1982), 267-273.

[13] E. Oja, H. Ogawa, and J. Wangviwattana, *Learning in nonlinear constrained Hebbian networks*, in: Artificial Neural Networks, Elsevier, Amsterdam, 1991, pp. 385 – 390.

[14] B. Schölkopf and A. Smola, *Learning with Kernels*, MIT Press, Cambridge, 2002.

[15] V. Vapnik and A. Chervonenkis, *Theory of Pattern Recognition*, Nauka, Moscow, 1974.

[16] G. Wahba, *Spline Models for Observational Data*, SIAM, Philadephia, 1990.

School of Industrial and Systems Engineering, Georgia Institute of Technology, 755 Ferst Drive NW, Atlanta, GA 30332-0205.

School of Industrial and Systems Engineering, Georgia Institute of Technology, 755 Ferst Drive NW, Atlanta, GA 30332-0205.

Quantitative Risk Management, Bank of America