Detecting Influential Observations in Kernel PCA

Michiel Debruyne[*,a], Mia Hubert[b], Johan Van Horebeek[c]

[a] *Dept. of mathematics and computer science, Universiteit Antwerpen, Middelheimlaan 1G, B-2020 Antwerpen, Belgium.*
[b] *Dept. of mathematics, K.U.Leuven - LStat, Celestijnenlaan 200B, B-3001 Leuven, Belgium.*
[c] *Center for Research in Mathematics (CIMAT), Apartado Postal 402, Guanajuato, Gto. 36000, México.*

Abstract

Kernel Principal Component Analysis extends linear PCA from a Euclidean space to any reproducing kernel Hilbert space. Robustness issues for Kernel PCA are studied. The sensitivity of Kernel PCA to individual observations is characterized by calculating the influence function. A robust Kernel PCA method is proposed by incorporating kernels in the Spherical PCA algorithm. Using the scores from Spherical Kernel PCA, a graphical diagnostic is proposed to detect points that are influential for ordinary Kernel PCA.

Key words: Robust statistics, Spherical PCA, kernel methods, influence function.

1. Introduction

Principal Component Analysis (PCA) is a well known technique designed to reduce the dimension of a data set by projecting onto a lower dimensional subspace. Kernel PCA (Schölkopf et al., 1998) is an extension of PCA where the data are first mapped into a high dimensional feature space. Then ordinary PCA is performed in this feature space. A remarkable aspect is that the explicit feature vectors are not needed to compute the resulting scores. Only the inner products between feature vectors are required. This makes it possible to apply the kernel trick: one replaces all inner products by a kernel function that is chosen beforehand (see for example Schölkopf and Smola (2002) for an extensive overview of kernel methods). This extension from linear to Kernel PCA has found many applications in recent years. It is for instance easy to consider types of non-linear PCA, simply by defining an appropriate non-linear kernel function. Also when the data consist of objects rather than real numbers, such a kernel formulation is very attractive: it suffices to define an appropriate kernel function between any two such objects. For example in text and string analysis, kernel methods enjoy an increasing popularity (Shawe-Taylor and Cristianini, 2004).

The current paper addresses some questions about influential observations in Kernel PCA. For linear PCA this has been studied intensively. It is known that some observations are relatively less important than others. Points close to the center for example do not really help a

[*]Corresponding author at: Dept. of mathematics and computer science, Universiteit Antwerpen, Middelheimlaan 1G, B-2020 Antwerpen, Belgium. Tel: +32(0)32653887. Fax: +32(0)32653777.
Email address: michiel.debruyne@ua.ac.be (Michiel Debruyne)

lot determining principal components. Observations far away from the center on the other hand are much more influential. Actually, it is even possible that one or a small fraction of observations in the data set almost fully determines the principal components. Sometimes this is not desirable, since then the structure of the majority of the data is not learned anymore. Therefore many robust PCA algorithms have been proposed, for instance by Locantore et al. (1999); Hubert et al. (2002); Croux and Ruiz-Gazen (2005); Hubert et al. (2005); Maronna (2005); Serneels and Verdonck (2008); Chen et al. (2009); Hubert et al. (2009). These methods are less affected by outliers and produce scores which do fit the majority of the data points. Additionally outliers can be detected using these methods in appropriate diagnostic tools.

The goal of this paper is to extend these robustness issues from linear PCA to general Kernel PCA. Our contribution is threefold. Firstly a theoretical analysis of the effect of contamination for ordinary Kernel PCA is provided. To this extent we calculate the influence function (Hampel et al., 1986) of Kernel PCA in Section 3. We show that the influence function can be arbitrary large for unbounded kernels. This means that very small fractions of observations can completely determine the results, such that the structure of the majority of the data is not reflected. When the kernel is bounded however, the influence function is bounded as well.

Since the influence function indicates that a small fraction of contamination can be very influential for a general unbounded kernel, our next step is to construct a robust Kernel PCA algorithm. To this end Spherical Kernel PCA is proposed. It is a generalization of the linear method from Locantore et al. (1999). The first step is a robust centering of the data, which is explained in detail in Section 4.1. The entire Spherical KPCA procedure is given in Section 4.2.

In practice it is often difficult to detect influential observations and to know how to handle them. To perform such detection for ordinary KPCA a visual display is constructed in Section 5 applying an idea from Pison and Van Aelst (2004).

Section 6 illustrates our new methods on some specific examples. It is shown that influential observations in the data can be neutralized and detected using Spherical KPCA, whereas classical KPCA fails to do so.

2. Kernel PCA

Assume that we have a sample of n observations in some non-empty set $\mathcal{X}$: $x_i \in \mathcal{X}$, $i = 1, \ldots, n$. The following definitions are taken from Steinwart and Christmann (2008).

Definition 1. *Let $\mathcal{X}$ be a non-empty set. Then a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a* kernel *on $\mathcal{X}$ if there exists a $\mathbb{R}$-Hilbert space $\mathcal{H}$ with an inner product $\langle \cdot, \cdot \rangle$ and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that for all $x, x' \in \mathcal{X}$ we have*

$$K(x, x') \;=\; \langle \Phi(x), \Phi(x') \rangle \,. \tag{1}$$

We call Φ a feature map and $\mathcal{H}$ a feature space of K.

Definition 2. *Let $\mathcal{X}$ be a non-empty set and $\mathcal{H}$ be a $\mathbb{R}$-Hilbert function space over X, i.e., a $\mathbb{R}$-Hilbert space that consists of functions mapping from $\mathcal{X}$ into $\mathbb{R}$.*

1. *A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{H}$ if we have $K(\cdot, x) \in \mathcal{H}$ for all $x \in \mathcal{X}$ and the reproducing property $f(x) = \langle f, K(\cdot, x) \rangle$ holds for all $f \in \mathcal{H}$ and all $x \in \mathcal{X}$.*

2. *The space $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) over $\mathcal{X}$ if for all $x \in \mathcal{X}$ the Dirac functional $\delta_x : \mathcal{H} \to \mathbb{R}$ defined by*

$$\delta_x(f) := f(x), f \in \mathcal{H}$$

is continuous.

Note that any reproducing kernel is a kernel in the sense of Definition 1. The RKHS is also a feature space of K, with feature map $\Phi : \mathcal{X} \to \mathcal{H}$ given by

$$\Phi(x) = K(\cdot, x). \qquad\qquad x \in \mathcal{X}.$$

We then call Φ the canonical feature map.

Given a specified reproducing kernel function, Kernel PCA basically performs linear PCA in a feature space $\mathcal{H}$ instead of the original space $\mathcal{X}$. Schölkopf et al. (1998) show that the solution of this problem can be obtained only in terms of the inner products between feature vectors. Assume that the feature vectors are mean-centered. Denote Ω the matrix containing $\langle \Phi(x_i), \Phi(x_j) \rangle$ as i, j-th entry. The first principal component equals the direction maximizing the variance of the projections onto this direction. Since a principal component is always contained in the space spanned by the observations, this direction can be written as a linear combination of the feature vectors. Let $\alpha = (\alpha_1, \ldots, \alpha_n)^t \in \mathbb{R}^n$. Then

$$\left\| \sum_{i=1}^{n} \alpha_i \Phi(x_i) \right\|^2 = \left\langle \sum_{i=1}^{n} \alpha_i \Phi(x_i), \sum_{i=1}^{n} \alpha_i \Phi(x_i) \right\rangle = \alpha^t \Omega \alpha.$$

Thus the condition $\alpha^t \Omega \alpha = 1$ ensures that the norm of $\sum_{i=1}^{n} \alpha_i \Phi(x_i)$ equals 1. Moreover the projection of feature vector $\Phi(x_j)$ onto such a direction equals

$$\left\langle \sum_{i=1}^{n} \alpha_i \Phi(x_i), \Phi(x_j) \right\rangle = \sum_{i=1}^{n} \alpha_i \left\langle \Phi(x_i), \Phi(x_j) \right\rangle = (\Omega \alpha)_j.$$

Thus maximizing the variance of these projections is equivalent to

$$\max \alpha^t \Omega^2 \alpha \qquad \text{subject to} \qquad \alpha^t \Omega \alpha = 1. \tag{2}$$

The maximum is obtained for α equal to the eigenvector of Ω corresponding to the largest eigenvalue λ_1, with norm equal to $\lambda_1^{-1/2}$. Denote the unit norm eigenvector corresponding to λ_1 as $\alpha^{(1)}$. Then the score of a new feature vector $\Phi(x)$ corresponds to

$$\left\langle \sum_{i=1}^{n} \frac{\alpha_i^{(1)}}{\sqrt{\lambda_1}} \Phi(x_i), \Phi(x) \right\rangle = \sum_{i=1}^{n} \frac{\alpha_i^{(1)}}{\sqrt{\lambda_1}} \langle \Phi(x_i), \Phi(x) \rangle \tag{3}$$

It is now clear that expressions (2) and (3) depend on the feature vectors $\Phi(x_i)$ only through pairwise inner products. For a Reproducing Kernel Hilbert Space $\mathcal{H}$ these inner products can be evaluated using the underlying kernel function (Definition 1).

Note that the feature vectors were assumed to be centered around zero. However, centering around the mean can be performed explicitly. Taking this into account, Schölkopf et al. (1998) end up with the following result:

Algorithm 1 (Kernel PCA). *Given a sample $x_1, \ldots, x_n \in \mathcal{X}$. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel with corresponding feature space $\mathcal{H}$. Then the kth Kernel PCA (KPCA) score function s_k at $x \in \mathcal{X}$ equals:*

$$s_k(x) = \sum_{i=1}^{n} \frac{\alpha_i^{(k)}}{\sqrt{\lambda_k}} \left(K(x_i, x) - \frac{1}{n} \sum_{l=1}^{n} K(x_l, x) \right)$$

with $\alpha^{(k)}$ the unit norm eigenvector belonging to the kth largest eigenvalue λ_k of the mean centered kernel matrix $\Omega_{c,mean}$ with entry i, j equal to

$$(\Omega_{c,mean})_{i,j} := \left(K(x_i, x_j) - \frac{1}{n} \sum_{k=1}^{n} K(x_k, x_j) - \frac{1}{n} \sum_{k=1}^{n} K(x_k, x_i) + \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l=1}^{n} K(x_k, x_l) \right). \quad (4)$$

Some well known kernels when $\mathcal{X} \subset \mathbb{R}^d$ are the linear kernel

$$K(u, v) = u^t v,$$

the polynomial kernel of degree $p > 0$ with offset $\tau \in \mathbb{R}^+$

$$K(u, v) = (u^t v + \tau)^p,$$

and the Radial Basis Function (RBF) kernel with bandwidth $\sigma \in \mathbb{R}^+$

$$K(u, v) = e^{-\|u-v\|^2/\sigma^2}.$$

Many more types of kernels exist, see for instance Schölkopf and Smola (2002) and Shawe-Taylor and Cristianini (2004).

Note that KPCA with a linear kernel produces exactly the same scores as traditional linear PCA. Although the computation in Algorithm 1 might seem unusual at first sight (using the eigenvectors of the $n \times n$ kernel matrix rather than the maybe more usual decomposition of the $d \times d$ sample covariance matrix), the resulting score function is the same. So KPCA with the linear kernel is just a different formulation of classical linear PCA, of course with the advantage that in the KPCA formulation only the inner products between the data vectors occur. This can then be exploited to extend PCA to an arbitrary reproducing kernel Hilbert space as done in Algorithm 1.

Once the score functions are determined one typically computes a new $n \times m$ data matrix containing the first m scores of the original inputs x_j: $s_k(x_j)$ for $j = 1, \ldots, n$ and $k = 1, \ldots, m$. This can have several purposes. When the original data x_j is high dimensional one can for instance achieve dimension reduction by choosing m not too large, which is usually the goal of linear PCA. For well chosen non linear kernels such as RBF or polynomial an additional goal can be capturing non linearity in the original data, such that a linear method can be used on the new data $s_k(x_j)$, even if the original data x_j exhibits a non linear structure. Several authors for instance use KPCA with a RBF kernel on high dimensional microarray data to construct lower dimensional score vectors. Afterward simple linear discriminant analysis is successfully used to classify the data, although the original data might show a non linear classification pattern (Liu et al., 2005; Pochet et al., 2004; Yang et al., 2005). Also in clustering applications such strategies are reported succesfully, especially with a RBF kernel (e.g. Liu et al. (2005)). Another application of KPCA is in the area of denoising. If data are given from non linear structure with additional noise, KPCA with a non linear kernel often succeeds in retrieving the underlying structure. This is typically very useful in image analysis, when noisy images are denoised by using score vectors obtained by KPCA (e.g. Mika et al. (1999)).

3. The influence function of Kernel PCA

Let P be a distribution on an arbitrary measurable space $\mathcal{X}$, let K be a reproducing kernel function on $\mathcal{X}$ with corresponding canonical feature space $\mathcal{H}$ and canonical feature map $\Phi : \mathcal{X} \to \mathcal{H}$ as in Definition 2. Recall that $\mathcal{H}$ is a Hilbert space of real functions on $\mathcal{X}$ and the reproducing property yields $\langle f, \Phi(x)\rangle = f(x)$ for every $f \in \mathcal{H}$, $x \in \mathcal{X}$. Throughout this section it is assumed that $\mathcal{H}$ is separable, $\Phi(x)$ is $P-$measurable $\forall x \in \mathcal{X}$ and $\mathbb{E}_P\|\Phi(X)\|^2 < \infty$. Define the centered covariance operator C_P by

$$C_P : \mathcal{H} \to \mathcal{H} : f \to C_P(f) = \mathbb{E}_P f(X)\Phi(X) - \mathbb{E}_P f(X)\mathbb{E}_P\Phi(X).$$

The operator C_P is a well-defined, compact, positive and self-adjoint Hilbert-Schmidt operator (Blanchard et al., 2007). Therefore it has a countable spectrum of positive eigenvalues $\lambda_{P,1} \geq \lambda_{P,2} \geq \ldots$ with an associated orthonormal basis of eigenfunctions $\{e_{P,i}\}$. Thus for any function $f \in \mathcal{H}$ we have that

$$f = \sum_{i=1}^{\infty} \langle f, e_{P,i}\rangle e_{P,i} \qquad \text{and} \qquad C_P(f) = \sum_{i=1}^{\infty} \lambda_{P,i}\langle f, e_{P,i}\rangle e_{P,i}.$$

Given a distribution P with $\mathbb{E}_P\|\Phi(X)\|^2 < \infty$, we denote by C, λ_i and e_i the maps such that $C(P) = C_P$, $\lambda_i(P) = \lambda_{P,i}$ and $e_i(P) = e_{P,i}$. The ith KPCA score $s_{P,i} \in \mathcal{H}$ in $x \in \mathcal{X}$ is the inner product of the ith eigenvector with the centered x: $s_{P,i}(x) = \langle e_{P,i}, \Phi(x) - \mathbb{E}_P\Phi(X)\rangle$.

We aim at quantifying the sensitivity of the maps λ_i and e_i under small changes of the underlying distribution P. To this end the influence function Hampel et al. (1986) is used.

Definition 3. *Given a statistical functional T mapping a distribution P onto $T(P)$. Consider the contaminated distribution*

$$P_{\epsilon,z} = (1-\epsilon)P + \epsilon\Delta_z$$

for small enough ϵ. The distribution Δ_z is the Dirac distribution which puts all probability mass at the point z. Then the influence function of T at the distribution P is defined as

$$IF(z; T, P) = \lim_{\epsilon \downarrow 0} \frac{T(P_{\epsilon,z}) - T(P)}{\epsilon}$$

in every point z where this limit exists.

Thus $IF(z; T, F)$ measures the effect on T under infinitesimally small contamination at the point z. For linear PCA the influence functions of the eigenvalues and the eigenvectors were derived by Critchley (1985).

3.1. Results

We prove the following theorem for KPCA.

Theorem 1. *Let P be a distribution on $\mathcal{X}$ such that $\mathbb{E}_P\|\Phi(X)\|^2 < \infty$. Let $i \in \mathbb{N}$ be such that $\lambda_{P,i} \neq \lambda_{P,j}$ for all $j \neq i$. Then the influence functions of λ_i and e_i at P in $z \in \mathcal{X}$ are given by*

$$IF(z; \lambda_i, P) = s_{P,i}(z)^2 - \lambda_{P,i}.$$

$$IF(z; e_i, P) = s_{P,i}(z) \sum_{j=1, j\neq i}^{\infty} \frac{s_{P,j}(z)}{\lambda_{P,i} - \lambda_{P,j}} e_{P,j}.$$

Proof

First note that

$$\mathbb{E}_{P_{\epsilon,z}}\|\Phi(X)\|^2 = (1-\epsilon)\mathbb{E}_P\|\Phi(X)\|^2 + \epsilon\|\Phi(z)\|^2 = (1-\epsilon)\mathbb{E}_P\|\Phi(X)\|^2 + \epsilon K(z,z).$$

Thus $\mathbb{E}_P\|\Phi(X)\|^2 < \infty$ implies that $\mathbb{E}_{P_{\epsilon,z}}\|\Phi(X)\|^2 < \infty$ and the operator $C_{P_{\epsilon,z}}$ is a well defined, positive, compact and self-adjoint Hilbert-Schmidt operator. Hence the maps λ_i and e_i exist at $P_{\epsilon,z}$ for any $\epsilon \in [0,1]$ and any $z \in \mathcal{X}$. Since $e_i(P)$ is a normalized eigenfunction for any P, it holds that

$$\langle e_i(P_{\epsilon,z}), e_i(P_{\epsilon,z})\rangle = 1.$$

Taking the derivative with respect to ϵ in $\epsilon = 0$ on both sides yields

$$\langle IF(z; e_i, P), e_{P,i}\rangle = 0.$$

Denote $\mathcal{H}^{\perp,i}$ the subspace of $\mathcal{H}$ orthogonal to the ith component. Then $IF(z; e_i) \in \mathcal{H}^{\perp,i}$. Furthermore we have that

$$\begin{aligned}\lambda_i(P_{\epsilon,z})e_i(P_{\epsilon,z}) &= C_{P_{\epsilon,z}}(e_i(P_{\epsilon,z})) \\ &= \mathbb{E}_{P_{\epsilon,z}}\langle e_i(P_{\epsilon,z}), \Phi(X)\rangle\Phi(X) - \mathbb{E}_{P_{\epsilon,z}}\langle e_i(P_{\epsilon,z}), \Phi(X)\rangle\mathbb{E}_{P_{\epsilon,z}}\Phi(X).\end{aligned}$$

Taking the derivative with respect to ϵ in $\epsilon = 0$, hereby using $P_{\epsilon,z} = (1-\epsilon)P + \epsilon\Delta_z$ and interchanging expectation and differentiation, yields

$$\begin{aligned}&IF(z;\lambda_i,P)e_i(P) + \lambda_i(P)IF(z;e_i,P) = -\mathbb{E}_P\langle e_i(P),\Phi(X)\rangle\Phi(X) + \langle e_i(P),\Phi(z)\rangle\Phi(z) \\ &+ \mathbb{E}_P\langle IF(z;e_i,P),\Phi(X)\rangle\Phi(X) - \mathbb{E}_P\langle IF(z;e_i,P),\Phi(X)\rangle\mathbb{E}_P\Phi(X) - \langle e_i(P),\Phi(z)\rangle\mathbb{E}_P\Phi(X) \\ &+ \mathbb{E}_P\langle e_i(P),\Phi(X)\rangle\mathbb{E}_P\Phi(X) - \mathbb{E}_P\langle e_i(P),\Phi(X)\rangle(\Phi(z) - \mathbb{E}_\Phi(X)).\end{aligned} \tag{5}$$

Observe that by definition of C_P it holds that

$$\mathbb{E}_P\langle IF(z;e_i,P),\Phi(X)\rangle\Phi(X) - \mathbb{E}_P\langle IF(z;e_i,P),\Phi(X)\rangle\mathbb{E}_P\Phi(X) = C_P(IF(z;e_i,P))$$

and

$$-\mathbb{E}_P\langle e_i(P),\Phi(X)\rangle\Phi(X) + \mathbb{E}_P\langle e_i(P),\Phi(X)\rangle\mathbb{E}_P\Phi(X) = -C_P(e_i(P)) = -\lambda_{P,i}e_i(P)$$

Thus equation (5) simplifies to

$$\begin{aligned}IF(z;\lambda_i,P)e_i(P) + \lambda_i(P)IF(z;e_i,P) &= -\lambda_{P,i}e_i(P) + C_P(IF(z;e_i,P)) \\ &+ \langle e_i(P),\Phi(z) - \mathbb{E}_P\Phi(X)\rangle(\Phi(z) - \mathbb{E}_P\Phi(X)).\end{aligned} \tag{6}$$

Moreover we have that

$$\langle C_P(IF(z;e_i,P)), e_i(P)\rangle = \langle IF(z;e_i,P), C_P(e_i(P))\rangle = \langle IF(z;e_i,P), \lambda_i(P)e_i(P)\rangle = 0$$

since $IF(z;e_i) \in \mathcal{H}^{\perp,i}$. Using this fact in (6) when taking the inner product of both sides with respect to $e_i(P)$ yields

$$IF(z;\lambda_i,P) = -\lambda_{P,i} + (\langle e_{P,i}, \Phi(z) - \mathbb{E}_P\Phi(X)\rangle)^2$$

proving the first statement. Substituting this result in equation (6) we have

$$(C_P - \lambda_i \mathrm{id}_{\mathcal{H}})(IF(z; e_i, P)) = (\langle e_{P,i}, \Phi(z) - \mathbb{E}_P\Phi(X)\rangle)^2 e_i(P) - \langle e_i(P), \Phi(z) - \mathbb{E}_P\Phi(X)\rangle(\Phi(z) - \mathbb{E}_P\Phi(X)). \tag{7}$$

Both $IF(z; e_i, P)$ and the right hand side of (7) are elements of $\mathcal{H}^{\perp,i}$. The operator $(C_P - \lambda_{P,i}\mathrm{id}_{\mathcal{H}})$ does not have an eigenvalue equal to 0 in $\mathcal{H}^{\perp,i}$. In that case the Fredholm alternative (see e.g. Phelps (1986)) shows that this operator is invertible and consequently the influence function equals the unique solution of (7). Moreover taking the inner product with $e_j(P)$ on both sides of (7) yields

$$(\lambda_{P,j} - \lambda_{P,i})\langle IF(z; e_i, P), e_j(P)\rangle = -\langle e_i(P), \Phi(z) - \mathbb{E}_P\Phi(X)\rangle\langle e_j(P), \Phi(z) - \mathbb{E}_P\Phi(X)\rangle$$

for any $j \neq i$. Thus

$$IF(z; e_i, P) = \langle e_i(P), \Phi(z) - \mathbb{E}_P\Phi(X)\rangle \sum_{j=1, j\neq i}^{\infty} \frac{\langle e_j(P), \Phi(z) - \mathbb{E}_P\Phi(X)\rangle}{\lambda_{P,i} - \lambda_{P,j}} e_{P,j}.$$

proving the second statement.

□

Recall that the influence function in a point $z \in \mathcal{X}$ provides information on the sensitivity with respect to adding an infinitesimally small amount of probability mass at position z. From the viewpoint of robustness it is important to have a bounded influence function, since the opposite means that the effect of an infinitesimally small distributional change can have an arbitrary large impact. From Theorem 1 it easily follows that the influence functions are bounded if the kernel is bounded.

Corollary 1. *For a bounded kernel, i.e. there exists $M > 0$ such that $||\Phi(z)||^2 = K(z, z) \leq M$, the following bounds hold:*

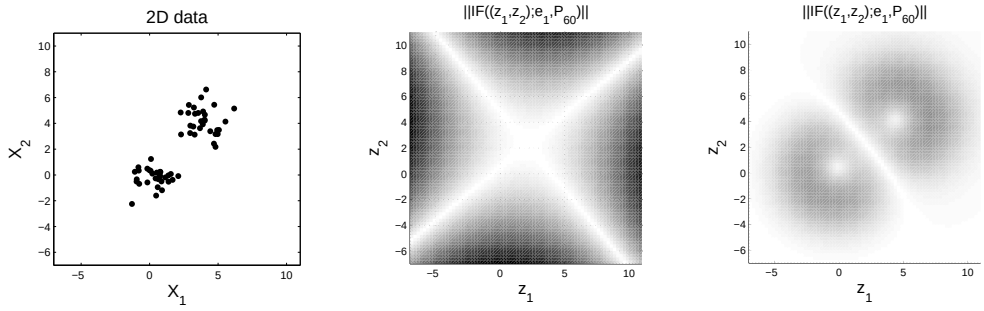
$$|IF(z; \lambda_i, P)| \leq 2M + \lambda_{P,i}.$$
$$||IF(z; e_i, P)|| \leq \frac{4M}{\min_j |\lambda_{P,i} - \lambda_{P,j}|}$$

In this sense KPCA with a bounded kernel is more robust than KPCA with an unbounded kernel. Similar results were obtained in the context of classification and regression with kernels (Steinwart and Christmann, 2008).

3.2. Empirical example

Let P_n be the empirical distribution of a sample $\{x_i \in \mathcal{X},\ i = 1, \ldots, n\}$. Applying the KPCA algorithm then yields the empirical eigenvalues $\lambda_{P_n,i}$ and the empirical score functions $s_{P_n,i}$. From Theorem 1 it follows that the norms of the influence function at the sample distribution P_n are easily computed for any $z \in \mathcal{X}$.

$$|IF(z; \lambda_i, P_n)| = |s_{P_n,i}(z)^2 - \lambda_{P_n,i}|.$$
$$||IF(z; e_i, P_n)|| = |s_{P_n,i}(z)| \left(\sum_{j=1, j\neq i}^{n} \frac{s_{P_n,j}(z)^2}{(\lambda_{P_n,i} - \lambda_{P_n,j})^2} \right)^{1/2}. \tag{8}$$



(*a*) (*b*) (*c*)

Figure 1: (*a*): Simple 2D data example; (*b*) – (*c*): $\|IF(z; e_1, P_{60})\|$ as a function of z. White represents values equal to 0, large values tend to black. (*b*): linear kernel; (*c*) RBF kernel.

A simple example is shown in Figure 1. Part (*a*) depicts the two dimensional artificially generated data set containing 60 observations. Denote by P_{60} the empirical distribution of this sample. Then the norm of the influence function (as in (8)) of the first KPCA eigenvector is shown as a function of $z = (z_1, z_2) \in \mathbb{R}^2$. In Figure 1(*b*) a linear kernel ($K(u, v) = u^t v$) is used. Figure 1(*c*) shows the result for a RBF kernel with $\sigma^2 = 3$. In these pictures values of 0 are shown as white and larger values tend to black. For the linear kernel the influence is 0 at the principal components themselves yielding the white cross in the picture. However, as z lies further away from these principal axes, the influence function increases as well. Moreover it keeps increasing attaining arbitrary large values as z trails away from the sample. For the RBF kernel a completely different effect takes place. The influence function is now local in the sense that it is large only at a small region of z values, i.e. at two half doughnut shaped regions around the centers of the two groups that are present in the data. The influence function is clearly bounded in this case, since the RBF kernel is bounded by 1 and thus Corollary 1 applies. This indicates an interesting difference between bounded and unbounded kernels in terms of robustness of kernel PCA. Similar conclusions were obtained for classification (Christmann and Steinwart, 2004) and regression (Christmann and Steinwart, 2007; Debruyne et al., 2008).

Of course this does not imply that bounded kernels are fully resistant against small amounts of contamination. The upper bound might be so large that the influence function can still attain very large values. From (8) and Corollary 1 it follows for instance that the difference between eigenvalues plays an important role. In this respect it is interesting to investigate the behavior of this difference for the RBF kernel. Let K be the RBF kernel with bandwidth σ and denote $r = \max_{k,l} \|x_k - x_l\|^2$. Since $1 - e^{-x} \leq x$ we have (using (4))

$$
\begin{aligned}
&|\,(\Omega_{c,\text{mean}})_{i,j}\,| \\
&= \left| (K(x_i, x_j) - 1) - \frac{1}{n}\sum_{l=1}^{n}(K(x_i, x_l) - 1) - \frac{1}{n}\sum_{k=1}^{n}(K(x_j, x_k) - 1) + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}(K(x_k, x_l) - 1) \right| \\
&\leq \frac{4r}{\sigma^2}.
\end{aligned}
$$

Using the Gershgorin circle theorem it follows that

$$
\exists i_1, j_1 : \quad |\lambda_{P_n,i} - (\Omega_{c,\text{mean}})_{i_1,j_1}| \leq (n-1)4\frac{r}{\sigma^2} \quad \text{and} \quad |\lambda_{P_n,j} - (\Omega_{c,\text{mean}})_{i_1,j_1}| \leq (n-1)4\frac{r}{\sigma^2}.
$$

Consequently

$$\frac{\sigma^2}{8(n-1)r} \leq \frac{1}{|\lambda_{P_n,i} - \lambda_{P_n,j}|}.$$

Thus although the influence function is bounded for the RBF kernel with any bandwidth σ, the bound in Corollary 1 itself can be arbitrary large as σ increases.

Finally also note that all theoretical analysis so far assumes fixed kernel parameters. In practice however the bandwidth of an RBF kernel is often chosen in a data dependent way. Then contamination can have a large impact on the choice of σ as well. Alzate and Suykens (2008) show some examples where KPCA with a RBF kernel with a data driven bandwidth selection severely suffers from a few influential observations in the data.

In summary it seems fair to say that although bounded kernels yield a bounded influence function, this does not automatically mean that bounded kernels guarantee good robustness in practice.

4. A robust alternative

In this section we propose an alternative PCA procedure which is robust for any type of kernel, whether it is bounded or not.

4.1. Robust centering

4.1.1. Spatial median in $\mathbb{R}^d$

The first step of PCA consists of centering the data, usually around the mean. However, the mean is not a robust measure of the center. Again one observation can have an arbitrary large influence. In this section we propose to use the L_1 M-estimate of location, which is a multivariate extension of the univariate median and which has been around for a long time (see for instance Haldane (1948) and Huber (1981)). This location measure is also known as the spatial median.

Definition 4. *Given a sample of inputs $x_i \in \mathbb{R}^d$, $i = 1, \ldots, n$. Then the spatial median θ is defined as the solution of*

$$\sum_{i=1}^{n} \frac{x_i - \theta}{||x_i - \theta||} = 0.$$

For the computation of this center, the following simple iterative algorithm exists (Gower, 1974; Huber, 1981; Hössjer and Croux, 1995). Given an initial guess $\theta^{(0)} \in \mathbb{R}^d$, iteratively define

$$\theta^{(k)} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

where

$$w_i = \frac{1}{||x_i - \theta^{(k-1)}||}.$$

Kuhn (1973) showed that this algorithm converges unless the starting point is in the domain of attraction of the data points. If the latter is the case, one can use the modification proposed by Vardi and Zhang (2000). However, in practice the simple algorithm above almost always converges.

4.1.2. Spatial median in feature space

Assume again that the inputs x_i are mapped into a high- (possibly infinite) dimensional feature space $\mathcal{H}$. Applying Definition 4 in feature space means that we want to find $\theta \in \mathcal{H}$ such that

$$\sum_{i=1}^{n} \frac{\Phi(x_i) - \theta}{\|\Phi(x_i) - \theta\|} = 0.$$

This is equivalent to demanding that

$$\left\| \sum_{i=1}^{n} \frac{\Phi(x_i) - \theta}{\|\Phi(x_i) - \theta\|} \right\|^2 = 0$$

or if we write out the norms as inner products

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \left\langle \frac{\Phi(x_i) - \theta}{\|\Phi(x_i) - \theta\|}, \frac{\Phi(x_j) - \theta}{\|\Phi(x_j) - \theta\|} \right\rangle = 0$$

which is equivalent to

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\langle \Phi(x_i), \Phi(x_j) \rangle - \langle \theta, \Phi(x_j) \rangle - \langle \theta, \Phi(x_i) \rangle + \langle \theta, \theta \rangle}{\sqrt{\langle \Phi(x_i), \Phi(x_i) \rangle - 2\langle \Phi(x_i), \theta \rangle + \langle \theta, \theta \rangle} \sqrt{\langle \Phi(x_j), \Phi(x_j) \rangle - 2\langle \Phi(x_j), \theta \rangle + \langle \theta, \theta \rangle}} = 0. \quad (9)$$

If the mapping Φ is explicitly known, one could use this equation to find the center θ. In most kernel applications this is of course not the case. However, the spatial median naturally lies in the space spanned by the n inputs, and any point in this $\min(n, d)$-dimensional space can be parametrized as a linear combination of the inputs. Thus the spatial median can be written as

$$\theta = \sum_{k=1}^{n} \gamma_k \Phi(x_k). \quad (10)$$

Using this representation in (9) we find

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \left\{ \frac{\langle \Phi(x_i), \Phi(x_j) \rangle - \sum_{k=1}^{n} \gamma_k \langle \Phi(x_k), \Phi(x_j) \rangle}{\sqrt{A_i}\sqrt{A_j}} \right.$$
$$\left. - \frac{\sum_{k=1}^{n} \gamma_k \langle \Phi(x_k), \Phi(x_i) \rangle + \sum_{k=1}^{n} \sum_{l=1}^{n} \gamma_k \gamma_l \langle \Phi(x_k), \Phi(x_l) \rangle}{\sqrt{A_i}\sqrt{A_j}} \right\} = 0 \quad (11)$$

with the notation

$$A_i = \langle \Phi(x_i), \Phi(x_i) \rangle - 2 \sum_{k=1}^{n} \gamma_k \langle \Phi(x_i), \Phi(x_k) \rangle + \sum_{k=1}^{n} \sum_{l=1}^{n} \gamma_k \gamma_l \langle \Phi(x_k), \Phi(x_l) \rangle.$$

Due to the parametrization of θ in (10), the spatial median can be expressed in terms of inner products only. Therefore this center can be computed in a kernel-induced feature space, using the same kernel trick as in kernel PCA replacing $\langle \Phi(u), \Phi(v) \rangle$ by $K(u, v)$.

Definition 5. *Given a sample of inputs $x_i \in \mathcal{X}$, $i = 1, \ldots, n$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R} : (u, v) \to K(u, v)$. Define the $n \times n$ kernel matrix as $\Omega_{i,j} = K(x_i, x_j)$. Denote $\Omega_{.,j}$ as the jth column of this matrix. Then the vector of coefficients $\gamma \in \mathbb{R}^n$ determining the spatial median in the kernel induced features space is defined by*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\Omega_{i,j} - \gamma^t \Omega_{.,j} - \gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}{\sqrt{\Omega_{i,i} - 2\gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}\sqrt{\Omega_{j,j} - 2\gamma^t \Omega_{.,j} + \gamma^t \Omega \gamma}} = 0.$$

To compute the vector γ the iterative algorithm in Section 4.1 can easily be modified to be computed in a kernel-induced feature space, only using the kernel inner product. Given an initial guess $\gamma^{(0)} \in \mathbb{R}^n$, iteratively define

$$\gamma^{(k)} = \frac{w}{\sum_{i=1}^{n} w_i}$$

where the components of the vector $w \in \mathbb{R}^n$ are given by

$$w_i = \frac{1}{\sqrt{\Omega_{i,i} - 2(\gamma^{(k-1)})^t \Omega_{.,i} + (\gamma^{(k-1)})^t \Omega \gamma^{(k-1)}}}.$$

For the starting point we take the coefficients corresponding to the mean: $\gamma^{(0)} = (1/n, \ldots, 1/n) \in \mathbb{R}^n$.

4.1.3. Centering the kernel matrix around the spatial median

The resulting center in the kernel feature space can of course not be computed. We do find the n coefficients γ_k such that the spatial median equals $\sum_{k=1}^{n} \gamma_k \Phi(x_k)$, but the feature map Φ is unknown. However, operations involving distances and inner products between feature vectors and the center often can be computed. A well known operation is for instance centering of the data. Suppose we want to center the data in feature space around the spatial median. We define a new feature map as

$$\tilde{\Phi}(x) = \Phi(x) - \sum_{i=1}^{n} \gamma_i \Phi(x_i).$$

The corresponding centered kernel function K_c becomes

$$\begin{aligned} K_c(x, z) &= \langle \tilde{\Phi}(x), \tilde{\Phi}(z) \rangle \\ &= \langle \Phi(x) - \sum_{i=1}^{n} \gamma_i \Phi(x_i), \Phi(z) - \sum_{i=1}^{n} \gamma_i \Phi(z_i) \rangle \\ &= K(x, z) - \sum_{i=1}^{n} \gamma_i K(x, x_i) - \sum_{i=1}^{n} \gamma_i K(z, x_i) + \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_i \gamma_j K(x_i, x_j) \end{aligned}$$

or expressed in terms of matrix operations on the kernel matrix:

$$\Omega_{c,\text{median}} = \Omega - \gamma 1_n^t \Omega - \Omega 1_n \gamma^t + \gamma^t \Omega \gamma 1_n 1_n' \tag{12}$$

where 1_n is a vector containing 1 in its n entries.

Thus first computing γ as in Definition 4 with the algorithm from the previous paragraph and then computing (12), gives a robustly centered kernel matrix centered around the spatial median instead of the mean.

4.2. Spherical KPCA

4.2.1. Spherical PCA in $\mathbb{R}^d$

Once the data is centered in an appropriate robust way, we can continue estimating the kernel principal components. We use the idea first mentioned in Locantore et al. (1999). Basically they project the data on a sphere around the L_1 median. Then the traditional components are computed for these projected data. Scores are computed by projecting the original, unsphered data on the principal directions. Due to the sphering the influence of the outlier is obviously heavily reduced leading to principal components capturing the structure of the majority of the data much better. Marden (1999) shows that these spherical principal components are exactly equal to the original ones at population level for a rather large class of distributions.

4.2.2. Spherical PCA in feature space

Assume that $\gamma \in \mathbb{R}^n$ is the vector of coefficients determining the spatial median in feature space $\sum_{k=1}^n \gamma_k \Phi(x_k)$. In the first step we project all feature vectors onto the unit sphere around the spatial median, giving us new feature vectors

$$\Phi^*(x_i) = \frac{\Phi(x_i) - \sum_{k=1}^n \gamma_k \Phi(x_k)}{||\Phi(x_i) - \sum_{k=1}^n \gamma_k \Phi(x_k)||}. \tag{13}$$

This implies that

$$\langle \Phi^*(x_i), \Phi^*(x_j) \rangle = \left\langle \frac{\Phi(x_i) - \sum_{k=1}^n \gamma_k \Phi(x_k)}{||\Phi(x_i) - \sum_{k=1}^n \gamma_k \Phi(x_k)||}, \frac{\Phi(x_j) - \sum_{k=1}^n \gamma_k \Phi(x_k)}{||\Phi(x_j) - \sum_{k=1}^n \gamma_k \Phi(x_k)||} \right\rangle.$$

In terms of the original and uncentered kernel matrix Ω, this leads to a new kernel matrix Ω^* with entries

$$\begin{aligned} \Omega^*_{i,j} &:= \langle \Phi^*(x_i), \Phi^*(x_j) \rangle \\ &= \frac{\Omega_{i,j} - \gamma^t \Omega_{.,j} - \gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}{\sqrt{\Omega_{i,i} - 2\gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma} \sqrt{\Omega_{j,j} - 2\gamma^t \Omega_{.,j} + \gamma^t \Omega \gamma}}. \end{aligned} \tag{14}$$

Thus once the spatial median is found, it is easy to compute the new kernel matrix Ω^* belonging to the sphered data based on the kernel matrix Ω of the original data.

In the second step, ordinary KPCA is applied to the sphered data. This means that we compute the eigenvectors and eigenvalues of Ω^* which we denote by $\alpha^{(k),*}$ resp. λ^*_k where $\lambda^*_1 \geq \lambda^*_2 \geq \ldots$ and $||\alpha^{(k),*}||^2 = 1$.

Thirdly the score $s^*_k(x)$ of any point x for the kth component is computed by

$$s^*_k(x) = \sum_{i=1}^n \frac{\alpha^{(k),*}_i}{\sqrt{\lambda_k}} \left\langle \Phi^*(x_i), \Phi(x) - \sum_{l=1}^n \gamma_l \Phi(x_l) \right\rangle.$$

Using (13) leads to the following result.

Algorithm 2 (Spherical Kernel PCA). *Given a sample $x_1, \ldots, x_n \in \mathcal{X}$. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel with corresponding feature space $\mathcal{H}$. Then the kth Sperical KPCA score function s^*_k at $x \in \mathcal{X}$ equals:*

$$s^*_k(x) = \sum_{i=1}^n \frac{\alpha^{(k),*}_i}{\sqrt{\lambda^*_k}} \frac{K(x_i, x) - \sum_{l=1}^n \gamma_k K(x_l, x) - \gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}{\sqrt{\Omega_{i,i} - 2\gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}}. \tag{15}$$

with $\alpha^{(k),*}$ the eigenvector belonging to the kth largest eigenvalue λ_k^* of the median centered and sphered kernel matrix Ω^* with entry i, j equal to

$$\Omega^*_{i,j} := \frac{\Omega_{i,j} - \gamma^t\Omega_{.,j} - \gamma^t\Omega_{.,i} + \gamma^t\Omega\gamma}{\sqrt{\Omega_{i,i} - 2\gamma^t\Omega_{.,i} + \gamma^t\Omega\gamma}\sqrt{\Omega_{j,j} - 2\gamma^t\Omega_{.,j} + \gamma^t\Omega\gamma}}. \tag{16}$$

These are the Spherical KPCA scores that provide a robust alternative to the classical KPCA scores from Algorithm 1. Note that the computational complexity of both algorithms is essentially the same. The most time consuming part consists of finding eigenvectors and eigenvalues of a $n \times n$ matrix. Also notice that Spherical KPCA, just like classical KPCA, only depends on the kernel matrix. No additional tuning constants are needed. This is a nice feature of the sphering concept compared to other types of robustification. Many robust algorithms for instance use reweighting. Then an appropriate weight function has to be chosen often implying an priori assumption on the distribution of the data. Especially in a general kernel induced feature space, this would be a difficult choice to make.

Note from (13) that all inner products are bounded between sphered feature vectors since

$$\langle \Phi^*(x_i), \Phi^*(x_j) \rangle \leq 1.$$

This means that the function $K^*(x_i, x_j) = \langle \Phi^*(x_i), \Phi^*(x_j) \rangle$ corresponding to the matrix Ω^* formally looks like a bounded kernel function. Since Spherical KPCA basically applies KPCA onto the sphered feature vectors, it is tempting to say that Spherical KPCA is nothing but ordinary KPCA with kernel K^* instead of K. Since K^* is bounded, the influence function of Spherical KPCA would be automatically bounded as well. Mathematically this is however not correct, since K^* is not a kernel function in the strict sense. To define K^* the spatial median is needed and the spatial median of course depends on the sample (or on the distribution in the continuous case). Implicitly the function K^* thus also depends on the distribution and this is not allowed to construct a valid kernel with corresponding feature space. Nevertheless it seems appropriate to conjecture that the influence function of Spherical KPCA is bounded due to the fact that the sphering part of the method maps everything into a bounded space For linear PCA a rigorous expression for the influence function of Spherical PCA was found by Croux et al. (2002). The extension of this result to arbitrary kernel feature spaces turns out to be mathematically involved, but can be interesting future work, potentially applying some of the ideas used by Gervini (2008) for functional PCA.

5. Visualizing influential observations

We propose a simple graphical display to assess the influence of observations with respect to ordinary KPCA. Our strategy is to use the spherical KPCA estimates in the expressions for the influence function in Theorem 1. For linear PCA this idea was applied by Pison and Van Aelst (2004). We use the score function $s_k^*(x)$ as a sample estimate of $s_{P,k}(x)$. However, as explained by Marden (1999) for linear PCA, the spherical eigenvalues λ_k^* are not always good estimates of $\lambda_{P,k}$ since they can be seriously biased. But since $\lambda_{P,k}$ equals the variance of the score function, one can re-estimate these eigenvalues by a measure of spread of the scores at the data points. For Spherical KPCA with a general kernel we propose the same strategy. Of course the measure of spread should not be influenced too much by individual observations either. We use the robust Median Absolute Deviation (MAD) to define

$$\lambda_k^{**} := \mathrm{MAD}(s_k^*(x_i)) = \left(\mathrm{median}(|s_k^*(x_i) - \mathrm{median}(s_k^*(x_i))|)\right)^2. \tag{17}$$

Other options, e.g. the Q_n-estimator (Rousseeuw and Croux, 1993), are possible as well of course. Now observe from Theorem 1 that

$$||IF(z; e_k, P)|| = |\langle e_{P,k}, \Phi(z)\rangle| \sqrt{\sum_{j=1, j\neq k}^{\infty} \frac{\langle e_{P,j}, \Phi(z)\rangle^2}{(\lambda_k - \lambda_j)^2}}$$

which gives us the following sample based influence diagnostic equal to the norm of the empirical influence function at z of the kth component:

$$||EIF_k(z)|| = |s_k^*(z)| \sqrt{\sum_{j=1, j\neq k}^{n} \frac{s_j^*(z)^2}{(\lambda_k^{**} - \lambda_j^{**})^2}}. \tag{18}$$

To obtain the influence of an observation x_i, just take $z = x_i$.

6. Examples

6.1. Toy example

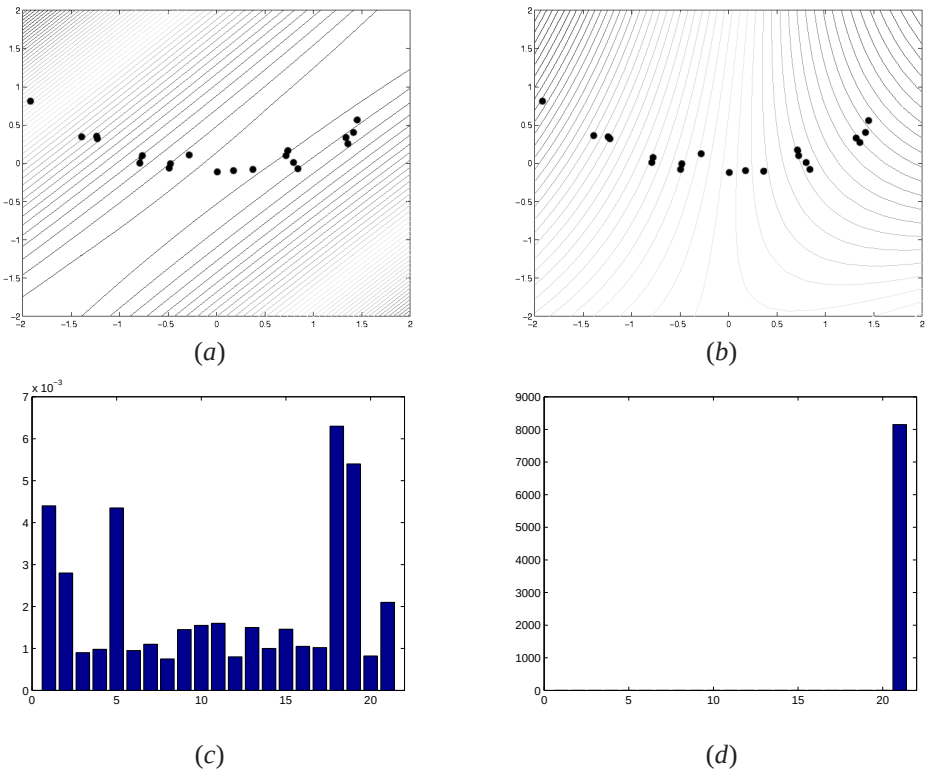


Figure 2: Score-contours and estimated influences for (*a*) – (*c*) classical KPCA, (*b*) – (*d*) spherical KPCA.

In Figure 2 we generated 20 data points showing a quadratic curvature, together with 1 outlier at $(-5, 5)$ (not visible on the plot). If we construct the score-contours corresponding to the first

	∅	a	c	g	ag	ca	cg	ga	gc	gg	gca	cag	ggc
gca	1	1	1	1	0	1	0	1	1	0	1	0	0
cag	1	1	1	1	1	1	1	0	0	0	0	1	0
ggc	1	0	1	2	0	0	0	0	1	1	0	0	1

Table 1: Explicit feature vectors with the all-subsequence kernel for the three strings gca, cag and ggc.

principal component of ordinary KPCA with a polynomial kernel of degree 2, we obtain Figure 2(*a*). Clearly the quadratic structure is lost completely due to the single outlier. For Spherical KPCA, Figure 2(*b*) depicts the corresponding score-contours. Now the quadratic structure of the majority of observations is learned, despite the outlier. Two bar plots of the empirical influence function from equation (18) for the 21 observations are shown in Figure 2 (*c*) using classical KPCA and (*d*) using spherical KPCA. A comparison of both plots visualizes the large difference between both methods. The diagnostic plot created with spherical KPCA (Figure 2(*d*)) correctly reveals that observation 21 (the outlier) causes this difference and that it is highly influential for classical KPCA.

6.2. String kernel

Consider a situation where the inputs are no vectors, but strings. Then many kernels exist that can be used to identify patterns in this set of strings. Here we concentrate on one example, i.e. the all-subsequence kernel. Then the strings are represented by feature vectors of which each component represents a possible substring. For the three strings ”gca”, ”cag” and ”ggc” for instance the corresponding feature vectors are shown in Table 1. So in this case every string can be represented as a vector with 13 components, and thus analysis could proceed in a 13-dimensional space. However, this example is extremely simple, since there are only three possible characters (a,c,g) and only strings of size three are considered. Unfortunately the dimension of the feature space increases exponentially with the size of the strings. For longer strings the explicit computation of the feature vectors thus becomes infeasible. However, when using a kernel method these explicit representations are not necessary. All we need are the inner products between any two feature vectors. For the all-subsequence kernel the kernel matrix containing these inner products can be computed with fast recursive algorithms (Shawe-Taylor and Cristianini, 2004). Since Spherical KPCA does not require explicit feature vectors either, but only the kernel matrix, applying the methodology from the previous sections is straightforward.

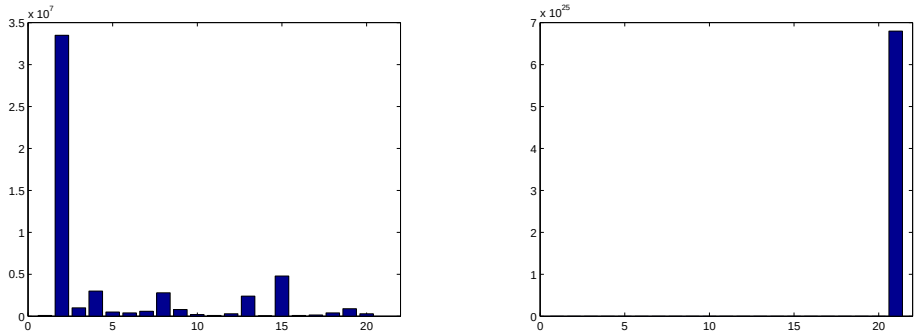
As an example take the first 20 DNA sequences in the ’Splice-junction gene sequences’ database from the UCI database (http://www.ics.uci.edu/~mlearn/MLRepository.html). This gives us 20 observations, all strings of size 60 composed out of 4 characters (A,C,G,T). The first 3 elements are shown below.

CCAGCTGCATCACAGGAGGCCAGCGAGCAGGTCTGTTCCAAGGGCCTTCGAGCCAGTCTG,
AGACCCGCCGGGAGGCGGAGGACCTGCAGGGTGAGCCCCACCGCCCCTCCGTGCCCCCGC,
GAGGTGAAGGACGTCCTTCCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATG.

As an example we add one strange string to the data set, observation 21, which is the following sequence:

CCCCCCCCCCCCCCCCAAAAAAAAAAAAAATTTTTTTTTTTTTTTGGGGGGGGGGGGGGGGG.

Although the length of this string and the number of A's,C's,G's and T's are both similar as for the other strings, this new observation 21 is clearly different due to the specific order of the characters. Next we perform KPCA and Spherical KPCA on this data set with the all-subsequence kernel. For each string we compute its influence measure as in (18) with respect to the first principal component. Figure 3(*a*) shows the result if we use the original KPCA scores and eigenvalues. String number 2 comes out as the most influential observation. Nevertheless it does not look extremely dominating and one would probably not suspect big problems. One would certainly not detect that observation 21 is an exceptional string, since its influence measure is very small. The results using Spherical KPCA are depicted in Figure 3(*b*). Then it is immediately



(*a*) (*b*)

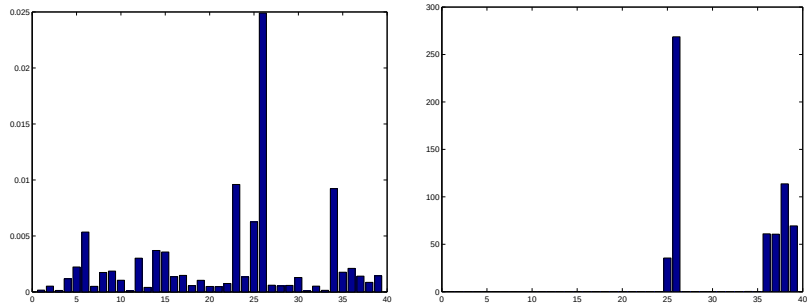
Figure 3: Estimated influences based on (*a*) KPCA, (*b*) Spherical KPCA.

clear that we have the same effect we discussed for the simple toy example in Figure 2. Observation 21 is in reality extremely influential, dominating the estimation of the ordinary first kernel principal component completely. This first pc is completely attracted by string 21. Therefore using this component results in a misleading plot of the influences. Only by using the spherical kernel principal components a correct assessment can be made about observations deviating from the mainstream. Also note that robust linear PCA methods cannot be used. They require the explicit feature vectors corresponding to the strings. However, according to Shawe-Taylor and Cristianini (2004), the dimension of these feature vectors would be likely to exceed 4^{30} in this example, which is obviously infeasible.

6.3. Octane data

The next example is the octane data set described in Esbensen et al. (1994). It contains near-infrared (NIR) absorbance spectra over 226 wavelengths of $n = 39$ gasoline samples with certain octane numbers. It is known that six of the samples $(25, 26, 36 - 39)$ contain added alcohol. The data set was also analyzed in Hubert et al. (2005), where it was shown that the robust linear PCA method ROBPCA was able to detect the six outlying samples in contrast to ordinary linear PCA. Now suppose that we increase the difficulty of the problem by using a polynomial kernel of degree 2. In theory the corresponding feature vectors could be computed by taking appropriately weighted squares and cross-products for all 226 variables. In practice the resulting dimension of these feature vectors will again be way too high. Explicitly calculating quadratic forms and then applying a robust method such as ROBPCA in feature space is thus infeasible.

Using a kernel method avoids this problem. All we need is the 39×39 dimensional kernel matrix, both for ordinary as spherical kernel PCA. The resulting diagnostic plots are shown in Figure 4. Part (*a*) of this plot depicts the results using ordinary KPCA. Of course some



(*a*) (*b*)

Figure 4: Octane data: estimated influences based on (*a*) KPCA, (*b*) Spherical KPCA.

points seem more influential than others, but no dramatic effects would be detected. Part (*b*) shows the influence measures using Spherical KPCA. Now we see what is really happening: six observations are extremely influential, dominating all others. These six observations are exactly the outlying samples that contain alcohol.

7. Conclusion

We investigated the effect of small amounts of contamination on classical Kernel PCA by calculating expressions for the influence function. We showed that the influence function can be unbounded if an unbounded kernel is used. Spherical Kernel PCA is proposed as a more robust alternative. The resulting algorithm is fast and only depends on the choice of the kernel, as in classical KPCA, so no additional tuning parameters are needed. We use this robust method to detect influential points in classical KPCA by a simple diagnostic plot. Examples illustrate that large differences can appear when outliers are present in the data. Using Spherical KPCA these outliers can be detected, whereas classical KPCA fails to do so.

Acknowledgements

M. Hubert acknowledges financial support by the GOA/07/04-project of the Research Fund KULeuven and by the IAP research network no. P6/03 of the Federal Science Policy, Belgium.

References

Alzate,C.,Suykens,J.A.K. Kernel component analysis using an epsilon insensitive robust loss function. IEEE T. Neural Networ. 19:1583–1598, 2008.

Blanchard,G.,Bousquet,O.,Zwald,L. Statistical properties of kernel principal component analysis. Mach. Learn. 33: 259–294, 2007.

Chen,T.,Martin,M.,Montague,G. Robust probabilistic PCA with missing data and contribution analysis for outlier detection. Comput. Stat. Dat. An. 53:3706–3716, 2009.

Christmann,A.,Steinwart,I. Consistency and robustness of kernel based regression. Bernoulli. 13:799-819, 2007.

Christmann,A.,Steinwart,I. On robust properties of convex risk minimization methods for pattern recognition. J. Mach. Learn. Res. 5:1007–1034, 2004.
Critchley,F. Influence in principal component analysis. Biometrika. 72:627–636, 1985.
Croux,C.,Ruiz-Gazen,A. High breakdown estimators for principal components: the projection-pursuit approach revisited. J.Multivariate.Anal. 95:206–226, 2005.
Croux,C.,Ollila,E.,Oja.,H. Sign and rank covariance matrices: statistical properties and application to principal components analysis. In Statistical Data Analysis Based on the L1-Norm and Related Methods, Birkhauser, Basel, pp. 257-271, 2002.
Debruyne,M.,Hubert,M.,Suykens,J.A.K. Model selection for kernel based regression using the influence function. *J. Mach. Learn. Res.*, 9:2377–2400, 2008.
Esbensen,K.H.,Schönkopf,S.,Midtgaard,T.. Multivariate Analysis in Practice. Camo, Trondheim, 1994.
Gervini,D. Robust functional estimation using the spatial median and spherical principal components. Biometrika, 95: 587-600, 2008.
Gower,J.C. The mediancentre. Appl. Stat. 23:466–470, 1974.
Haldane,J.B.S. Note on the median of a multivariate distribution. Biometrika, 35:414–415, 1948.
Hampel,F.R.,Ronchetti,E.M.,Rousseeuw,P.J.,Stahel,W.A. Robust Statistics: The Approach Based on Influence Functions. Wiley, New York, 1986.
Hössjer,O.,Croux,C. Generalizing univariate signed rank statistics for testing and estimating a multivariate location parameter. Non-parametric statistics. 4:293–308, 1995.
Huber,P.J. Robust Statistics. Wiley, New York, 1981.
Hubert,M.,Rousseeuw,P.J.,Verboven,S.. A fast robust method for principal components with applications to chemometrics. Chemometr. Intell. Lab. 60:101–111, 2002.
Hubert,M.,Rousseeuw,P.J.,Vanden Branden,K. ROBPCA: a new approach to robust principal components analysis. Technometrics. 47:64–79, 2005.
Hubert,M.,Rousseeuw,P.J.,Verdonck,T. Robust PCA for skewed data and its outlier map. Comput. Stat. Dat. An. 53: 2264–2274, 2009.
Kuhn,H.W. A note on Ferniat's problem. Math. Program. 4:98–107, 1973.
Liu,Z.,Chen,D.,Bensmail,H. Gene expression data classification with kernel principal component analysis. J. Biomed. Biotechnol. 2:155–169, 2005.
Liu,Z.,Chen,D.,Bensmail,H.,Xu,Y. Clustering gene expression data with kernel principal components. J Bioinf. Comput. Biol. 3:303–316, 2005.
Locantore,N.,Marron,J.S.,Simpson,D.G.,Tripoli,N.,Zhang,J.T.,Cohen,K.L. Robust principal component analysis for functional data. Test. 8:1–73, 1999.
Marden,J.I. Some robust estimates of principal components. Stat. Probabil. Lett. 43:349–359, 1999.
Maronna,R.A. Principal components and orthogonal regression based on robust scales. Technometrics. 47:264–273, 2005.
Mika,S.,Schölkopf,B.,Smola,A.,Müller,K.R.,Scholz,M.,Rätsch,G. Kernel PCA and denoising in feature spaces. In Advances in Neural Information Processing Systems, 11:536-542, MIT Press, Cambridge, MA, 1999.
Phelps,R. Convex functions, monotone operators and differentiability, volume 1364 of Lecture notes in math. Springer, 1986.
Pison,G.,Van Aelst,S. Diagnostic plots for robust multivariate methods. J. Comput. Graph. Stat. 13:310–329, 2004.
Pochet,N.,De Smet,F.,Suykens,J.A.K.,De Moor,B. Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. Bioinformatics. 20:3185–3195, 2004.
Rousseeuw,P.J.,Croux,C. Alternatives to the median absolute deviation. J. Am. Stat. Assoc. 88:1273–1283, 1993.
Schölkopf,B.,Smola,A. Learning with Kernels. MIT Press, Cambridge, MA, 2002.
Schölkopf,B.,Smola,A.,Müller,K.R. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10: 1299–1319, 1998.
Serneels,S.,Verdonck,T. Principal component analysis for data containing outliers and missing elements. Comput. Stat. Dat. An. 52:1712–1727, 2008.
Shawe-Taylor,J.,Cristianini,N. Kernel methods for pattern analysis. Cambridge university press, Cambridge, 2004.
Shawe-Taylor,J.,Williams,C.,Cristianini,N.,Kandola,J. Eigenspectrum of the gram matrix and its relationship to the operator eigenspectrum. In Algorithmic learning theory: 13th international conference, ALT2002 of lecture notes in computer science, 2533, pp. 23–40. Springer-Verlag, 2002.
Small,C.G. A survey of multidimensional medians. Int. Stat. Rev. 58:263–277, 1990.
Steinwart,I.,Christmann,A. Support Vector Machines. Springer, New York, 2008.
Vardi,Y.,Zhang,C.H. The multivariate L1-median and associated data depth. Proceedings of the National Academy of Sciences of the United States, 97:1423–1426, 2000.
Yang,J.,Frangi,A.F.,Yang,J.Y.,Zhang,D.,Jin,Z. KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition. IEEE T. Pattern Anal. 27:230–244, 2005.