Adv Data Anal Classif (2010) 4:151–167
DOI 10.1007/s11634-010-0068-1

REGULAR ARTICLE

Robust kernel principal component analysis and classification

Michiel Debruyne · Tim Verdonck

Received: 30 November 2009 / Revised: 17 May 2010 / Accepted: 25 May 2010 /
Published online: 24 June 2010

Abstract Kernel principal component analysis (KPCA) extends linear PCA from a real vector space to any high dimensional kernel feature space. The sensitivity of linear PCA to outliers is well-known and various robust alternatives have been proposed in the literature. For KPCA such robust versions received considerably less attention. In this article we present kernel versions of three robust PCA algorithms: spherical PCA, projection pursuit and ROBPCA. These robust KPCA algorithms are analyzed in a classification context applying discriminant analysis on the KPCA scores. The performances of the different robust KPCA algorithms are studied in a simulation study comparing misclassification percentages, both on clean and contaminated data. An outlier map is constructed to visualize outliers in such classification problems. A real life example from protein classification illustrates the usefulness of robust KPCA and its corresponding outlier map.

Keywords Principal component analysis · Kernel methods · Classification · Robustness

Mathematics Subject Classification (2000) 62H30 · 62G35

1 Introduction

Principal component analysis (PCA) is a well known technique in multivariate analysis to reduce the dimension of a data set by projecting onto a lower dimensional subspace.

M. Debruyne (✉) · T. Verdonck
Middelheimlaan 1G, 2020 Antwerp, Belgium
e-mail: michiel.debruyne@ua.ac.be

T. Verdonck
e-mail: tim.verdonck@ua.ac.be

Kernel PCA (Schölkopf et al. 1998) extends linear PCA by mapping the data into a high dimensional so-called feature space. Then ordinary linear PCA is performed in the feature space. In this computation only the pairwise inner products between feature vectors are required, not the explicit feature vectors. This makes it possible to apply the kernel trick: one replaces all inner products by a kernel function that is chosen beforehand to define the feature space (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 non numeric objects, 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).

For linear PCA it is well known that outliers can have a large impact on the resulting scores when applying the classical PCA algorithm maximizing the variance of projected observations. Several authors propose robust PCA algorithms that are less affected by outliers (Locantore et al. 1999; Hubert et al. 2002, 2005; Croux and Ruiz-Gazen 2005; Maronna 2005). In Debruyne et al. (2009a) it was shown that outliers are not only an issue for linear PCA: outliers can affect the results for KPCA with any unbounded kernel. They also proposed to use a kernel version of spherical PCA (Ohst 1988; Locantore et al. 1999) as a diagnostic tool to assess the influence of individual observations on classical KPCA. Moreover, in the computer science literature some approaches to apply KPCA in a robust manner have already been proposed (Takahashi and Kurita 2002; Lu et al. 2004; Alzate and Suykens 2008; Nguyen and De la Torre 2009). However, those methods are mostly restricted to the case of a Gaussian kernel and are not based on high breakdown methods in the linear case.

In this paper we will further focus on robust KPCA algorithms. We show that the projection pursuit (Croux and Ruiz-Gazen 2005) can be computed in any kernel feature space, yielding the kernel projection pursuit (KPP) algorithm. We consider an approach based on the ROBPCA method (Hubert et al. 2005) as well, yielding Kernel ROBPCA (KROBPCA). Secondly we will apply these robust KPCA algorithms in a classification context, by applying a simple linear classification method such as discriminant analysis on the KPCA scores. Seemingly a rather simple way of dealing with complex high-dimensional classification problems, this approach often produces competitive results. We will show that in this context of classification, dealing with outliers is very important as well. Using a robust KPCA method instead of the classical one can result in very large improvements in classification performance in presence of outliers. Furthermore we will be able to compare the performance of the different robust KPCA algorithms by comparing misclassification errors, both on clean and contaminated data. Finally an outlier map is proposed visualizing outliers in such classification problems.

The remainder of the paper is organized as follows. In Sect. 2 a short introduction is provided on KPCA. Three robust KPCA algorithms are described in Sect. 3. Classification with robust KPCA and robust linear discriminant analysis is explained in Sect. 4. Section 5 deals with the corresponding outlier map. A simulation study is discussed in Sect. 6 and Sect. 7 contains a real life example.

2 Kernel principal component analysis

Assume that we have a sample of n observations in some non-empty set $\mathcal{X}$: $x_i \in \mathcal{X}, i = 1, \dots, n$.

Definition 1 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 $\mathcal{H}$ and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that for all $x, x' \in \mathcal{X}$ we have

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle.$$

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

We require a Hilbert space, i.e. an inner space which is also complete and separable, to ensure that the feature space is defined in standard spaces with a coordinate system, such as L_2 or $\mathbb{R}^n$ for some n. Kernel PCA basically performs linear PCA in the 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. We will see in a moment that the kernel function K computes such products directly from the input data, i.e. without the need to construct the feature vectors themselves. Assume that the feature vectors are mean-centered. Denote Ω the matrix containing $\langle \Phi(x_i), \Phi(x_j) \rangle$ as i, jth entry. The first principal component equals the direction maximizing the variance of the projections onto this direction in $\mathcal{H}$. 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 $\Phi(x_i)$ $(i = 1, \dots, n)$. Let $\alpha = (\alpha_1, \dots, \alpha_n)^t \in \mathbb{R}^n$. Then the variance of the projection onto the direction α is

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

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

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

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

$$\left\langle \sum_{i=1}^{n} \frac{\alpha_i^{(1)}}{\sqrt{\lambda_1}} \Phi(x_i), \Phi(x) \right\rangle = \sum_{i=1}^{n} \frac{\alpha_i^{(1)}}{\sqrt{\lambda_1}} \langle \Phi(x_i), \Phi(x) \rangle . \tag{2}$$

It is now clear that expressions (1) and (2) depend on the feature vectors $\Phi(x_i)$ only through pairwise inner products. For a feature 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 f_k at $x \in \mathcal{X}$ equals:

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

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

$$\left(\Omega_{c,\text{mean}}\right)_{i,j} := \left(K(x_i, x_j) - \frac{1}{n} \sum_{k=1}^{n} K(x_k, x_j) - \frac{1}{n} \sum_{k=1}^{n} K(x_k, x_i) + \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l=1}^{n} K(x_k, x_l) \right). \tag{3}$$

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

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

for which the solutions are equivalent to classical linear PCA; the polynomial kernel of degree $p > 0$ with offset $\tau \in \mathbb{R}^+$

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

and the 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).

3 Robust KPCA algorithms

In this section we explain three robust KPCA algorithms. The spherical KPCA algorithm has already been described in Debruyne et al. (2009a). The kernel versions of projection pursuit and ROBPCA are new. Let us stress again that for a linear kernel, these algorithms are equivalent to the existing linear counterparts. For example the kernel projection pursuit algorithm with a linear kernel produces exactly the same scores as the linear projection pursuit algorithm from Croux and Ruiz-Gazen (2005). However, with the kernels versions other kernels can be used as well, making these algorithms applicable to a much broader framework. Some examples will be given in Sect. 6 with a polynomial kernel and Sect. 7 with a textstring kernel.

3.1 Spherical KPCA

Spherical PCA is discussed by Ohst (1988) and Locantore et al. (1999). Some theoretical properties are also proven by Marden (1999). The key idea is to project the data on a sphere around the spatial median, then performing PCA on the sphered data. Debruyne et al. (2009a) show that both the spatial median and the spherical PCA scores can be computed in any feature space yielding spherical kernel PCA (SKPCA).

Algorithm 2 (*Spatial median*) 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 $\mathcal{H}$. The spatial median in $\mathcal{H}$ can be written as a linear combination of the feature vectors. Denote $\gamma = (\gamma_1, \ldots, \gamma_n)^t$ the vector of coefficients, thus $\sum_{k=1}^n \gamma_k \Phi(x_k)$ equals the spatial median in feature space.
Initialize $\gamma^{(0)} = (1/n, \ldots, 1/n) \in \mathbb{R}^n$. Recursively compute

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

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

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

Then $\gamma^{(k)}$ converges to γ. Usually 10 steps is more than sufficient to obtain a good approximation with this algorithm. The corresponding centered kernel matrix containing the inner products between centered feature vectors around the spatial median is easily computed by

$$\Omega_{c,\text{med}} = \Omega - 1_n \gamma^t \Omega - \Omega \gamma 1_n^t + \gamma^t \Omega \gamma 1_n 1_n' \quad (4)$$

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

Once the spatial median is computed, the spherical KPCA scores are obtained by applying the following algorithm.

Algorithm 3 (*Spherical KPCA*) 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 spherical KPCA score function f_k at $x \in \mathcal{X}$ equals:

$$f_k(x) = \sum_{i=1}^{n} \frac{\alpha_i^{(k)}}{\sqrt{\lambda_k}} \frac{K(x_i, x) - \sum_{l=1}^{n} \gamma_k K(x_l, x) - \gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}{\sqrt{\Omega_{i,i} - 2\gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}}. \tag{5}$$

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

$$\Omega_{i,j}^* := \frac{\Omega_{i,j} - \gamma^t \Omega_{.,j} - \gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}{\sqrt{\Omega_{i,i} - 2\gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}\sqrt{\Omega_{j,j} - 2\gamma^t \Omega_{.,j} + \gamma^t \Omega \gamma}}. \tag{6}$$

3.2 Kernel projection pursuit

Projection pursuit (PP) (Friedman and Tukey 1974; Huber 1985) is one of the most successful methods for robust estimation for high-dimensional data, as it basically reduces the multivariate robustness problem to many univariate ones via projection. For PCA, projection pursuit comes down to constructing all possible directions in the space spanned by the data variables and then searching for the vector with maximal variance. When the first principal component is found, the data are projected on the subspace which is orthogonal to it and the procedure is started over again to find the second PC. Successive PCs are found likewise.

In order to obtain robust principal components, it suffices to replace the variance by a robust counterpart. In practice a robust scale estimator is used. In the earliest proposal (Li and Chen 1985), an M-estimator for scale is chosen. However, the M-estimators considered there do not have a high breakdown point. Simple high-breakdown estimators of scale are the median absolute deviation and the Q_n estimator (Rousseeuw and Croux 1993), which have a breakdown of 50%. Both these estimators have been examined for robust PCA and it was concluded that the Q_n estimator gave the best results (Croux and Ruiz-Gazen 1996, 2005). Mathematical properties of the PP estimator for PCA are obtained by Li and Chen (1985) and Cui et al. (2003).

Of course, in practice not all directions can be constructed and projection pursuit algorithms always yield approximative solutions to the maximization problem. It is then a question how the algorithm should be designed such that it constructs a limited number of directions while yielding a good approximation to the true solution and without suffering from numerical imprecision. Adapting the fast and simple algorithm of Croux and Ruiz-Gazen (2005) for approximating the PP-estimators to kernels leads to the following result.

Algorithm 4 (*Kernel PP*)

– For $k = 1$, define

$$A_{n,1} := \left\{ \frac{\Omega_{c,\mathrm{med}}^1 e_i}{\sqrt{e_i^t \Omega_{c,\mathrm{med}}^1 e_i}} | i = 1, \ldots, n \right\}$$

where $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^t$ is the ith basis vector and $\Omega^1_{c,\text{med}}$ is the centered kernel matrix around the spatial median (4).
Compute the scores on the first component as

$$y^1 = \underset{a \in A_{n,1}}{\text{argmax}}\ S_n(a) \tag{7}$$

where S_n is the robust scale estimator (e.g. Q_n estimator).
Denote $1 \leq l \leq n$ the index for which the maximum was obtained in (7).

– For $k = 2, \ldots, p$, define recursively
  1. The matrix

$$\left(\Omega^k_{c,\text{med}}\right)_{i,j} := \left(\Omega^{k-1}_{c,\text{med}}\right)_{i,j} - y_i^{k-1} \frac{\left(\Omega^{k-1}_{c,\text{med}}\right)_{l,j}}{\left(\Omega^{k-1}_{c,\text{med}}\right)_{l,l}} - y_j^{k-1} \frac{\left(\Omega^{k-1}_{c,\text{med}}\right)_{i,l}}{\left(\Omega^{k-1}_{c,\text{med}}\right)_{l,l}} + y_i^{k-1} y_j^{k-1}.$$

  2. The set $A_{n,k} := \left\{ \frac{\Omega^k_{c,\text{med}} e_i}{\sqrt{e_i^t \Omega^k_{c,\text{med}} e_i}} | i = 1, \ldots, n \right\}$.
  3. The scores on the kth principal component are computed as

$$y^k = \underset{a \in A_{n,k}}{\text{argmax}}\ S_n(a).$$

It is instructive to look at the very first step of the algorithm. Croux and Ruiz-Gazen (2005) consider each normed centered observation around the spatial median as a possible direction. In a feature space $\mathcal{H}$ the projection of the jth centered feature vector onto the direction determined by the normed ith centered feature vector is given by

$$\begin{aligned}
&\left\langle \Phi(x_j) - \sum_{k=1}^{n} \gamma_k K(x_j, x_k), \frac{\Phi(x_i) - \sum_{k=1}^n \gamma_k K(x_i, x_k)}{\|\Phi(x_i) - \sum_{k=1}^n \gamma_k K(x_i, x_k)\|} \right\rangle \\
&= \frac{\langle \Phi(x_j) - \sum_{k=1}^n \gamma_k K(x_j, x_k), \Phi(x_i) - \sum_{k=1}^n \gamma_k K(x_i, x_k)\rangle}{\langle \Phi(x_i) - \sum_{k=1}^n \gamma_k K(x_i, x_k), \Phi(x_i) - \sum_{k=1}^n \gamma_k K(x_i, x_k)\rangle} \\
&= \frac{(\Omega_{c,\text{med}})_{j,i}}{(\Omega_{c,\text{med}})_{i,i}}
\end{aligned}$$

The entire vector of projections on the ith centered feature vector thus equals

$$\frac{\Omega_{c,\text{med}} e_i}{\sqrt{e_i^t \Omega_{c,\text{med}} e_i}},$$

which is indeed the expression in the first step of Algorithm 4. For the sake of brevity we will not explain all other calculations in detail, but all steps in Algorithm 4 can be obtained in a similar way from the description in Croux and Ruiz-Gazen (2005).

3.3 Kernel ROBPCA

In the classical multivariate setting the Stahel–Donoho outlyingness (Stahel 1981; Donoho and Gasko 1992) of the ith observation $x_i = (x_{i1}, \dots, x_{ip})^T$ of a $n \times p$ data matrix $X = (x_1, \dots, x_n)^T$ is defined as:

$$r(x_i, X) = \sup_{v \in \mathbb{R}^p} \left\| \frac{x_i^t v - M(Xv)}{S(Xv)} \right\|, \tag{8}$$

where M and S denote location and scale estimators, respectively, and v is again a p-variate direction. Hubert et al. (2005) propose to use the univariate MCD (Rousseeuw 1984; Rousseeuw and Van Driessen 1999) estimators of location and scale for these purposes. Once r has been obtained for all data points, they keep the subset of $h < n$ data points having the smallest values for r. The value h determines the robustness as well as the efficiency of the method and has to be chosen in advance. A typical default value is $h \approx 0.75n$. They then compute the classical principal components of this subset and project the data on these principal components. In the original ROBPCA algorithm the covariance matrix of these projected points are then estimated by MCD (Rousseeuw 1984; Rousseeuw and Van Driessen 1999) and its eigenvectors are backtransformed to obtain the robust principal components. The robustness properties of this method are also known; they derive from the robustness properties of the outlyingness (8) and from those of the MCD estimator. The influence function of ROBPCA is bounded but nonsmooth (Debruyne and Hubert 2009).

Since the Stahel–Donoho outlyingness can be computed in any feature space (Debruyne 2009), a kernel version of ROBPCA is possible too.

Algorithm 5 (*Kernel ROBPCA*)

- For $d = 1 : 500$ (number of directions)
  1. Take a vector λ with all zeros except for 1 at position i and -1 at position j, where i and j are chosen at random. This vector represents a direction in $\mathcal{H}$ between two data points,
  2. Compute the vector containing all univariate projections of the feature vectors on the corresponding direction

$$a = \frac{\Omega\lambda}{\sqrt{\lambda^t \Omega \lambda}},$$

  3. Calculate the outlyingness for each observation on this direction

$$r^d = \frac{a - M(a)}{S(a)}.$$

4. Remember for each observation its maximum outlyingness.
- Take the h observations with lowest outlyingness.
- Apply classical kernel PCA on this data.

Note that the linear ROBPCA algorithm performs an additional step applying the MCD estimator in the PCA subspace. However, in the remainder of the paper we will focus on classification using the KPCA scores. Therefore we are only interested in the entire PCA subspace, not in the individual components spanning it. Thus the final step is useless in this setup, and therefore we do not consider it, although it might be worthwhile in other situations.

4 Classification with KPCA

Once the score functions are determined a new $n \times k$ score matrix can be computed containing the first k scores of the original inputs $x_j : f_i(x_j)$ for $j = 1, \dots, n$ and $i = 1, \dots, k$. Typically k is chosen relatively small, such that the scores are rather low dimensional. Moreover well chosen non linear kernels such as RBF or polynomial often capture potential non linearity in the original data, such that a simpler linear structure remains in the scores. This means that a simple linear discriminant method can be used on the new data $f_i(x_j)$, even if the original data x_j exhibits non-linearity.

This strategy is pretty successful in a binary classification context, when apart from inputs $x_i \in \mathcal{X}$ group labels y_i are given as well, with $y_i \in \{-1, 1\}$ indicating the group where input x_i belongs to. The goal of binary classification is to find a classification rule that assigns a label $y \in \{-1, 1\}$ to any new observation $x \in \mathcal{X}$. If $x \in \mathbb{R}^d$ with d small, then linear discriminant analysis can be used to construct such a classification rule. However, in high dimensional data with large d or when the inputs are not numerical, linear discriminant analysis fails. In those situations the following two stage algorithm can be applied. First apply KPCA with an appropriate kernel to obtain the scores $f_i(x_j)$. Then apply linear discriminant analysis on these k dimensional numerical score vectors. This algorithm performs well in many situations, see e.g. Liu et al. (2005) and Yang et al. (2005). Interestingly Yang et al. (2004) show that this two stage procedure coincides with Kernel Fisher discriminant analysis (Mika et al. 1999), which is in turn very related to least squares support vector machines (Suykens et al. 2002).

Robust classification in a similar way is now of course straightforward. First apply a robust KPCA algorithm, then proceed by robust linear discriminant analysis on the scores. For the latter we use the RDA algorithm proposed by Hubert and Van Driessen (2004), implemented in the LIBRA toolbox (Verboven and Hubert 2005). In the linear case this strategy was successfully used by Hubert and Engelen (2004) applying ROBPCA followed by RDA. By using any of the robust KPCA algorithms in Sect. 3, we are now able to perform such analysis with any type of kernel.

5 An outlier map

Using the robust estimators from the previous sections, an outlier map can be constructed to detect outliers. Following Hubert et al. (2005) this requires two ingredients:

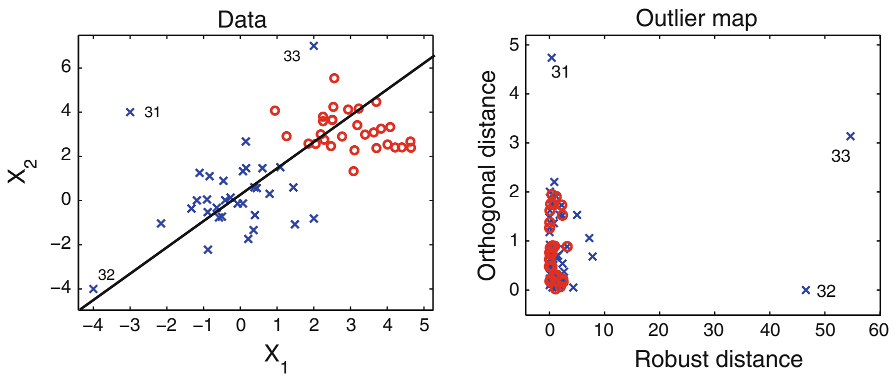

Fig. 1 Two dimensional toy example. *Left* data containing two groups and three outliers. *Right* outlier map revealing the three outliers and their type

orthogonal distances and score distances. First consider the orthogonal distance. This is the distance between a centered feature vector and its projection onto the k-dimensional subspace determined by the principal components. By Pythagoras' theorem this distance in $\mathcal{H}$ is given by

$$\mathrm{od}_i = (\Omega_c)_{i,i} - \sum_{j=1}^{k} f_j(x_i)^2.$$

The orthogonal distances are thus easily computed once the scores are obtained. It is also interesting to find outliers within the computed PCA subspace. To this end one can compute the robust distances as proposed by Hubert and Van Driessen (2004).

An interesting visualization tool is a scatterplot of the orthogonal distances on the vertical axis versus the robust distances on the horizontal axis. This provides an outlier map to detect outliers and to gain insight in the type of outlier. Consider for instance the simple two dimensional data shown in Fig. 1. Two groups of data were generated from two bivariate normal distributions. Three outliers labeled 31, 32 and 33 are added to the first group. The first principal direction computed by KPP is plotted as a solid black line. The outlier map constructed for this data set, with KPP and a linear kernel, is shown on the right hand side. The majority of observations from both groups are on the left lower side of the outlier map. This indicates a regular orthogonal distance and a regular robust distance, so these observations are the regular ones. However, the outliers clearly pop out on the outlier map. Moreover these three are easily categorized. Observation 31 is on the upper left side of the outlier map. This indicates that 31 has a large orthogonal distance, but once projected on the first principal component, it is not outlying anymore in this subspace. Observation 32 is the opposite case. Its orthogonal distance is small, meaning that it is not a dangerous outlier for PCA. But once projected in the PCA subspace, it is outlying within this space, with a large robust distance to the projected center of its group. Finally observation 33 combines both a large orthogonal distance as well as a large robust distance in the PCA subspace.

6 Simulation study

6.1 General setup

In this section we investigate the behavior of the kernel PCA methods combined with discriminant analysis in a simulation experiment. The simulation setup is based on the setup of Maronna and Zamar (2002) and Debruyne et al. (2009b). We examine clean and contaminated data, generated from the normal distribution. For each of the data configurations, the results of the classical and the robust kernel PCA algorithms with the classical and robust discriminant analysis are compared. In the simulation study we consider the linear and the polynomial kernel.

In all situations we simulated data with 50 cases for both classes and an independent test set with 1000 cases belonging to each class, each with a dimension of $p = 100$. In each simulation the percentage of misclassifications in the test set is counted. Means over 50 simulation runs are reported in a set of tables. After applying the kernel PCA method, we always retained one principal component. For the kernel PCA methods we used our own implementation, whereas the classical and robust discriminant analysis implementations are obtained from the Matlab library LIBRA (Verboven and Hubert 2005). All simulations were run in MATLAB R2007a (The MathWorks, Natick, MA).

6.2 Linear case

At first we consider a series of classification problems where the optimal solution is linear, such that linear PCA should be able to obtain good results. The two groups are generated from multivariate normal distributions with identical covariance matrices. The covariance matrix is taken as G^2 with $G_{jj} = 1$ and $G_{jk} = \rho$ (for $j \neq k$). The squared multiple correlation ρ^2_{mult}, which is the R^2 obtained by regressing any coordinate of the data matrix on all of the others, can be calculated as a function of ρ. By varying ρ^2_{mult} several levels of dependence can be considered (Maronna and Zamar 2002). In the simulations we have taken ρ such that ρ_{mult} equals respectively 0, 0.5 and 0.9.

The mean of the first group is always the origin. The center of the second group is $m(1, \dots, 1)^t$, with $m > 0$ determining the distance between both centers. We chose m as a function of optimal error rates. If we denote the optimal error rate by κ, then the corresponding m is easily found. We know that the optimal discriminating direction equals the first diagonal $e = \left(\frac{1}{\sqrt{p}}, \dots, \frac{1}{\sqrt{p}}\right)$. After projecting the data on this optimal direction e, the distance between both group centers equals $m\sqrt{p}$. Thus for the probability that an observation x from the first group is misclassified using the optimal discriminating direction to equal κ, it is required that

$$P\left(x \leq m\frac{\sqrt{p}}{2}\right) = 1 - \kappa.$$

The corresponding m thus equals $\frac{2}{\sqrt{p}}\phi^{-1}\left(1 - \kappa, 0, e^t Ge\right)$. In the simulation study we considered $\kappa = 0.01, 0.15$ and 0.30.

Table 1 Simulation results for the linear kernel; clean data

	CL-KPCA+ CLDA	SKPCA+ RLDA	KPP+ RLDA	KROBPCA+ RLDA
$\kappa = 0.01$				
$\rho_m = 0$	0.0177	0.0188	0.1696	0.0210
$\rho_m = 0.5$	0.0102	0.0101	0.0102	0.0101
$\rho_m = 0.9$	0.0106	0.0107	0.0107	0.0107
$\kappa = 0.15$				
$\rho_m = 0.5$	0.1519	0.1525	0.1526	0.1525
$\rho_m = 0.9$	0.1518	0.1526	0.1526	0.1526
$\kappa = 0.30$				
$\rho_m = 0.5$	0.3002	0.3005	0.3006	0.3007
$\rho_m = 0.9$	0.3012	0.3011	0.3010	0.3010

Results are shown in Table 1 for several values of the optimal error rate κ and the multiple correlation coefficient ρ_{mult}. We do not report the situation $\rho_{\text{mult}} = 0$ for $\kappa = 0.15$ and $\kappa = 0.30$. In both situations none of the methods were able to perform a decent classification, since all misclassification percentages were situated around 50%. Of course, if no method performs better than random guessing, there is not much point in comparing the results. However, in the other situations classical linear PCA plus classical LDA works very well, with error rates close to the optimal ones given by κ. The robust methods perform slightly worse, but this is of course expected since there are no outliers in this setup. Comparing the robust methods among each other, they produce very similar results except for the case $\kappa = 0.01$ and $\rho_m = 0$. There the projection pursuit approach is clearly unable to obtain a competitive result. Next the same situation is considered with contamination. We added 5 outliers to each group, generated from a multivariate normal distribution with independent components. The center of the group of outliers was taken along the direction $(1, -1, 1, -1, \ldots)$, thus orthogonal to the optimal discriminating direction. By putting the centers of the groups of outliers at large distance along this direction, the classical method breaks down. Results are reported in Table 2. The classical approach is no better than random guessing, but the robust alternatives are able to recover the two groups, showing misclassification errors that are only slightly higher than the case of clean data. Again PP performs rather bad in the case $\kappa = 0.01$ and $\rho_m = 0$.

6.3 Polynomial case

For the simulation in the previous subsection kernels are actually not necessary, since we only considered the linear kernel for which all algorithms are exactly equivalent to their linear counterparts from the literature. To investigate the behavior of Kernel PCA with other types of kernels we consider a similar setup with the polynomial kernel. The data are generated in a similar way as before, except that the data from the second class is now equally divided at location $-m(1, \ldots, 1)$ and $m(1, \ldots, 1)^t$. When choosing m the same as in the linear case, the corresponding κ is of course not necessarily the optimal error rate anymore. To obtain approximately a certain error

Table 2 Simulation results for the linear kernel; data with 10% contamination

	CL-KPCA + CLDA	SKPCA + RLDA	KPP + RLDA	KROBPCA + RLDA
$\kappa = 0.01$				
$\rho_m = 0$	0.4991	0.0205	0.1702	0.0214
$\rho_m = 0.5$	0.4990	0.0106	0.0103	0.0105
$\rho_m = 0.9$	0.4971	0.0105	0.0108	0.0104
$\kappa = 0.15$				
$\rho_m = 0.5$	0.4991	0.1547	0.1522	0.1539
$\rho_m = 0.9$	0.4979	0.1534	0.1521	0.1530
$\kappa = 0.30$				
$\rho_m = 0.5$	0.4992	0.3039	0.3004	0.3092
$\rho_m = 0.9$	0.4984	0.3018	0.3015	0.3019

Table 3 Simulation results for the polynomial kernel; clean data

	CL-KPCA+ CLDA	SKPCA+ RLDA	KPP+ RLDA	KROBPCA+ RLDA
$\kappa = 0.01$				
$\rho_m = 0$	0.0037	0.0059	0.0335	0.0033
$\rho_m = 0.5$	0.0020	0.0019	0.0019	0.0019
$\rho_m = 0.9$	0.0019	0.0020	0.0020	0.0020
$\kappa = 0.15$				
$\rho_m = 0.5$	0.1256	0.1167	0.1166	0.1167
$\rho_m = 0.9$	0.1264	0.1174	0.1174	0.1174
$\kappa = 0.30$				
$\rho_m = 0.5$	0.3424	0.3398	0.3398	0.3398
$\rho_m = 0.9$	0.3434	0.3405	0.3405	0.3405

rate κ, we multiplied the corresponding m from the linear case with a certain value. For $\kappa = 0.01, 0.15$ and 0.30 we multiplied respectively by 2, 1.5 and 1.25. By construction it is impossible for the linear methods to obtain good classification performance in this case. However, the polynomial kernel performs well as can be seen in Table 3. The robust KPCA algorithms generally perform well too. For the case $\kappa = 0.01$ and $\rho_m = 0$ there is again a clear difference between the robust methods. KPP is less able to keep up with the classical approach. The same setup is now considered with additional outliers. Results are shown in Table 4. The classical approach completely breaks down. The robust KPCA algorithms are able to recover the group structure despite the outliers. Again the case $\kappa = 0.01$ and $\rho_m = 0$ turns out to be difficult for KPP, whereas SKPCA and KROBPCA obtain overall good and comparable results.

7 Example

The protein data set taken from Pollack et al. (2005) contains 131 protein sequences. It is available in the Protein Classification Benchmark Collection at http://net.icgeb.org (accession number PCB00015). We consider here classification task number 10

Table 4 Simulation results for the polynomial kernel; data with 10% contamination

	CL-KPCA + CLDA	SKPCA + RLDA	KPP + RLDA	KROBPCA + RLDA
$\kappa = 0.01$				
$\rho_m = 0$	0.5000	0.0069	0.0267	0.0036
$\rho_m = 0.5$	0.5000	0.0023	0.0023	0.0023
$\rho_m = 0.9$	0.5000	0.0021	0.0022	0.0022
$\kappa = 0.15$				
$\rho_m = 0.5$	0.5000	0.1258	0.1258	0.1220
$\rho_m = 0.9$	0.5000	0.1261	0.1260	0.1142
$\kappa = 0.30$				
$\rho_m = 0.5$	0.5000	0.3410	0.3410	0.3412
$\rho_m = 0.9$	0.5000	0.3416	0.3416	0.3418

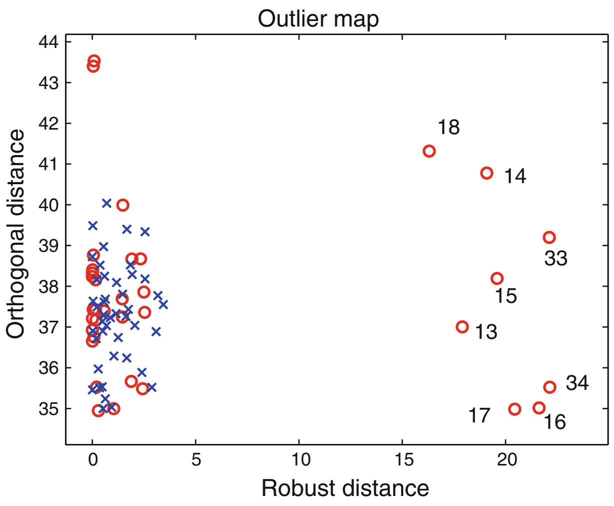

Fig. 2 Outlier map for the protein data set

where the positive group consists of 35 Eukaryota. The negative group consists of 4 Archaea and 40 Bacteria. Note that observations are now strings instead of numerical values. However, by choosing an appropriate kernel such data are easily analyzed. We use the local alignment kernel (Saigo et al. 2004) with default parameter values: gap opening penalty = 11, gap extension penalty = 1, scaling parameter = 0.5.

Applying KROBPCA with this kernel, we find that the first principal component explains 45% of all variance. Subsequent components do not add more than 7% to the fraction of explained variance, indicating that the first principal component is of primary importance. Thus we will retain the first principal component. After projection we can apply RDA. The resulting outlier map is shown in Fig. 2. The group of Eukaryota forms a very homogeneous group as well as a majority of the group of Bacteria. However, a few protein sequences in the group of Bacteria are outlying: 13–18 and 33–34. As it turns out, the group of Bacteria is indeed a heterogeneous group containing several subgroups. Samples 13–17 are such a subgroup, named the Euglenozoa. Note that 18 (named Q8SRZ8), which belongs to the Fungi, was clustered in the group of Euglenozoa by Pollack et al. (2005); this is actually confirmed by the outlier map. Samples 33 and 34 are outlying as well. They form, together with 32, the group of Stramenopiles. Note that the different behavior of sample 32 from its fellow Stramenopiles is again a confirmation of the analysis by Pollack et al. (2005): their clustering method assigned 32 (named Q8H721) in the main group of Eukaryota Metazoa. Also in the outlier map 32 is situated in the main group, whereas 33 and 34 are clearly outlying with respect to the main group.

Finally, to assess the classification performance in this example we used 10-fold cross validation. KROBPCA then misclassifies 8 observations out of 79. The other robust KPCA algorithms obtain the same performance. However, on close inspection of these 8 misclassifications, it turns out that they are exactly the 8 outliers appearing on the outlier map. Since they belonged indeed to different biological subgroups, it is not surprising that we are unable to classify these protein strings correctly. Thus robust KPCA plus RDA performs well in detecting the outlying protein strings and correctly classifying all strings from the main group.

8 Conclusion

The PP and ROBPCA methods for robust linear PCA are extended to a more general kernel framework. A two stage algorithm combining a robust KPCA method with robust linear discriminant analysis was investigated for classification in kernel feature spaces. A simulation study is performed comparing classical KPCA, KPP, KROBCPA and SKPCA for the linear and the polynomial kernel. We showed that outliers are a genuine concern, also for the polynomial kernel, and that the use of robust KPCA can improve classification performance dramatically if the data are contaminated. In the most noisy situations KPP sometimes performed significantly worse than SKPCA and KROBPCA. In the linear case this confirms results by Croux et al. (2007) where the relatively bad behavior of PP in several high dimensional noisy situations was shown. Note that they also propose an adaption of the PP algorithm to improve this behavior, but this algorithm cannot be computed in any kernel feature space. Thus as a robust KPCA algorithm we would suggest to use KROBPCA or SKPCA rather than KPP, although our results are of course limited to some specific simulation settings and by no means conclusive. An outlier map has been introduced to visualize different types of outliers in feature space. A real life example from protein string analysis illustrated the practical relevance of robust KPCA and its outlier map.

References

Alzate C, Suykens JAK (2008) Kernel component analysis using an epsilon-insensitive robust loss function. IEEE Trans Neural Netw 19:1583–1598

Croux C, Ruiz-Gazen A (1996) A fast algorithm for robust principal components based on projection pursuit. In: COMPSTAT: Proceedings in computational statistics, pp 211–216

Croux C, Ruiz-Gazen A (2005) High breakdown estimators for principal components: the projection-pursuit approach revisited. J Multivar Anal 95:206–226

Croux C, Filzmoser P, Oliveira MR (2007) Algorithms for projection-pursuit robust principal component analysis. Chemom Intell Lab Syst 87:218–225

Cui H, He X, Ng KW (2003) Asymptotic distributions of principal components based on robust dispersions. Biometrika 90:953–966

Debruyne M (2009) An outlier map for support vector machine classification. Ann Appl Stat 3(4):1566–1580

Debruyne M, Hubert M (2009) The influence function of the Stahel-Donoho covariance estimator of smallest outlyingness. Stat Probab Lett 79:275–282

Debruyne M, Hubert M, Van Horebeek J (2009a) Detecting influential observations in Kernel PCA. Comput Stat Data Anal (in press). doi:10.1016/j.csda.2009.08.018

Debruyne M, Serneels S, Verdonck T (2009b) Robustified least squares support vector classification. J Chemometrics 23(9):479–486

Donoho DL, Gasko M (1992) Breakdown properties of location estimates based on half-space depth and projected outlyingness. Ann Stat 20:1803–1827

Friedman JH, Tukey JW (1974) A projection pursuit algorithm for exploratory data analysis. IEEE Trans Comput C-23(9):881–890

Huber PJ (1985) Projection pursuit. Ann Stat 13:435–475

Hubert M, Engelen S (2004) Robust PCA and classification in biosciences. Bioinformatics 20:1728–1736

Hubert M, Van Driessen K (2004) Fast and robust discriminant analysis. Comput Stat Data Anal 45:301–320

Hubert M, Rousseeuw PJ, Verboven S (2002) A fast robust method for principal components with applications to chemometrics. Chemom Intell Lab Syst 60:101–111

Hubert M, Rousseeuw PJ, Vanden Branden K (2005) ROBPCA: a new approach to robust principal components analysis. Technometrics 47:64–79

Li G, Chen Z (1985) Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and Monte Carlo. J Am Stat Assoc 80:759–766

Liu Z, Chen D, Bensmail H (2005) Gene expression data classification with kernel principal component analysis. J Biomed Biotechnol 2:155–169

Locantore N, Marron JS, Simpson DG, Tripoli N, Zhang JT, Cohen KL (1999) Robust principal component analysis for functional data. Test 8:1–73

Lu C-D, Zhang T-Y, Du X-Z, Li C-P (2004) A robust kernel PCA algorithm. Proc Int Conf Mach Learn Cybernet 5:3084–3087

Marden JI (1999) Some robust estimates of principal components. Stat Probab Lett 43:349–359

Maronna RA (2005) Principal components and orthogonal regression based on robust scales. Technometrics 47:264–273

Maronna RA, Zamar R (2002) Robust estimates of location and dispersion for high-dimensional data sets. Technometrics 44:307–317

Mika S, Rätsch G, Weston J, Schölkopf B, Müller KR (1999) Fisher discriminant analysis with kernels. In: IEEE international workshop on neural networks for signal processing IX, pp 41–48

Nguyen MH, De la Torre F (2009) Robust kernel principal component analysis. Adv Neural Inf Process Syst 21:1185–1192

Ohst C (1988) Beste approximierende Kreise und ihre Eigenschaften (Best approximating spheres and their properties). Diplomarbeit in Mathematik, Institut für Statistik und Wirtschaftsmathematik, RWTH Aachen University

Pollack JD, Li Q, Pearl DK (2005) Taxonomic utility of a phylogenetic analysis of phosphoglycerate kinase proteins of Archaea, Bacteria, and Eukaryota: insights by Bayesian analyses. Mol Phylogenet Evol 35:420–430

Rousseeuw PJ (1984) Least median of squares regression. J Am Stat Assoc 79:871–880

Rousseeuw PJ, Croux C (1993) Alternatives to the median absolute deviation. J Am Stat Assoc 88:1273–1283

Rousseeuw PJ, Van Driessen K (1999) Fast algorithm for the minimum covariance determinant estimator. Technometrics 41:212–223

Saigo H, Vert J, Ueda N, Akutsul T (2004) Protein homology detection using string alignment kernels. Bioinformatics 20:1682–1689

Schölkopf B, Smola A (2002) Learning with kernels. MIT Press, Cambridge

Schölkopf B, Smola A, Müller K-R (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10:1299–1319

Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge university press, Cambridge

Stahel WA (1981) Robuste Schätzungen: Infinitesimale Optimalität und Schätzungen von Kovarianzmatrizen. PhD thesis, ETH Zürich

Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J (2002) Least squares support vector machines. World Scientific, Singapore

Takahashi T, Kurita T (2002) Robust de-noising by kernel PCA. In: Proceedings of the international conference on artificial neural networks. Lecture notes in computer science, vol 2415, pp 739–744

Verboven S, Hubert M (2005) LIBRA: a MATLAB library for robust analysis. Chemom Intell Lab Syst 75:127–136

Yang J, Jin Z, Yang JY, Zhang D, Frangi AF (2004) Essence of kernel Fisher discriminant: KPCA plus LDA. Pattern Recognit 37:2097–2100

Yang J, Frangi AF, Yang JY, Zhang D, Jin Z (2005) KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition. IEEE Trans Pattern Anal Mach Intell 27:230–244