Robust PCA for high-dimensional data based on characteristic transformation

Lingyu He¹, Yanrong Yang² and Bo Zhang³,*

Hunan University, The Australian National University and University of Science and Technology of China

Summary

In this paper, we propose a novel robust principal component analysis (PCA) for high-dimensional data in the presence of various heterogeneities, in particular strong tailing and outliers. A transformation motivated by the characteristic function is constructed to improve the robustness of the classical PCA. The suggested method has the distinct advantage of dealing with heavy-tail-distributed data, whose covariances may be non-existent (positively infinite, for instance), in addition to the usual outliers. The proposed approach is also a case of kernel principal component analysis (KPCA) and employs the robust and non-linear properties via a bounded and non-linear kernel function. The merits of the new method are illustrated by some statistical properties, including the upper bound of the excess error and the behaviour of the large eigenvalues under a spiked covariance model. Additionally, using a variety of simulations, we demonstrate the benefits of our approach over the classical PCA. Finally, using data on protein expression in mice of various genotypes in a biological study, we apply the novel robust PCA to categorise the mice and find that our approach is more effective at identifying abnormal mice than the classical PCA.

Key words: characteristic function; heavy-tailed data; high-dimensional data; kernel PCA; robust PCA; spiked covariance model.

1. Introduction

Principal component analysis (PCA) (Jolliffe 2002; Anderson 2003) is a widely used technique for dimension reduction and data exploration. High-dimensional data are encountered frequently due to the rapid development of modern technologies; therefore, PCA is gaining popularity due to its capacity to summarise high-dimensional data using some low-dimensional projections (see e.g. Donoho 2000; Johnstone & Titterington 2009; Yata & Aoshima 2012; Lee, Zou & Wright 2014; Shen et al. 2016b; Morales-Jimenez et al. 2021). Mathematically, classical PCA is based on the eigen-decomposition of the population covariance matrix, and the leading eigenvectors serve as the directions of the projections. However, the population covariance matrix is highly sensitive to the population distribution and data quality, such as heavy-tailed distributions and outliers in the data. In probability
theory, heavy-tailed distributions are probability distributions whose tails are not exponentially bounded, that is, they have heavier tails than the exponential distribution. It is therefore difficult for classical PCA to process such data with poor quality. As dimensionality increases, heterogeneity—the variety of statistical characteristics of the data—becomes more common. For example, heavy-tailed variables are more likely to appear along with the normal distributed features in high-dimensional data, and the classical PCA is sensitive to this type of heterogeneity, as shown in Li & Chen (1985) and He et al. (2020). Other types of heterogeneity include heteroscedastic noise and outliers. In view of this fact, it is of great urgency to develop new dimension-reduction approaches for the high-dimensional regime to deal with heterogeneities.

In this paper, we propose a new robust dimension-reduction approach for high-dimensional data. The new method is particularly helpful for data coming from heavy-tailed distributions, which has also been taken into account in recent works such as He et al. (2020) and Chen, Dolado & Gonzalo (2021). Consider a scenario in which the population distributions of the data have infinite second moments or even infinite first moments. In this case, any dimension-reduction method that relies on those moments, such as the classical PCA and the robust covariance-based PCA (see e.g. Croux & Haesbroeck 2000) are invalid. Motivated by this difficulty, we propose a novel robust PCA, with the key step being the transformation of the original data according to the shape of the characteristic function. As discussed in Baxter (1995), transforming unusually distributed data before analysis provides special merits. Our proposed method is robust to different styles of heterogeneity, particularly to data with infinite population moments. The transformation’s appealing qualities are primarily where the robustness originates from. Recall that the characteristic function of a real-valued random variable $y$ is $\phi(t) = E\{\exp\{ity\}\}$ ($t \in \mathbb{R}, i^2 = -1$), which completely describes the probability distribution of $y$ and $|\exp\{ity\}| = 1$ for any $t$. As a result, the transformation $z_i = \exp\{iy_i\}$ ($i \in \{1, \ldots, p\}$) preserves the distribution information of $y_i$ ($i \in \{1, \ldots, p\}$), and more importantly, is a bounded random variable no matter whether $y_i$ ($i \in \{1, \ldots, p\}$) is bounded or not. Thus, by performing the classical PCA on the transformed variables $z_i$ ($i \in \{1, \ldots, p\}$), the suggested method is robust to heavy-tailed data.

The non-robustness issue of PCA has been studied in the area of robust statistics. A natural and straightforward solution is to use a more robust estimator in place of the sample covariance matrix. Croux & Haesbroeck (2000) investigated the influence functions and efficiency of some robust covariance matrix estimators. Li & Chen (1985) suggested an alternative method that measures the dispersion of the projections by using a projection pursuit index rather than variance. Locantore et al. (1999) proposed a robust PCA method by projecting the original data on the unit sphere (centred on the spatial median). Subsequently, Li et al. (2022) examined the covariance matrix’s properties for the transformed data in high-dimensional regimes. The study of PCA in terms of a low-rank matrix approximation problem and the minimisation of the robust loss function has sparked interest in computer science (see e.g. Candès et al. 2011). Recently, He et al. (2020) and Chen, Dolado & Gonzalo (2021) investigated the effect of heavy-tailedness in large-dimensional factor analysis using a PCA-based approach. Further reviews can be found at She, Li & Wu (2016) and Vidal, Ma & Sastry (2016). Our proposed method primarily contributes to robust methods for heavy-tailed data without finite moments, which is similar to the data structures considered in He et al. (2020). We project the original data to a different space to achieve robustness, which is similar to Locantore et al. (1999) but with a different transformation.
The proposed method is also a special case of Kernel PCA, which is commonly used in pattern recognition and novelty detection (see e.g. Mika et al. 1998; Hoffmann 2007; Vidal, Ma & Sastry 2016). Kernel PCA, like our proposed method, first maps the data into some feature space using a function and then performs PCA on the mapped data. The corresponding kernel function is given in Remark 2. Debruyne, Hubert & Van Horebeek (2010) demonstrated that kernel PCA with a bounded kernel is more robust than kernel PCA with an unbounded kernel. Because our kernel function is bounded, this further supports the robustness of our method. Furthermore, because the kernel function is non-linear, the proposed method aids in the exploration of the original data’s non-linear relationship. Following the literature on Kernel PCA (Blanchard, Bousquet & Zwald 2007), we study the properties of the reconstruction errors and the excess error (the difference between the optimal (population) reconstruction error and the empirical reconstruction error) of the PCA methods in Section 3.1. As Kernel PCA projects data into a feature space and does not require pre-images in the original space (Mika et al. 1998), the corresponding errors for our method refer to the transformed data. Even if the original data do not have finite variances, the upper bound of the excess error of our method can be relatively small.

In addition to heterogeneity, high dimensionality of the data itself is a critical issue. Works such as Johnstone (2001), Lam, Yao & Bathia (2011), Yata & Aoshima (2012), Lee, Zou & Wright (2014), Shen et al. (2016b), Wang & Fan (2017), Cai, Han & Pan (2017) have made efforts to understand the behaviour of empirical eigenvalues in various high-dimensional settings. However, for heavy-tailed data without finite variances, the results of that literature are no longer applicable to the original data, but remain valid on the transformed data. Furthermore, in the statistical study of sample covariance matrices, the Johnstone’s spiked covariance matrix model (Johnstone 2001) is a popular model. A few spikes, or eigenvalues separated from the bulk eigenvalue spectrum, are added to the population covariance matrix in this model. According to the literature, the stronger spikeness of the population and the larger sample size allow larger dimension in consistently recovering the population eigenvalues from the empirical eigenvalues. We are interested in how the transformation in our method affects the spike covariance structure when the original data are normal distributed and have a spiked covariance structure. We explore the behaviour of the largest $k$ eigenvalues for the original data and the transformed data under various settings using simulations. Under our simulated data, we observe that the transformation keeps the spiked structure. For the heavy-tailed data, the empirical eigenvalues of the classical PCA vary widely, but the robust PCA produces more stable results.

Furthermore, the reconstruction of the original data in our method is as important as the one in Kernel PCA (Mika et al. 1998). With several examples, we demonstrate the advantage of our proposed method over classical PCA in terms of mean squared reconstruction error (MSE). These examples include data with heterogeneity in variances, data with outliers, and data from three different heave-tailed distributions. Overall, we find that our proposed method recovers these data more accurately than classical PCA. Lastly, we show how the method can be used in real-world data analysis by analysing mice protein expression measurements from Higuera, Gardiner & Cios (2015). Because the majority of the proteins have heavy tails or extreme outliers in their expression levels, it is essential to apply robust methods to analyse the data. The proposed method is used to classify mice based on their protein expression data into different genotypes. Compared to classical PCA, our proposed method can more accurately identify mice with abnormal genotypes.

© 2023 Statistical Society of Australia.
The rest of this paper is organised as follows. Section 2 describes our proposed method in detail. Section 3 investigated the statistical properties. Simulations to illustrate the reconstruction performance under different cases are presented in Section 4. Section 5 gives an example of a real data application. Section 6 concludes the paper.

Some notations are given here. For any vector $a$, $\|a\|$ is the $\ell_2$-norm; for any matrix $A$, $\|A\|$ is the spectral norm, that is defined as $\|A\| = \sup_{\|u\| = 1} \|Au\|$; and for a scalar $a$, $|a|$ is the absolute value of $a$.

2. Methodology

Let us recall the settings of the classical principal component analysis (PCA). Assume we have $n$ data points $y_1, \ldots, y_n$ that were generated by a random vector $y = (y_1, \ldots, y_p)^\top \in \mathbb{R}^p$. Classical PCA aims to find a subspace $S \subset \mathbb{R}^p$ of dimension $k$ ($k < p$) that best fits these data points. The problem can be expressed mathematically as the following optimisation problem:

$$\min_{u, U, \{x_i\}} \sum_{i=1}^n \|y_i - u - Ux_i\|^2 \quad \text{s.t.} \quad U^\top U = I_k \quad \text{and} \quad \sum_{i=1}^n x_i = 0,$$

where $u$ is a point which representing the centre of the subspace, $U$ is a $p \times k$ matrix whose columns are the basis of the subspace and $x_i \in \mathbb{R}^k$ is the vector of the new coordinates of $y_i$ in the subspace. The optimal solution of the classical PCA (Vidal, Ma & Sastry 2016, Chapter 2.1.2) can be obtained as

$$\hat{u} = \frac{1}{n} \sum_{i=1}^n y_i \quad \text{and} \quad \hat{x}_i = \hat{U}^\top (y_i - \hat{u}),$$

where $\hat{U}$ is a $p \times k$ matrix whose columns are the eigenvectors corresponding to the largest $k$ eigenvalues of the sample covariance matrix

$$\hat{\Sigma}_n = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{u}) (y_i - \hat{u})^\top.$$

Then, $\hat{U}\hat{x}_i$ is the low-rank approximation of $y_i$, assuming without loss of generality that $E(y) = 0$ and $\hat{u} = 0$. However, it is well known that if the data contain extreme values or have a heavy-tailed distribution, the above optimisation is unreliable, and the solution $\hat{U}\hat{x}_i$ is not a good low-rank approximation of the original high-dimensional data. For example, if the data points come from a heavy-tailed distribution with no finite second moment, the covariance matrix $\hat{\Sigma}_n$ will be extremely unreliable and invalid for making inferences about the population covariance matrix.

To address this issue, we propose a new PCA method for obtaining a good approximation of $y$ that is robust to outliers and heavy-tailed distributions in this paper. The idea is to find a transformation that is robust to the heavy-tailed distribution or extreme values of the original data, and then perform classical PCA on the transformed data instead. The details of the method are described below.

Let $z = (z_1, \ldots, z_p)^\top$ be the transformed data of $y$, where the transformation is

$$z_j = \exp\{iy_j\}, \quad j \in \{1, \ldots, p\},$$
and $i$ is the imaginary unit. The reasons for this transformation come from the special properties of $z$. First, because it has the form of the characteristic function of $y$, $z$ has finite second moments and contains most of the information in $y$, which solves the problem that $y$ comes from heavy-tailed distributions, especially those without second moments. Second, $z_j$ has an absolute value of 1 for each $j \in \{1, \ldots, p\}$, indicating that its variance is bounded. This property reduces the effect of the possible outliers or extremely different variances on the dimension-reduction result. Third, because of the non-linear property of the transformation, it is able to reveal non-linear relationships between the components in $y$, unlike the classical PCA, which can only reveal linear relationships.

In more detail, we provide two instances to illustrate this motivation.

**Instance 1**: Consider the $p$-dimensional random vector $y$ under consideration, which follows the Cauchy distribution. In this case, no moments of this random vector $y$ exist, not even the first moment. Obviously, it is nonsense to perform PCA directly on $y$.

The proposed transformation has the advantage of well defining all moments of the transformed data $z$, because $\mathbb{E}(z_d^j) = \mathbb{E}(\exp(i \pi y_j))$ is the exponential function of $y_j$ valued at $d$. Furthermore, it is possible to perform a PCA on $z$ since its second moment exists.

**Instance 2**: Consider outliers from the normal distribution $N(100, 1)$, while the regular data observations follow $N(0, 1)$. The poor quality of the data observations affects statistical inference, such as the sample mean estimator.

Because the transformed data are bounded by 1, the distance between the outliers and the regular observations is reduced after the transformation. This idea is consistent with regularisation in statistics.

While $z$ has desirable properties, it contains complex elements that complicate the situation. On the other hand, according to Euler’s formula, $z_j$ can be written as:

$$z_j = \exp(i y_j) = \cos y_j + i \sin y_j, \quad j \in \{1, \ldots, p\}.$$ 

Then, if we define $r = (\cos y_1, \ldots, \cos y_p, \sin y_1, \ldots, \sin y_p)^\top$, we have $z$ as a linear transform of $r$:

$$z = \begin{pmatrix}
1 & 0 & \cdots & 0 & i & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 & 0 & i & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 & 0 & 0 & \cdots & i
\end{pmatrix}
\begin{pmatrix}
\cos y_1 \\
\vdots \\
\cos y_p \\
\sin y_1 \\
\vdots \\
\sin y_p
\end{pmatrix} := (I_p \quad i I_p) r.$$

Assume there also exists a low-rank subspace that best fits the data points $z_1, \ldots, z_n$ generated from $z$. Then, to find a good low-rank approximation of $z_i (i \in \{1, \ldots, n\})$, we need only perform classical PCA on $r_i (i \in \{1, \ldots, n\})$, which are real-valued random vectors.

Suppose $\Sigma_r$ is the covariance matrix of $r$, and $\beta_1, \ldots, \beta_k$ are the orthonormal eigenvectors corresponding to the $k$ largest eigenvalues $\lambda_1 > \cdots > \lambda_k$ of $\Sigma_r$. By the classical PCA method, $r$ is approximated by
\[ \tilde{r} = E(r) + \sum_{j=1}^{k} \beta_j \beta_j^\top (r - E(r)) \]

\[ = E(r) + \left( \sum_{j=1}^{k} (r - E(r))^\top \beta_j \beta_j^{(\cos)} \right) + \sum_{j=1}^{k} (r - E(r))^\top \beta_j \beta_j^{(\sin)} \right)^\top, \]

where \( \beta_j^{(\cos)} = (\beta_{1,j}, \ldots, \beta_{p,j})^\top \) and \( \beta_j^{(\sin)} = (\beta_{p+1,j}, \ldots, \beta_{2p,j})^\top \), which are the first half and the second half of \( \beta_j \) respectively.

Therefore, the low-dimensional approximation of \( z \) is

\[ \tilde{z} = \left( I_p \ iI_p \right) \tilde{r} = \left( I_p \ iI_p \right) E(r) + \left( I_p \ iI_p \right) \left( \sum_{j=1}^{k} (r - E(r))^\top \beta_j \beta_j^{(\cos)} \right) + i \sum_{j=1}^{k} (r - E(r))^\top \beta_j \beta_j^{(\sin)} \right)^\top, \]

\[ = E(z) + \sum_{j=1}^{k} \beta_j^{(\cos)} \beta_j^\top (r - E(r)) + i \sum_{j=1}^{k} \beta_j^{(\sin)} \beta_j^\top (r - E(r)). \]

With data points \( r_1, \ldots, r_n \), it is straightforward to estimate \( E(r), E(z) \) and \( \Sigma_r \) by

\[ \tilde{r} = \frac{1}{n} \sum_{i=1}^{n} r_i, \quad \tilde{z} = \left( I_p \ iI_p \right) \tilde{r}, \quad \text{and} \quad \tilde{\Sigma}_{r,n} = \frac{1}{n} \sum_{i=1}^{n} (r_i - \tilde{r}) (r_i - \tilde{r})^\top, \]

respectively. We also estimate \( \beta_j (j \in \{1, \ldots, k\}) \) by the eigenvectors \( \hat{\beta}_j (j \in \{1, \ldots, k\}) \) of \( \tilde{\Sigma}_{r,n} \).

The method of estimating \( k \) can be different and we use the accumulative variance as the criterion in the empirical analysis for simplicity. Other appropriate criteria may be used for different purposes.

Finally, to recover the original data, we just need to transform back from \( \tilde{z}_i (i \in \{1, \ldots, n\}) \).

The approximation of \( y_i (i \in \{1, \ldots, n\}) \) is:

\[ \tilde{y}_i = \frac{1}{1} \ln(\tilde{z}_i) + 2h_i \pi I \]

\[ = \frac{1}{1} \ln \left( \tilde{z} + \sum_{j=1}^{k} \hat{\beta}_j^{(\cos)} \hat{\beta}_j^\top (r_i - \tilde{r}) + i \sum_{j=1}^{k} \hat{\beta}_j^{(\sin)} \hat{\beta}_j^\top (r_i - \tilde{r}) \right) + 2h_i \pi I, \quad (1) \]

where \( \ln(a) = (\ln(a_1), \ldots, \ln(a_n))^\top \) for any \( n \)-dimensional vector \( a \), and \( h_i (i \in \{1, \ldots, n\}) \) is an integer which needs to be estimated in practice.

**Remark 1.** The computational algorithm is summarised in Algorithm 1. Note that \( \tilde{z}_i (i \in \{1, \ldots, n\}) \) consists of complex numbers. The complex logarithm can have an infinite number of values due to the periodicity of the complex exponential function. According to Euler's formula, these values differ by multiples of \( 2h_i \pi \). Therefore, in (1), we need to find the \( h_i \) to ensure that \( \tilde{y}_i \) is a good approximation of \( y_i \). Therefore, in practice, we estimate \( h_i \) for each data point \( y_i \) by
arg min_{h_i \in \mathbb{Z}} \left\| y_i - \left( \frac{1}{4} \ln \tilde{z}_i + 2h_i \pi I \right) \right\|_2^2 (i \in \{1, \ldots, n\}).

then

\hat{h}_i = \left[ \frac{1}{2\pi} \left( \sum_{j=1}^{p} \left( y_{ij} - \frac{1}{4} \ln \tilde{z}_{ij} \right) \right) + 0.5 \right],

where \lfloor x \rfloor retains the integer part of x.

Remark 2. Our proposed method can be viewed as a special kind of nonlinear and kernel PCA (Vidal, Ma & Sastry 2016, Chapter 4.1). The nonlinear transformation is \( \phi(\cdot): \mathbb{R}^p \rightarrow \mathbb{R}^{2p} \), where \( \phi(y) = (\cos y_1, \ldots, \cos y_p, \sin y_1, \ldots, \sin y_p)^T \) and the kernel function is

\[ \kappa(y_j, y_j) = \phi(y_j)^\top \phi(y_j) = \sum_{m=1}^{p} \cos y_{mi} \cos y_{mj} + \sum_{m=1}^{p} \sin y_{mi} \sin y_{mj}. \]

This shows that our method is capable of exploring the non-linear relationship between the original data. In addition, we can compute the principal components with the kernel function described above according to the Kernel PCA algorithm (see e.g. Algorithm 4.1 in Vidal, Ma & Sastry 2016), which is particularly useful when the dimension \( p \) is too large to compute the covariance matrix of the transformed data.

---

Algorithm 1: Robust PCA for High-dimensional Data

Input: Data \( \tilde{Y} = (y_1, \ldots, y_n) \in \mathbb{R}^{p \times n}; \) Desired rank \( \leq p. \)

Output: Low-dimensional representation of \( \tilde{Y}. \)

Transformation Step:
1. Compute \( \tilde{R} = (r_1, \ldots, r_n) \in \mathbb{R}^{(2p) \times n}, \) where
   \[ r_i = (\cos y_{1i}, \ldots, \cos y_{pi}, \sin y_{1i}, \ldots, \sin y_{pi})^T; \]
2. Compute the sample mean \( \tilde{r} = \sum_{i=1}^{n} r_i / n; \)
3. Compute the sample variance-covariance matrix \( \tilde{\Sigma}_{r,n} = \sum_{i=1}^{n} (r_i - \tilde{r})(r_i - \tilde{r})^\top / n; \)
4. Conduct eigen decomposition on \( \tilde{\Sigma}_{r,n} \) and get \( \tilde{\beta}_1, \ldots, \tilde{\beta}_\xi \), the eigenvectors corresponding to the largest \( \xi \) eigenvalues of \( \tilde{\Sigma}_{r,n}; \)

Inverse Transformation Step:
5. Compute \( \tilde{\beta}_j^{(\text{cos})} = (\beta_{1j}, \ldots, \beta_{pj})^\top \) and \( \tilde{\beta}_j^{(\text{sin})} = (\beta_{p+1j}, \ldots, \beta_{pj})^\top; \)
6. Compute \( \tilde{z} = (I_p - i I_p) \tilde{r}; \)
7. Compute \( \tilde{z}_i = \tilde{z} + \sum_{j=1}^{\xi} \tilde{\beta}_j^{(\text{cos})} \beta_j^{(\text{cos})} (r_i - \tilde{r}) + i \sum_{j=1}^{\xi} \tilde{\beta}_j^{(\text{sin})} \beta_j^{(\text{sin})} (r_i - \tilde{r}); \)
8. Compute \( \hat{h}_i = \arg \min_{h_i \in \mathbb{Z}} \left\| y_i - (\ln \tilde{z}_i / i + 2h_i \pi I) \right\|_2^2, \ i \in \{1, \ldots, n\}; \)
9. Compute \( \tilde{y}_i = \ln(\tilde{z}_i / i + 2h_i \pi I), \ i \in \{1, \ldots, n\}. \)

3. Statistical properties

In this section, we study the statistical properties of the robust PCA. First, we investigate the inequalities of the optimal and empirical reconstruction errors and provide a general upper bound for the excess error of the PCA methods. It is shown that the upper bound does
not hold for the classical PCA when the original data are heavy tailed without finite variance. The robust PCA, on the other hand, can achieve a relatively smaller excess error because the proposed transformation ensures bounded variances for the transformed variables. We also discuss the conditions under which the empirical reconstruction error of the sample eigenvectors approaches the optimal reconstruction error. Second, we study the behaviour of the eigenvalues of both the original data and the transformed data. By assuming a spike covariance structure on the original data, we find that the transformation in our proposed method still preserves the spiked structure, which ensures that the proposed method can effectively extract information from noises. Furthermore, we show that when the data are extremely heavy tailed, the proposed robust PCA yields more stable empirical eigenvalues.

3.1. The upper bound of the excess error

In this section, we give the upper bound for the aforementioned excess error as well as the order of the difference between the empirical reconstruction error of the sample eigenvectors and the optimal reconstruction error in Theorem 1. Theorem 1 holds in general for both the classical PCA and the newly proposed robust PCA. We demonstrate with this theorem that for extremely heavy-tailed data, the proposed method is able to achieve a small excess error and an empirical reconstruction error that is close to the optimal error. However, the classical PCA may fail under the same conditions.

Let us first introduce some notation and definitions to illustrate the results. Suppose a random vector \( y = (y_1, \ldots, y_p)^\top \in \mathbb{R}^p \) has mean 0 and covariance matrix \( \Sigma \). \( y_1, \ldots, y_n \) are \( n \) independent samples of \( y \) and the corresponding sample covariance matrix is \( \hat{\Sigma} \). Let \( \beta_1, \ldots, \beta_p \) be the orthonormal eigenvectors corresponding to the eigenvalues of \( \Sigma \) in descending order, and \( \hat{\beta}_1, \ldots, \hat{\beta}_p \) be those of \( \hat{\Sigma} \). Denote \( B_k = (\beta_1, \ldots, \beta_k) \) and \( \hat{B}_k = (\hat{\beta}_1, \ldots, \hat{\beta}_k) \) (\( k < p \) is fixed). With \( B_k \), the basis of the optimal low-dimensional subspace, we have

\[
y = \sum_{i=1}^{k} \beta_i \beta_i^\top y + \sum_{i=k+1}^{p} \beta_i \beta_i^\top y = B_k B_k^\top y + u(B_k)
\]

and

\[
y_j = \sum_{i=1}^{k} \beta_i \beta_i^\top y_j + \sum_{i=k+1}^{p} \beta_i \beta_i^\top y_j = B_k B_k^\top y_j + u_j(B_k), \quad j \in \{1, \ldots, n\}.
\]

Similarly, with \( \hat{B}_k \), the empirical equivalent of \( B_k \), we have

\[
y = \sum_{i=1}^{k} \hat{\beta}_i \hat{\beta}_i^\top y + \sum_{i=k+1}^{p} \hat{\beta}_i \hat{\beta}_i^\top y = \hat{B}_k \hat{B}_k^\top y + u(\hat{B}_k)
\]

and

\[
y_j = \sum_{i=1}^{k} \hat{\beta}_i \hat{\beta}_i^\top y_j + \sum_{i=k+1}^{p} \hat{\beta}_i \hat{\beta}_i^\top y_j = \hat{B}_k \hat{B}_k^\top y_j + u_j(\hat{B}_k), \quad j \in \{1, \ldots, n\}.
\]

So the (true) reconstruction error with \( B_k \) and \( \hat{B}_k \) can be written as
\[ R(B_k) = \mathbb{E} \left( \left( y - \sum_{i=1}^{k} \beta_i \beta_i^\top y \right) \left( y - \sum_{i=1}^{k} \beta_i \beta_i^\top y \right) \right) = \mathbb{E} \left( u(B_k)^\top u(B_k) \right) \]

and

\[ R(\hat{B}_k) = \mathbb{E} \left( \left( y - \sum_{i=1}^{k} \hat{\beta}_i \hat{\beta}_i^\top y \right) \left( y - \sum_{i=1}^{k} \hat{\beta}_i \hat{\beta}_i^\top y \right) \right) = \mathbb{E} \left( u(\hat{B}_k)^\top u(\hat{B}_k) \right). \]

We call \( R(B_k) \) the optimal error. The difference \( R(\hat{B}_k) - R(B_k) \) is the so-called excess error of \( \hat{B}_k \) with respect to its optimal \( B_k \). Furthermore, the corresponding empirical reconstruction errors are

\[ R_n(B_k) = \frac{1}{n} \sum_{j=1}^{n} \left( \left( y_j - \sum_{i=1}^{k} \beta_i \beta_i^\top y_j \right) \left( y_j - \sum_{i=1}^{k} \beta_i \beta_i^\top y_j \right) \right) \]

\[ = \frac{1}{n} \sum_{j=1}^{n} \left( u_j(B_k)^\top u_j(B_k) \right) \]

and

\[ R_n(\hat{B}_k) = \frac{1}{n} \sum_{j=1}^{n} \left( \left( y_j - \sum_{i=1}^{k} \hat{\beta}_i \hat{\beta}_i^\top y_j \right) \left( y_j - \sum_{i=1}^{k} \hat{\beta}_i \hat{\beta}_i^\top y_j \right) \right) \]

\[ = \frac{1}{n} \sum_{j=1}^{n} \left( u_j(\hat{B}_k)^\top u_j(\hat{B}_k) \right). \]

Define

\[ d_k := \frac{\mathbb{E} \left( u_j(B_k)^\top u_j(B_k) \right)}{\mathbb{E} \left( y_j^\top y_j \right)}, \quad k \in \{1, 2, \ldots, \min(n, p)\}. \]

We have the following results for the true and empirical reconstruction errors:

**Theorem 1.** Suppose that \( k \) is the number of spiked eigenvalues for the population covariance matrix, then we have

\[ \Pr \left( \left| (R(\hat{B}_k) - R_n(\hat{B}_k)) \right| \leq d_k \left( \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) \right)^{1/2} \frac{\sqrt{c \xi}}{2n} \right) \geq 1 - 2e^{-\xi}, \quad (2) \]

and

\[ \Pr \left( 0 \leq \left( R(\hat{B}_k) - R(\hat{B}_k) \right) \leq 2d_k \left( \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) \right)^{1/2} \frac{\sqrt{c \xi}}{2n} \right) \geq 1 - 4e^{-\xi}, \quad (3) \]

where \( c \) is a constant.

In particular, as \( \xi \to \infty \), if \( d_k \left( \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) \right)^{1/2} \frac{\sqrt{c \xi}}{2n} \to 0 \), then we have

\[ \left| (R(B_k) - R_n(\hat{B}_k)) \right| = O_p \left( d_k \left( \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) \right)^{1/2} \frac{\sqrt{c \xi}}{2n} \right) = o_p(1). \quad (4) \]
Remark 3. We interpret the three differences in Theorem 1 as follows:

1. Inequality (2) is the difference between the estimated low-dimensional subspace’s true reconstruction error and empirical reconstruction error. It can be understood as a confidence interval on the true reconstruction error calculated from purely empirical data.
2. Inequality (3) is the so-called excess error, which measures the distance between the estimated and optimal subspace in terms of true reconstruction error.
3. Finally, the difference in (4) indicates the distance between the purely empirical reconstruction error and the optimal error.

We have two requirements in order to get the upper bounds mentioned above close to zero:

1. \( \xi \) is large enough to assure \( \exp(-\xi) \to 0 \) and the probability close to 1;
2. \( d_k \left( \sum_{i=1}^{p} E \left( y_{ij}^2 \right) \right) \sqrt{c \xi / 2n} \to 0 \) with \( k \) being fixed and \( p, n \to \infty \).

We can see from the above requirements that whether \( E(y_{ij}^2) \) is finite or not matters. If \( E(y_{ij}^2) \) is infinite, it is hard to satisfy the second requirement in high-dimensional settings. If \( E(y_{ij}^2) \) is finite, then \( \left( \sum_{i=1}^{p} E \left( y_{ij}^2 \right) \right) \sqrt{c \xi / 2n} = O \left( p \sqrt{\xi / n} \right) \) as \( c \) is a constant. If \( \xi \) is a large constant, for example \( \xi = 10 \), it is large enough to satisfy the first requirement, and the second requirement becomes \( d_k p / \sqrt{n} \to 0 \) where \( k \) is fixed and \( p, n \to \infty \). Then, to satisfy the second requirement, we need \( d_k = o(\sqrt{n} / p) \). For example, if we assume \( p / \sqrt{n} = O(1) \), which is common in high-dimensional statistics, then we only need \( d_k = o(1) \). It is worth noting that if we assume a spiked covariance structure for the random vector, \( d_k \) should meet this requirement well with a fixed \( k \).

We have mentioned that Theorem 1 holds for both the classical PCA and the proposed robust PCA. Similar statements can be found in Blanchard, Bousquet & Zwald (2007), where they provide results that are valid for both classical PCA and Kernel PCA. To be more specific, in Theorem 1, the random variable \( y_{ij} \) refers to the original random vector for classical PCA while to the transformed vector for the robust PCA. If the original data are heavy tailed with infinite variances, the classical PCA is difficult to meet the above requirements and may result in infinite excess error. The transformed data, on the other hand, have finite variances even when the original data do not. As a result, as long as the transformed data meet the requirements for finite \( E(y_{ij}^2) \)’s, such as having a spiked covariance structure, the robust PCA will have empirical reconstruction error close to optimal error with order \( o_p(1) \). When the original random vector has finite covariance matrix, so does the transformed vector, and there is not much difference between the classical PCA and the robust PCA in view of the order of the differences in Theorem 1. In summary, the proposed method is of great importance especially when the original data have an infinite covariance matrix.

The remainder of this section provides a proof of Theorem 1. In the following section (Section 3.2), we will discuss more about the spiked covariance structure.

Proof 1. We employ the following lemma of the concentration inequality to complete our proof.
Lemma 1. (McDiarmid 1989) Let $X_1, \ldots, X_n$ be $n$ independent random variables taking values in $\mathcal{X}$ and let $Z = f(X_1, \ldots, X_n)$ where $f$ is such that

$$
\sup_{x_1, \ldots, x_n, x'_1 \in \mathcal{X}} |f(x_1, \ldots, x_n) - f(x_1, \ldots, x'_1, \ldots, x_n)| \leq c_i, \ \forall \ 1 \leq i \leq n,
$$

then

$$
\Pr \left[ Z - \mathbb{E}(Z) \geq \xi \right] \leq e^{-2\xi^2/(c_1^2 + \cdots + c_n^2)} \quad \text{and} \quad \Pr \left[ \mathbb{E}(Z) - Z \geq \xi \right] \leq e^{-2\xi^2/(c_1^2 + \cdots + c_n^2)}.
$$

Let $\mathcal{X}$ be the set of all independent samples of $y$ and

$$
Z = f(y_1, \ldots, y_n) = R_n(\hat{B}_k) - R_n(\hat{B}_0)
$$

$$
= \mathbb{E} \left( u(\hat{B}_k)^\top u(\hat{B}_k) \right) - \frac{1}{n} \sum_{j=1}^{n} \left( u_j(\hat{B}_k)^\top u_j(\hat{B}_k) \right).
$$

Then we have $\forall 1 \leq i \leq n,$

$$
\sup_{y_1, \ldots, y_n, y'_1 \in \mathcal{X}} |f(y_1, \ldots, y_n) - f(y_1, \ldots, y'_1, \ldots, y_n)|
$$

$$
= \sup_{y_1, \ldots, y_n, y'_1 \in \mathcal{X}} \left| \frac{1}{n} \left( u_i(\hat{B}_k)^\top u_i(\hat{B}_k) - u_i(\hat{B}_k)^\top u_i(\hat{B}_k) \right) \right|.
$$

To apply Lemma 1, we only need to find the upper bound of the above quantity.

The following evaluation

$$
\mathbb{E} \left( u_j(\hat{B}_k)^\top u_j(\hat{B}_k) \right) = \sum_{i=1}^{p} \mathbb{E} \left( u_{ij}^2 \right) = O \left( d_k \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) \right),
$$

which indicates

$$
u_j (\hat{B}_k)^\top u_j (\hat{B}_k) = O_p \left( d_k \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) \right). \quad (5)
$$

Then from (5), we have for any $i$ and $j,$

$$
\left| \frac{1}{n} \left( u_i(\hat{B}_k)^\top u_i(\hat{B}_k) - u_j(\hat{B}_k)^\top u_j(\hat{B}_k) \right) \right| = O_p \left( \frac{d_k}{n} \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) \right).
$$

Let $c_j = c d_k \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) / n (j \in \{1, \ldots, n\}),$ where $c$ is a constant that may be different from line to line. According to Lemma 1, we have

$$
\Pr \left( |Z - \mathbb{E}(Z)| \leq t \right) \geq 1 - 2e^{2t^2 \left( \sum_{j=1}^{n} \left( c_j d_k \sum_{i=1}^{p} \mathbb{E} \left( y_{ij}^2 \right) \right)^2 \right)}
$$

$$
= 1 - 2e^{2t^2 \left( \sum_{j=1}^{n} \mathbb{E} \left( y_{ij}^2 \right) \right)^2}.
$$
Let $\xi = 2t^2 \left( \frac{c d_k^2 \left( \sum_{i=1}^{p} E \left( y_{ij}^2 \right) \right)^2}{n} \right)$, which leads to $t = d_k \left( \sum_{i=1}^{p} E \left( y_{ij}^2 \right) \right) \sqrt{c \xi / 2n}$. Then we can rewrite the above inequality as

$$\Pr \left( \left| (R(\tilde{B}_k) - R_n(\tilde{B}_k)) \right| \leq d_k \left( \sum_{i=1}^{p} E \left( y_{ij}^2 \right) \right) \sqrt{\frac{c \xi}{2n}} \right) \geq 1 - 2e^{-\xi},$$

which is the first part of Theorem 1.

For the second part of Theorem 1, we first have

$$R(\tilde{B}_k) - R_n(\tilde{B}_k) \geq 0 \quad \text{and} \quad R_n(\tilde{B}_k) - R_n(B_k) \leq 0$$

due to that $B_k$ minimised the true reconstruction error and $\tilde{B}_k$ minimised the empirical reconstruction error according to PCA. Hence we have

$$0 \leq R(\tilde{B}_k) - R_n(B_k) \quad \text{(according to the first inequality in (7))}$$

$$= (R(\tilde{B}_k) - R_n(\tilde{B}_k)) - (R(B_k) - R_n(B_k)) + \left( R_n(\tilde{B}_k) - R_n(B_k) \right)$$

$$\leq (R(\tilde{B}_k) - R_n(\tilde{B}_k)) - (R(B_k) - R_n(B_k))$$

$$\quad \text{(according to the second inequality in (7))}$$

$$\leq \left| (R(\tilde{B}_k) - R_n(B_k)) \right| + \left| (R(B_k) - R_n(B_k)) \right|.$$ 

The first term is controlled by (6). Following the same procedure, we also have

$$\Pr \left( |R(B_k) - R_n(B_k)| \leq d_k \left( \sum_{i=1}^{p} E \left( y_{ij}^2 \right) \right) \sqrt{\frac{c \xi}{2n}} \right) \geq 1 - 2e^{-\xi}.$$ 

As a result, we have

$$\Pr \left( 0 \leq R(\tilde{B}_k) - R_n(B_k) \leq 2d_k \left( \sum_{i=1}^{p} E \left( y_{ij}^2 \right) \right) \sqrt{\frac{c \xi}{2n}} \right) \geq 1 - 4e^{-\xi}.$$ 

Inequality (6) and (8) are the final results in Theorem 1.

### 3.2. Behaviour of the leading eigenvalues under spiked covariance structure

As discussed before, the empirical reconstruction error’s distance to the optimal error is largely dependent on the covariance structure of data. In this part, we make assumptions about the covariance structure of the data and use simulation to show how leading eigenvalues behave under different assumptions. Following the assumptions in Cai, Han & Pan (2017) and Wang & Fan (2017), we first assume that the original data are normal distributed and have a spiked covariance structure. We are interested in how the transformation in our method affects the spike covariance structure under this assumption. Next, we assume that the original data do not have a finite covariance while the transformed data have a spiked covariance structure. We show that under this assumption, the estimations of the classical PCA vary greatly, while the robust PCA produces more stable results.

**Remark 4.** A wide range of literature has made effort to understand the behaviour of empirical eigenvalues in high-dimensional settings. Yata & Aoshima (2012), Shen et al.
L. HE, Y. YANG, AND B. ZHANG (2016b) and others focused on the high-dimension, low-sample-size (HDLSS) scenario, in which the dimension $p$ can go to infinity with a fixed sample size $n$. Meanwhile, Johnstone (2001), Lam, Yao & Bathia (2011), Lee, Zou & Wright (2014), Wang & Fan (2017), Cai, Han & Pan (2017) studied cases in which both the sample size $n$ and the dimension $p$ can go to infinity. Shen, Shen & Marron (2016a) nicely described how the relationships between dimension, sample size and spike size effect PCA consistency. Particularly, we are interested in the spiked covariance model, of which the distribution of the empirical eigenvalues has been explored in Cai, Han & Pan (2017), Wang & Fan (2017), and others. Typically, the spiked covariance model assumes that there are several eigenvalues larger than the rest. The larger eigenvalues are called spiked eigenvalues, whereas the remaining eigenvalues are called non-spiked eigenvalues. Specifically, Cai, Han & Pan (2017) and Wang & Fan (2017) assume that the population covariance matrix has $k (k \to 0, p$ is the number of dimension) well separated spiked eigenvalues and the non-spiked eigenvalues are all bounded but otherwise arbitrary.

In Wang & Fan (2017), the asymptotic normality of the spiked empirical eigenvalues was proved under a spiked covariance model (see Wang & Fan 2017, Assumptions 2.1–2.3 and Theorem 3.1). Assume the population covariance model has $k$ spiked eigenvalues $\{\hat{\lambda}_j\}_{j=1}^k$, and the corresponding empirical eigenvalues are $\{\hat{\lambda}_j\}_{j=1}^k$. The theorem shows that $\hat{\lambda}_j$ asymptotically unbiased, it requires $c_j \to 0$ for $j \leq k$. If we assume that the original data meet the above assumptions, the classical PCA should perform well in estimating leading eigenvalues of the original data. What about our proposed robust PCA? Will the transformation still retain a spiked transformed covariance matrix, ensuring that the principle component results are valid? To demonstrate the effect, we run a simulation with normal-distributed data. We intend to show how the population covariance structure changes following the proposed transformation, as well as how accurate the empirical spiked eigenvalues are for both the original and transformed data with different sample sizes.

**Remark 5.** The following is how the data for the simulations are generated. We simulate $P \times 1 \ (P = 100 \text{ in this case})$ vector $y_n \ (n \in \{1, \ldots, N\})$ by

$$y_n = \sum_{i=1}^{3} \alpha_i b_i k_{i,n} + \varepsilon_n, \quad (9)$$

where $b_i \ (i \in \{1, 2, 3\})$ are $P \times 1$ vectors generated by QR decomposition, $k_{i,n} \ (i \in \{1, 2, 3\}, n \in \{1, \ldots, N\})$ are generated independently from standard normal $N(0, 1)$, and $\varepsilon_n \ (n \in \{1, \ldots, N\})$ are $P \times 1$ error vectors with elements generated independently from $N(0, 1)$. Furthermore, $(\alpha_1, \alpha_2, \alpha_3) = (7, 5, 3)$.

The QR decomposition is detailed below. We intend to generate $b_i \ (i \in \{1, 2, 3\})$ such that $b_1, b_2, \text{ and } b_3$ are orthogonal to each other. First, we generated a $P \times P$ matrix $A$ with elements drawn at random from $N(0, 1)$. Then decompose $A$ into a product $A = QR$, where $Q$ is an orthogonal matrix and $R$ is an upper triangular matrix. We use the first three columns of the orthogonal matrix $Q$ as the values of vectors $b_i \ (i \in \{1, 2, 3\})$ in the simulation studies.

Let $\hat{\lambda}_j^{\text{pca}}$ and $\hat{\lambda}_j^{\text{pca}}$ represent the two sets of population eigenvalues for original and
Figure 1. Approximated population eigenvalues.

Table 1. Mean and SD of $\hat{\lambda}_{\text{cpca}}^j/\lambda_{\text{cpca}}^1 - 1$ and $\hat{\lambda}_{\text{rpca}}^1/\lambda_{\text{rpca}}^1 - 1$.

|      | 50     | 100    | 500    | 1000   | 5000   |
|------|--------|--------|--------|--------|--------|
|      | Bias   | SD     | Bias   | SD     | Bias   | SD     | Bias   | SD     | Bias   | SD     |
| Normal | cpca   | 0.051  | 0.195  | 0.030  | 0.137  | 0.005  | 0.062  | 0.004  | 0.044  | 0.000  | 0.020  |
|       | rpca   | 0.180  | 0.127  | 0.094  | 0.097  | 0.023  | 0.070  | 0.011  | 0.064  | 0.004  | 0.057  |

The average and standard variation of the difference between the estimated eigenvalue and the population eigenvalue are shown in Table 1. Smaller mean and standard deviation of the difference are represented by the bold numbers.

The transformed data respectively. Figure 1 shows $\lambda_{\text{cpca}}^j$ s and $\lambda_{\text{rpca}}^j$ s (approximated) for the simulated example. The $\lambda_{\text{cpca}}^j$ s and $\lambda_{\text{rpca}}^j$ s are approximated by simulating the data for 50 times with $n = 100,000$. On the left and right figures, we show $\lambda_{\text{cpca}}^j$ s and $\lambda_{\text{rpca}}^j$ s respectively. The original and the transformed data both have spiked covariance structures, although the number of leading eigenvalues may be different. In this example, the transformation will retain a spiked structure if the original data contain a spiked covariance structure. As a result, both the classical PCA and the proposed robust PCA should perform well on these data.

Next we compare the estimations of the largest eigenvalue for the original and the transformed data. It is worth noting that $\lambda_{\text{rpca}}^1$ is smaller than $\lambda_{\text{cpca}}^1$. According to the previously specified rate $c_j = p/(n \lambda_j)$, the transformation may affect the convergence rate of the eigenvalues. The effect is empirically presented in Table 1. Table 1 shows that in this example, $\lambda_{\text{rpca}}^1$ is more biased than $\lambda_{\text{cpca}}^1$ with the same sample size. However, as $n$ increases, the

Table 2. Variation of the largest empirical eigenvalues.

|      | 50     | 100    | 500    | 1000   | 5000   |
|------|--------|--------|--------|--------|--------|
|      | Bias   | SD     | Bias   | SD     | Bias   | SD     | Bias   | SD     | Bias   | SD     |
| t(2) | cpca   | 4.796  | 16.082 | 17.431 | 4.752  | 5.488  |
|      | rpca   | 0.105  | 0.093  | 0.074  | 0.070  | 0.068  |

The numbers in Table 2 show how the largest empirical eigenvalues vary from their averages. The bold values represent smaller variation.
bias of $\hat{\lambda}_1^{rpca}$ tends to 0, as desired. Moreover, in high-dimensional cases ($n \in \{50, 100\} \leq p$), $\hat{\lambda}_1^{rpca}$ varies less than $\hat{\lambda}_1^{cpca}$.

We have showed that both the classical PCA and the robust PCA work well in the normal case. However, if the original data do not have finite covariance, the classical PCA will fail, but the robust PCA should still work. To demonstrate the phenomenon, we employ a simulation. We simulate the same data as before, but with $k_{i,n}$ generated via a t-distribution with degree of freedom 2. It should be noted that, the population eigenvalue of the transformed data $\hat{\lambda}_1^{rpca}$ is well defined in this situation, but $\hat{\lambda}_1^{cpca}$ does not exist. Table 2 displays $sd(\hat{\lambda}_1^{cpca})/mean(\hat{\lambda}_1^{cpca})$ and $sd(\hat{\lambda}_1^{rpca})/mean(\hat{\lambda}_1^{rpca})$, which represent the variation of the largest empirical eigenvalues relative to their averages. It shows that $\hat{\lambda}_1^{cpca}$ changes greatly, whereas $\hat{\lambda}_1^{rpca}$ is significantly more stable. In this extremely heavy-tailed case, the aforementioned asymptotic normal result for leading eigenvalues is valid on the transformed data but not on the original data. As a result, we cannot trust the results of classical PCA in this case. It provides a strong evidence that the classical PCA is invalid with extremely heavy-tailed data, and that our proposed robust PCA is necessary in such cases.

4. Reconstruction performance under different situations

In this section, we illustrate the advantage of our proposed method (rpca) against the classic PCA (cpca) in recovering original data under several scenarios. We evaluate both the in- and out-of-sample performance. For the in-sample part, we extract eigenvectors and reconstruct the same simulated data. To assess performance, we utilise the mean squared error (MSE) of the approximation:

$$
\text{MSE} = \frac{1}{NP} \sum_{n=1}^{N} ||\hat{y}_n - y_n||_2^2,
$$

where the $\hat{y}_n$ is the approximation from rpca or cpca (both recovering at least 80% of the total variance) and $y_n$ is the original data. $N$ is the sample size and $P$ is the dimension of $y_n$. The number of components $k$ is determined by the threshold of the cumulative sum of the eigenvalues, that is,

$$
\hat{k} = \min \left\{ k : \frac{\sum_{i=1}^{k} \hat{\lambda}_i}{\sum_{i=1}^{P} \hat{\lambda}_i} \geq \gamma \right\},
$$

where $\hat{\lambda}_i$ are the estimated eigenvalues and $\gamma$ is the threshold. We set $\gamma = 80\%$ in the following simulations. We repeat 1000 times for each case and report the average in-sample MSE.

For the out-of-sample part, we use 10-fold cross-validation to evaluate the performance of out-of-sample reconstruction. We divide the entire sample into 10-fold for each case. Then, one of the folds is used as the test set, while the remaining folds are utilised as the training set. We extract the eigenvectors from the training set using the rpca and cpca respectively, and those eigenvectors are used to recover the test data. The procedure is repeated until all the 10-fold have been used as a test set. The test MSE is recorded for each fold, and the cross-validation MSE is the average of the 10 test MSEs. For each case, we repeat the preceding steps 100 times and report the average cross-validation MSE (out-of-sample MSE).
Example 1 shows rpca’s remarkable ability to handle data with heterogeneity in variances. Example 2 demonstrates that rpca outperforms cpca when approximating data containing outliers. In Example 3, we simulate data from three different heavy-tailed distributions, as well as the normal distribution as a benchmark, and find the rpca recovers the data more accurately than the cpca. Let us now discuss the simulations in details.

4.1. Example 1: heterogeneity in variances

The heterogeneity in variances is common in real-world data, and variables with very significant variances tend to dominate the results of classic PCA (Jolliffe 2002). As a result, the information in other variables is masked, making the classic PCA less informative. In this example, we show that rpca can handle this issue and recover the original data more precisely.

We simulate \( y_n : P \times 1 \) by \((y^{(1)}_n, y^{(2)}_n)^T (n \in \{1, \ldots, N\})\), where \( y^{(1)}_n \) and \( y^{(2)}_n \) are \((P/2) \times 1\) vectors generated by

\[
y^{(1)}_n = \sum_{i=1}^{3} \alpha_i b^{(1)}_i k^{(1)}_{i,n} + \varepsilon^{(1)}_n, \quad y^{(2)}_n = \sum_{i=1}^{3} \alpha_i b^{(2)}_i k^{(2)}_{i,n} + \varepsilon^{(2)}_n,
\]

where \( b^{(1)}_i \) and \( b^{(2)}_i \) \((i \in \{1, 2, 3\})\) are \((P/2) \times 1\) vectors generated independently by QR decomposition, \( k^{(1)}_{i,n}, k^{(2)}_{i,n} \) \((i \in \{1, 2, 3\}, n \in \{1, \ldots, N\})\) are generated independently from N(0, 1), \( \varepsilon^{(1)}_n \) and \( \varepsilon^{(2)}_n \) are both \((P/2) \times 1\) error vectors with elements generated independently from N(0, 1) and N(0, 3) respectively. Furthermore, \((\alpha_1, \alpha_2, \alpha_3) = (7, 5, 3)\).

Thus, \( y_n \) is made up of two parts with vastly different variations. We can visualise the data and variance of a 100 × 100 sample matrix of \( y_n \) in Figures 2 and 3. The colour in both images shows the size of the value: the darker the colour, the larger the value. Figure 2
displays the original data matrix, and it is evident that some of the right regions have much more variations than the rest. The top half of Figure 3, which shows the sample variances of the original data, emphasises the widely different variances. However, the differences in variances are reduced after transforming the data, as shown in the bottom section of Figure 3, which shows the sample variances of \( \exp(\mathbf{y}_{i,n}) \) \( i \in \{1, \ldots, P\} \). The transformation aids in reducing the effect of the heterogeneity in variances on PCA results.

Next we compare the performance of cpca and rpca in approximating both in-sample and out-of-sample data. We simulate this example for several sets of \((P, N): (50, 40), (50, 100), (100, 100), (100, 200), (200, 190)\), which includes the situations of \( P<N \), \( P=N \), as well as \( P>N \). Furthermore, despite the fact that the values of \( P \) and \( N \) are not extremely large, we can consider \((50, 40), (100, 100)\) and \( (200, 190) \) to be high-dimensional settings because the ratios \( P/N \) \( \geq 1 \). Table 3 displays the average in-sample and out-of-sample MSEs. The rpca has much smaller in-sample MSEs than the cpca. The out-of-sample MSEs are large, while rpca still has smaller out-of-sample MSEs than cpca. It is obvious that rpca outperforms cpca in recovering simulated data with widely varying variances. For such data, classic PCA focuses on variables with huge variances while ignoring others that may be equally relevant. However, as demonstrated in Figure 3, our proposed method automatically shrinks those differences, resulting in a more accurate approximation.

### 4.2. Example 2: outliers

Outliers in data are becoming more common as the volume of data grows. This example simulates data with outliers and shows that our proposed method is robust to such data since our transformation can reduce the extreme of outliers.
Table 3. Average in-sample and out-of-sample MSEs from Example 1.

| (P, N)    | (50, 40) | (50, 100) | (100, 100) | (100, 200) | (200, 190) |
|-----------|----------|-----------|------------|------------|------------|
| **In-sample MSE** |          |           |            |            |            |
| rpca      | 0.282    | 0.290     | 0.281      | 0.283      | 0.272      |
| cpca      | 1.516    | 1.565     | 1.281      | 1.292      | 1.142      |
| **Out-of-sample MSE** |          |           |            |            |            |
| rpca      | 2.013    | 2.013     | 2.013      | 2.013      | 2.013      |
| cpca      | 2.921    | 2.921     | 2.921      | 2.921      | 2.921      |

The average in-sample and out-of-sample mean squared errors (MSE) for reconstructing simulated data of Example 1. Smaller values are highlighted and stand for better reconstruction performance.

We first simulate $P \times 1$ vector $y_n$ by

$$y_n = \sum_{i=1}^{3} \alpha_i b_i k_{i,n} + e_n,$$

which is identical to how we constructed $y_n^{(1)}$ in Example 1, except with dimension $P$ rather than $P/2$. After simulating $N$ samples, we have a matrix $Y: P \times N$ with columns of $y_1, \ldots, y_N$. Then we randomly substitute 2.5%, 6.4%, 14.4% of the components in this matrix with values generated independently from $N(0, 6)$ or $N(0, 36)$. Thus, about 2.5%, 6.4% or 14.4% of the elements in $Y$ are outliers, with a variance 6 or 36.

The same as Example 1, we show values and variances of a $100 \times 100$ sample for Example 2 (outlier proportion is 6.4% and from $N(0, 36)$) in Figure 4 and 5. Figure 4 clearly shows some squares that are significantly darker or lighter in colour than the others.
and these are outliers. Figure 5 shows that there are some huge variances (top part of the figure) in the original data produced by outliers, which is not a good indicator for classical PCA, whereas our method can decrease those differences (bottom part of the figure) by the proposed transformation. We try different sets of \((P, N)\) (the same as Example 1) and report the average in-sample MSEs in Table 4 and the average out-of-sample MSEs in Table 5. In all circumstances, the rpca outperforms the c pca in the in-sample part. When there are fewer outliers or no extremely huge outliers (case 1 and case 2), c pca performs well in the out-of-sample part. However, when the proportion or variance of the outliers is increased (case 3 and case 4), rpca outperforms c pca. To summarise, the rpca method performs well in a variety of outlier circumstances.

4.3. Example 3: heavy-tailed data

Now we will look into if rpca works effectively with data from various types of heavy-tailed distributions. Because a considerable amount of real-world data has been demonstrated to be heavy tailed, it is quite likely that a dataset with large dimensions contains heavy-tailed variables. We simulate data from the t-distribution, the Pareto distribution and the Cauchy distribution, all of which are quite common heavy-tailed distributions in real-world data. As a benchmark, we simulate data from the normal distribution.

We simulate \(P \times 1\) vector \(y_n\) by

\[
y_n = \sum_{i=1}^{3} \alpha_i b_i k_{i,n} + \epsilon_n,
\]

Figure 5. Variance from Example 2.
Table 4. Average in-sample MSEs from Example 2.

| (P, N)   | (50, 40) | (50, 100) | (100, 100) | (100, 200) | (200, 190) |
|----------|----------|-----------|------------|------------|------------|
| Case 1: proportion 2.5% from N(0,6) | | | | | |
| rpca     | 0.226    | 0.235     | 0.216      | 0.219      | 0.203      |
| cpca     | 0.508    | 0.526     | 0.375      | 0.380      | 0.301      |
| Case 2: proportion 6.4% from N(0,6) | | | | | |
| rpca     | 0.233    | 0.241     | 0.223      | 0.226      | 0.211      |
| cpca     | 0.536    | 0.550     | 0.404      | 0.410      | 0.336      |
| Case 3: proportion 6.4% from N(0,36) | | | | | |
| rpca     | 0.235    | 0.242     | 0.224      | 0.226      | 0.212      |
| cpca     | 0.864    | 0.894     | 0.759      | 0.776      | 0.714      |
| Case 4: proportion 14.4% from N(0,36) | | | | | |
| rpca     | 0.243    | 0.252     | 0.237      | 0.240      | 0.227      |
| cpca     | 1.315    | 1.375     | 1.281      | 1.301      | 1.243      |

The average in-sample mean squared errors (MSE) for reconstructing simulated data of Example 2. Smaller values are highlighted and stand for better reconstruction performance. Each case illustrates a different scenario of outliers.

which is the same as the first step in Example 2, except \( k_{i,n} (i \in \{1, 2, 3\}, n \in \{1, \ldots, N\}) \), are independently generated from N(0, 1) for the normal distribution, \( t(2) \) for the \( t \)-distribution, Pareto(\( \text{scale} = 0.5, \text{shape} = 1.5 \)) for the Pareto distribution (by function \( \text{rpareto} \) in R package VGAM), and Cauchy(\( \text{location} = 0, \text{scale} = 1 \)) for the Cauchy distribution. For this example, we try \((P, N) \in \{(100, 100), (200, 190)\}\) and the average in-sample MSEs and out-of-sample MSEs are reported in Table 6.

First, we notice that the in-sample performance of rpca is better than that of cpca on normal-distributed data, while the differences are not extremely large, implying that our proposed method is not worse than the classical PCA on normal-distributed data. Second, rpca has smaller in-sample and out-of-sample MSEs than cpca for data from the three heavy-tailed distributions. The uncertainty of the second moments of the heavy-tailed data is one of the reasons behind cpca’s poor performance. For example, because the Cauchy distribution has no finite second moments, the sample covariances computed in cpca are invalid, resulting in the extremely poor performance displayed in Table 6. However, the transformation of rpca assures that the transformed data have finite second moments, ensuring that PCA on transformed data is feasible.

5. Empirical application

In this section, we applied the robust PCA approach to a real dataset to show how it may be used in real-world data analysis. The data, which contain 77 variables and 1080 samples, are obtained from Higuera, Gardiner & Cios (2015), which also contains the specifics of the experiment and measurements. The data consist of protein expression measurements of 77 proteins from normal genotype control mice and Down syndrome (DS) mice, both with and without shock and drug treatments. In the experiment, 72 mice were used, and 15 measurements of each protein were taken. As a result, each protein has 1080 (=72 \times 15) expression measurements. We performed a preprocessing step to deal with missing values.
Table 5. Average out-of-sample MSE from Example 2.

| (P, N)     | (50, 40) | (50, 100) | (100, 100) | (100, 200) | (200, 190) |
|------------|---------|---------|---------|---------|---------|
|            |         |         |         |         |         |
| Case 1: proportion 2.5% from $N(0, 6)$ |         |         |         |         |         |
| rpca       | 1.492   | 1.062   | 1.221   | 0.962   | 1.144   |
| cpca       | **1.029** | **0.854** | **0.859** | **0.725** | **0.795** |
| Case 2: proportion 6.4% from $N(0, 6)$ |         |         |         |         |         |
| rpca       | 1.594   | 1.121   | 1.304   | 1.019   | 1.232   |
| cpca       | **1.161** | **0.921** | **0.959** | **0.791** | **0.905** |
| Case 3: proportion 6.4% from $N(0, 36)$ |         |         |         |         |         |
| rpca       | 2.409   | 1.730   | 1.922   | 1.424   | 1.870   |
| cpca       | 4.363   | 3.119   | 3.832   | 2.983   | 3.800   |
| Case 4: proportion 14.4% from $N(0, 36)$ |         |         |         |         |         |
| rpca       | **1.594** | **1.124** | **1.311** | **1.021** | **1.235** |
| cpca       | 2.128   | 1.487   | 1.129   | 1.401   |         |

The average out-of-sample mean squared errors (MSE) for reconstructing simulated data of Example 2. Smaller values are highlighted and stand for better reconstruction performance. Each case illustrates a different scenario of outliers.

Table 6. Average in-sample and out-of-sample MSEs from Example 3.

| (P, N)     | (100, 100) | (200, 190) |
|------------|------------|------------|
|            | Normal     | t          | Pareto     | Cauchy    | Normal     | t          | Pareto     | Cauchy    |
|            |            |            |            |           |            |            |            |           |
| In-sample MSE |
| rpca       | **0.212** | **0.235** | **0.200** | **0.258** | **0.198** | **0.221** | **0.194** | **0.245** |
| cpca       | 0.335      | 1.174      | 1.318      | 127.242   | 0.278      | 0.840      | 1.058      | 388.595   |
| Out-of-sample MSE |
| rpca       | **1.277** | **1.230** | **1.208** | **1.226** | **1.210** | **1.220** | **1.190** | **1.237** |
| cpca       | 2.289      | 1.678      | 1.758      | 1.106 $\times 10^8$ | 1.767      | 1.571      | 1.976      | 9.833 $\times 10^7$ |

The average in-sample and out-of-sample mean squared errors (MSE) for reconstructing simulated data of Example 3. Smaller values are highlighted and stand for better reconstruction performance.

The histograms of the expression measurements for the first 12 proteins in the data are shown in Figure 6. Although certain proteins have almost normal-distributed expression levels, the majority of proteins, including DTRK1A, ITSN1, pCAMKII, and pERK, have heavy tails or extreme outliers in their expression levels. As a result, robust statistical approaches are required to analyse these data.

We first compare the robust PCA (rpca) and classic PCA (cpca) approximations for the entire dataset using four criteria: the mean squared error (MSE) of the low-rank representation, the number of principals extracted based on threshold 0.8 of the total variance, the estimated smallest spiked eigenvalue, and the spiked ratio $P/(N \hat{\gamma})$ (which we discussed in Section 3.2). The results are reported in Table 7. Both of the spiked ratios are small, with the rpca one being larger than the cpca one. It could indicate that rpca reduces the spiked eigenvalues and the smallest spiked one is more biased than that of cpca. On the other hand, rpca outperforms cpca when seven leading eigenvalues are selected with a threshold 0.8. It is worth mentioned...
that, although rpca is more flexible than cpla due to a larger $\hat{\tau}$, the better out-of-sample performance provided later illustrates its appropriate flexibility.

One possible analysis for this dataset is to categorise the mice based on their protein expression levels. There were 38 control mice and 34 DS mice in the study. For the treatment and control groups, the experiment in Higuera, Gardiner & Cios (2015) included shock and drug treatment. The shock treatments were of two types: context-shock (CS), which allowed the mice to explore a novel cage for several minutes before gave a brief electric shock, and shock-context (SC), which did the opposite. Including with and without the drug memantine, the mice are divided into eight groups. As a result, each group contains seven to nine mice. The number of mice in each class is shown in Table 8. ‘c’ stands for the control group, and ‘t’ stands for the test group, which is made up of DS mice. ‘m’ stands for the drug memantine, and ‘s’ stands for saline, which acts as a placebo.

We use principal logistic regression to classify a subset of the data for the groups ‘c-CS-s’ and ‘t-CS-s’. The shock and drug treatment were identical for these two groups, but the genotype was different. The normal mice are in one group, whereas the DS mice are in the other. The comparison of these two groups is biologically meaningful, according to Higuera, Gardiner & Cios (2015), because it is related to the initial trisomy versus control differences. We intend to use protein expression levels by principal logistic regression to distinguish DS mice from normal ones.

Figure 6. The histogram of the expression measurements for the first 12 proteins.
Table 7. The comparison of rpca and cpca on the entire data. The mean squared errors (MSE) for reconstructing the mice data, the number of components used, the ratio of dimension $p$ to the product of the sample size $N$ and the estimated $r$th largest eigenvalue, and the estimated $r$th largest eigenvalue. The better reconstruction performance of the rpca method in comparison to the cpca method is evidenced by the smaller MSE, which is highlighted in the table.

| Method | MSE | $\tilde{r}$ | ratio ($P/(N\tilde{r}_f)$) | $\tilde{r}_f$ |
|--------|-----|------------|-----------------|-------------|
| rpca   | 0.009 | 7         | 0.558           | 0.126       |
| cpca   | 0.014 | 3         | 0.137           | 0.512       |

There are 240 measurements and 77 proteins in the subgroup. We first randomly divided the data into training (75%) and test sets (25%) in order to cross-validate the prediction performance of the rpca and cpca. Then, using the rpca and cpca on the training data, we extract the eigenvectors and construct the principal components as the design matrix of the logistic regression. For rpca, the principal design matrix is constructed by $\hat{B} \top [\hat{Y} \top, \hat{Y} \top] \top$, where $\hat{B}$ is the $(2P) \times \hat{r}_1$ eigenvector matrix of the transformed training data, $\hat{r}_1$ is the estimated number of eigenvalues of rpca, and $\hat{Y}$ is the original training data with $P$ variables. The equivalent principal design matrix for cpca is $\hat{D} \top \hat{Y}$, where $\hat{D}$ is the $P \times \hat{r}_2$ eigenvector matrix of the original data, $\hat{r}_2$ is the estimated number of eigenvalues of cpca, and $\hat{Y}$ is the original data. The principal design matrices and class labels are then used to build logistic models and produce prediction values for the test set. If the prediction value is larger than 0.5, we set it to be ‘t-CS-s’; otherwise we set it to be ‘c-CS-s’. Finally, we record the prediction accuracies for both methods. We repeat the process 1000 times to ensure that we have a variety of training and test sets. The histogram of the prediction accuracies and

Figure 7. Comparing the classification on mice data.
Table 8. Number of mice in each class, from Higuera, Gardiner & Cios (2015).

| Classes            | No. of mice |
|--------------------|-------------|
| Control mice       |             |
| c-SC-s             | 9           |
| c-SC-m             | 10          |
| c-CS-s             | 9           |
| c-CS-m             | 10          |
| Down syndrome(DS) mice |         |
| t-SC-s             | 9           |
| t-SC-s             | 9           |
| t-SC-s             | 7           |
| t-SC-s             | 9           |

the mean accuracy for both methods is shown in Figure 7. We can observe that when the principal design matrix constructed from robust PCA is used to fit the logistic model, almost all of the prediction accuracy is larger than 0.5, with the majority of them being around 0.78. However, cpca performs much worse than rpca, with the majority of the prediction accuracy hovering around 0.68. This is because heavy-tailed measurements and outliers affect the validity of the cpca, while the rpca method reduces these effects and results in a better performance. Our proposed method can help efficiently distinguish DS mice from the normal ones based on protein expression levels. This example shows that the robust PCA can definitely perform an essential role in classification models and other statistical analyses.

6. Conclusion

In this paper, we addressed the challenge of applying PCA on high-dimensional data in the presence of several types of heterogeneities, especially the heavy tailedness. To deal with potential heterogeneities, we proposed a robust PCA based on a characteristic function-type transformation, which is especially effective when the data are heavy tailed (e.g. with infinite variance). We show that, assuming a spiked covariance structure for the data, the approach is more robust than classical PCA in terms of excess error. We also studied how the transformation affected the spikeness of the spiked covariance structure. We illustrate through simulations that the transformation retains a well-separable spiked covariance matrix. The proposed method, in particular, should perform well when the original data have infinite variance, but the classical method is invalid. With the presence of heterogeneities in the data, simulations and empirical analysis show that the proposed robust PCA method outperforms the classical PCA method. Besides, the proposed method can detect non-linear relationships between variables.

References

ANDERSON, T.W. (2003). An Introduction to Multivariate Statistical Analysis, 3rd edn. Hoboken: John Wiley & Sons Inc.
BAXTER, M.J. (1995). Standardization and transformation in principal component analysis, with applications to archaeometry. Applied Statistics 44, 513–527.
BLANCHARD, G., BOUSQUET, O. & ZWALD, L. (2007). Statistical properties of kernel principal component analysis. Machine Learning 66, 259–294.
Cai, T., Han, X. & Pan, G. (2017). Limiting laws for divergent spiked eigenvalues and largest non-spiked eigenvalue of sample covariance matrices. arXiv preprint arXiv:1711.00217.

Candes, E.J., Li, X., Ma, Y. & Wright, J. (2011). Robust principal component analysis? Journal of the ACM (JACM) 58, 1–37.

Chen, L., Dolado, J.J. & Gonzalo, J. (2021). Quantile factor models. Econometrica 89, 875–910.

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

Debruyne, M., Hubert, M. & Van Horebeek, J. (2010). Detecting influential observations in kernel PCA. Computational Statistics & Data Analysis 54, 3007–3019.

Donoho, D.L. (2000). High-dimensional data analysis: the curses and blessings of dimensionality. AMS Math Challenges Lecture 1, 32.

He, Y., Kong, X., Yu, L. & Zhang, X. (2020). Large-dimensional factor analysis without moment constraints. Journal of Business and Economic Statistics 40(1), 302–312.

Higuera, C., Gardiner, K.J. & Cios, K.J. (2015). Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. PloS ONE 10, e0129126.

Hoffmann, H. (2007). Kernel PCA for novelty detection. Pattern Recognition 40, 863–874.

Johnstone, I.M. (2001). On the distribution of the largest eigenvalue in principal components analysis. The Annals of Statistics 29, 295–327.

Johnstone, I.M. & Titterington, D.M. (2009). Statistical challenges of high-dimensional data. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 367, 4237–4253.

Jolliffe, I.T. (2002). Principal Component Analysis. Springer Series in Statistics, 2nd edn. New York: Springer-Verlag.

Lam, C., Yao, Q. & Bathia, N. (2011). Estimation of latent factors for high-dimensional time series. Biometrika 98, 901–918.

Lee, S., Zou, F. & Wright, F.A. (2014). Convergence of sample eigenvalues, eigenvectors, and principal component scores for ultra-high dimensional data. Biometrika 101, 484–490.

Li, G. & Chen, Z. (1985). Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and Monte Carlo. Journal of the American Statistical Association 80, 759–766.

Li, W., Wang, Q., Yao, J. & Zhou, W. (2022). On eigenvalues of a high-dimensional spatial-signal covariance matrix. Bernoulli 28, 606–637.

Locantore, N., Marron, J., Simpson, D., Tripoli, N., Zhang, J., Cohen, K., Boente, G., Fraiman, R., Brumback, B., Croux, C. et al. et al. (1999). Robust principal component analysis for functional data. Test 8, 1–73.

Mika, S., Schölkopf, B., Smola, A.J., Müller, K.R., Scholz, M. & Rätsch, G. (1998). Kernel PCA and de-noising in feature spaces. In NIPS. pp. 536–542.

Morales-Jimenez, D., Johnstone, I.M., McKay, M.R. & Yang, J. (2021). Asymptotics of eigenstructure of sample correlation matrices for high-dimensional spiked models. Statistica Sinica 31, 571.

She, Y., Li, S. & Wu, D. (2016). Robust orthogonal complement principal component analysis. Journal of the American Statistical Association 111, 763–771.

Shen, D., Shen, H. & Marron, J.S. (2016a). A general framework for consistency of principal component analysis. Journal of Machine Learning Research 17, 1–34.

Shen, D., Shen, H., Zhu, H. & Marron, J.S. (2016b). The statistics and mathematics of high dimension low sample size asymptotics. Statistica Sinica 26, 1747–1770.

Vidal, R., Ma, Y. & Sastry, S. (2016). Generalized Principal Component Analysis, 1st edn. New York: Springer-Verlag.

Wang, W. & Fan, J. (2017). Asymptotics of empirical eigenstructure for high dimensional spiked covariance. Annals of Statistics 45, 1342.

Yata, K. & Aoshima, M. (2012). Effective PCA for high-dimension, low-sample-size data with noise reduction via geometric representations. Journal of Multivariate Analysis 105, 193–215.