AUTHOR	Una Radojičić, Niko Lictzén, Klaus Nordhausen, Joni Virta
TITLE	Dimension Estimation in Two-Dimensional PCA
YEAR	2021
DOI	10.1109/ISPA52656.2021.9552114
VERSION	Author's accepted manuscript
CITATION	U. Radojičić, N. Lictzén, K. Nordhausen and J. Virta, "Dimension Estimation in Two-Dimensional PCA," 2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA), 2021, pp. 16-22, doi: 10.1109/ISPA52656.2021.9552114.

Dimension Estimation in Two-Dimensional PCA

Una Radojičić
Computational Statistics Group
Vienna University of Technology
Vienna, Austria
una.radojicic@tuwien.ac.at

Niko Lietzén
Department of Mathematics and Statistics
University of Turku
Turku, Finland
niko.lietzen@utu.fi

Klaus Nordhausen
Department of Mathematics and Statistics
University of Jyväskylä
Jyväskylä, Finland
klaus.k.nordhausen@jyu.fi

Joni Virta
Department of Mathematics and Statistics
University of Turku
Turku, Finland
joni.virta@utu.fi

***Abstract*—We propose an automated way of determining the optimal number of low-rank components in dimension reduction of image data. The method is based on the combination of two-dimensional principal component analysis and an augmentation estimator proposed recently in the literature. Intuitively, the main idea is to combine a scree plot with information extracted from the eigenvectors of a variation matrix. Simulation studies show that the method provides accurate estimates and a demonstration with a finger data set showcases its performance in practice.**

***Index Terms*—augmentation, dimension estimation, dimension reduction, image data, scree plot**

I. Introduction

A classical problem in image processing is that of low-rank image reconstruction where the original image is decomposed into a superposition of several low-rank components. The process has numerous practical applications, the most well-known of these perhaps being *eigenfaces*, see [1], in which a collection of facial pictures is decomposed using a joint set of low-rank components. Typically, each low-rank component represents a particular collection of facial features (face, mouth and eye shapes, etc.) and the original faces are obtained as weighted combinations of them. This representation allows, for example, the generation of artificial faces by choosing the weights of the components randomly, see [2]. Another common application, not specific to any type of image data, is image compression where the least relevant low-rank components are discarded to achieve a size reduction.

A problem shared by all applications of low-rank image reconstruction is the need to choose a suitable number of low-rank components. On one hand, we want to retain a large enough number to not lose any relevant information whereas, on the other hand, the number of components should be kept sufficiently small to avoid including noise and redundancies. In practice, the optimal cut-off point is not known *a priori*. Typical solutions involve either rule of thumb where enough components are selected to reach a pre-specified amount of "explained variation" [3, Chapter 6] or more involved statistical procedures [4], [5]. However, these approaches are either highly subjective or involve strict distributional assumptions, which hinders their applicability in the context of image data.

The work of NL was supported by the Academy of Finland (Grant 321968)
The work of JV was supported by the Academy of Finland (Grant 335077)

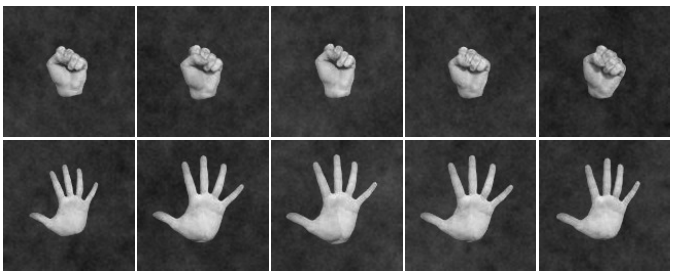


Fig. 1. A collection of images from the *fingers* data set.

Motivated by the previous, the objective of the current work is to propose an automatic tool for determining the optimal number of components. Contrary to eigenfaces and related similar methods, we do not vectorize the set of images but instead treat them throughout as matrices, such that each element represents the grayscale intensity of the corresponding pixel. A similar approach has previously been successfully applied in the context of image data [6]–[8] where the corresponding methods are often categorized as tensor decompositions. One particular consequence of changing the viewpoint from vectors to matrices is that the rows and the columns of the images are compressed separately in the latter. Consequently, we need to determine optimal cut-off points for the rows and columns separately. While this leads to more involved procedures, it also gives more information on the compressibility of the data when compared to the vector approach, which summarizes the compression using a single number/dimension. As far as we know, automated dimension selection in this context has been developed earlier only by [9] who use Stein's unbiased risk estimation (SURE) for the task.

Our running example in this work will be the *fingers* data set available freely in https://www.kaggle.com/koryakinp/fingers and consisting of 128×128 grayscale images of hands with 0-5 fingers extended. For simplicity, we restrict ourselves to the subset of 3000 pictures depicting either 0 or 5 extended fingers on left hands. A sample of the included images is shown in Figure 1. A naive, non-automated way to determine the dimensionalities of the rows and columns in the data is to run $(2D)^2$PCA [6], a matrix-version of PCA, to produce a pair of scree plots, one for the rows and one for the columns,

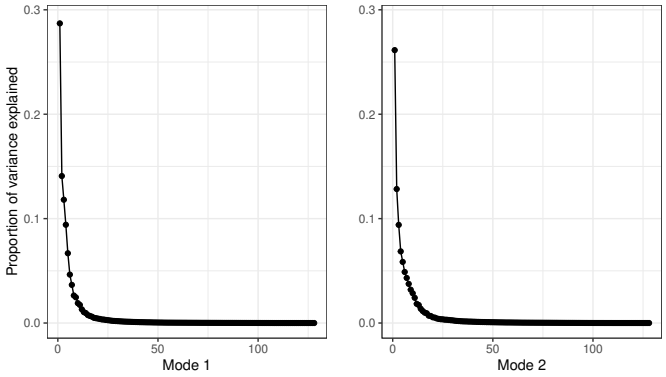


Fig. 2. The scree plots for the mode 1 PCs (left panel) and mode 2 PCs (right panel) extracted from the finger data with (2D)^{2}PCA.

and search for "elbows" (points where the curves turn flat). The two plots are given in Figure 2 and clearly show that no such cut-offs are visible. Hence, Figure 2 on its own is not sufficient to solve the problem and, to supplement it, we propose combining it with the information contained in the eigenvectors produced by (2D)^{2}PCA. This procedure, proposed originally in [10] for vector-valued data under the name of "predictor augmentation", aims to create a "reverse scree plot" where the curve stays flat until the optimal cut-off point is reached and increases afterward. The sum of the "reverse scree plot" and the scree plot is then minimized at the optimal dimension, enabling its straightforward detection, both visually and automatically. A more technical description of the construction of the curve is given in Section III.

The contents of the manuscript are as follows: (2D)^{2}PCA [6], along with our proposed model, is detailed in Section II. The proposed augmentation estimator is presented in Section III. A simulation study and the finger data example are presented in Sections IV and V, respectively. Finally, we end with some discussion in Section VI. Proofs of all technical results will be given in an extended version of the paper.

II. Two-dimensional PCA

Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be our observed set of images, represented as $p_1 \times p_2$ matrices. Throughout the paper we assume that this sample is drawn, independently and identically, from the model,

$$\mathbf{X} = \boldsymbol{\mu} + \mathbf{U}_1 \mathbf{Z} \mathbf{U}_2' + \boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{p_1 \times p_2}$ is the mean image, $\mathbf{U}_1 \in \mathbb{R}^{p_1 \times d_1}$, $\mathbf{U}_2 \in \mathbb{R}^{p_2 \times d_2}$ are unknown matrices with orthonormal columns and $\mathbf{Z}$ is a $d_1 \times d_2$ "core image" with zero mean and dimensions $d_1 \leq p_1$, $d_2 \leq p_2$. Additionally, we make the technical assumptions that $\mathbb{E}\|\mathbf{Z}\|^2 < \infty$ and that $\mathbb{E}(\mathbf{Z}\mathbf{Z}')$ and $\mathbb{E}(\mathbf{Z}'\mathbf{Z})$ are positive definite matrices. The additive $p_1 \times p_2$ noise matrix $\boldsymbol{\varepsilon}$ is taken to be independent from the core $\mathbf{Z}$ and to have a matrix spherical distribution [11], implying that $\mathbb{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}') = \sigma^2 \mathbf{I}_{p_1}$ for some $\sigma^2 \geq 0$.

Model (1) can be thought of as a form of dimension reduction for the images where, for each original image $\mathbf{X}_i$, there exists a low-rank latent core image $\mathbf{Z}_i$ that contains the signal/information content of the image. This signal is then contaminated by the noise $\boldsymbol{\varepsilon}_i$ to produce the observed image. Thus the "true" row and columns dimensions of the images are d_1 and d_2, respectively, and our objective is precisely to determine their values based on the sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$ alone.

The problem of determining the dimension is closely connected to the estimation of the core images and we next briefly review how (2D)^{2}PCA [6] can be used to carry out the latter task. Throughout the following, we assume, without loss of generality, that the random matrix $\mathbf{X}$ is centered in the sense that $\mathrm{E}(\mathbf{X}) = \mathbf{0}$ (this is equivalent to having $\boldsymbol{\mu} = \mathbf{0}$ in (1)). Similarly, for the sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$, we assume that the corresponding mean matrix is zero, $\bar{\mathbf{X}} := (1/n)\sum_{i=1}^n \mathbf{X}_i = \mathbf{0}$. Finally, we also assume, for the remainder of this section, that the dimensions d_1 and d_2 are known.

The (2D)^{2}PCA solution to Model (1) is now found as $\mathbf{V}_1'\mathbf{X}\mathbf{V}_2$ where the $p_1 \times d_1$ matrix $\mathbf{V}_1$ contains any d_1 eigenvectors of $\mathrm{E}(\mathbf{X}\mathbf{X}')$ corresponding to its d_1 largest eigenvalues and the $p_2 \times d_2$ matrix $\mathbf{V}_2$ contains any d_2 eigenvectors of $\mathrm{E}(\mathbf{X}'\mathbf{X})$ corresponding to its d_2 largest eigenvalues. It can be shown that $\mathbf{V}_1$ equals the matrix $\mathbf{U}_1$ in (1) up to post-multiplication by an orthogonal matrix, and similarly for $\mathbf{V}_2$ and $\mathbf{U}_2$. Hence, the (2D)^{2}PCA solution $\mathbf{V}_1'\mathbf{X}\mathbf{V}_2$ is equal to the contaminated core, $\mathbf{Z} + \mathbf{U}_1'\boldsymbol{\varepsilon}\mathbf{U}_2$, up to orthogonal transformations. This orthogonal ambiguity is usually tolerated in practice but in case one wants to get rid of it, additional assumptions on the multiplicities of the eigenvalues of the matrices $\mathrm{E}(\mathbf{Z}\mathbf{Z}')$ and $\mathrm{E}(\mathbf{Z}'\mathbf{Z})$ can be placed. In practice, the matrices $\mathrm{E}(\mathbf{X}\mathbf{X}')$ and $\mathrm{E}(\mathbf{X}'\mathbf{X})$ are unknown and they have to be replaced with their sample counterparts, $(1/n)\sum_{i=1}^n \mathbf{X}_i\mathbf{X}_i'$ and $(1/n)\sum_{i=1}^n \mathbf{X}_i'\mathbf{X}_i$, respectively.

III. Augmentation estimator

A. The main idea

We next detail the proposed strategy of estimating the dimensions d_1 and d_2. By the symmetry of Model (1), it is sufficient to focus on d_1 only, as d_2 can be estimated by applying the same procedure to the transposed sample, $\mathbf{X}_1', \ldots, \mathbf{X}_n'$. A naive way of choosing the dimension would be to plot the eigenvalues of $\mathrm{E}(\mathbf{X}\mathbf{X}')$ as a scree plot and search for an "elbow". As this is often difficult to locate (see Figure 2), our proposed augmentation estimator supplements the scree plot with information extracted from the eigenvectors of $\mathrm{E}(\mathbf{X}\mathbf{X}')$. More precisely, the augmentation estimator concatenates the observed $\mathbf{X}$ with additional artificial normally distributed rows that mimic the first and second-moment behavior of the error $\boldsymbol{\varepsilon}$ in Model (1) to produce the augmented observation $\mathbf{X}^*$. Then the augmented (artificially added) part of the first d_1 eigenvectors of $\mathrm{E}\{\mathbf{X}^*(\mathbf{X}^*)\}$ turns out to be negligible when compared to the augmented parts of the latter eigenvectors, allowing us to distinguish between the eigenvectors belonging to the first d_1, significant, eigenvalues, and the remaining ones. This idea is formalized in the following paragraphs. For

more details on the procedure in general, see [10] where the augmentation estimator was first proposed (in the context of vector-valued data).

In Model (1), we have $\mathrm{E}(\mathbf{X}\mathbf{X}') = \mathbf{U}_1\mathrm{E}(\mathbf{Z}\mathbf{Z}')\mathbf{U}_1' + \mathrm{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}')$ where $\mathbb{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}') = \sigma^2\mathbf{I}_{p_1}$ for some $\sigma^2 \geq 0$. Consequently, the rank of $\mathrm{E}(\mathbf{X}\mathbf{X}') - \sigma^2\mathbf{I}_{p_1}$ is precisely the dimension d_1 we aim to estimate. Let now, for $r > 0$, $\mathbf{X}_S \in \mathbb{R}^{r\times p_2}$ be a random matrix with independent $\mathcal{N}(0, \sigma^2/p_2)$-elements, implying that $\mathbb{E}(\mathbf{X}_S) = \mathbf{0}$ and $\mathrm{E}(\mathbf{X}_S\mathbf{X}_S') = \sigma^2\mathbf{I}_r$. The augmented observation is then defined as the $(p_1 + r) \times p_2$ matrix $\mathbf{X}^* = (\mathbf{X}', \mathbf{X}_S')'$ and satisfies,

$$\mathrm{E}\{\mathbf{X}^*(\mathbf{X}^*)'\} = \begin{pmatrix} \mathbf{U}_1\mathrm{E}(\mathbf{Z}\mathbf{Z}')\mathbf{U}_1' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sigma^2\mathbf{I}_{p_1+r}.$$

If we now define $\mathbf{M}^* := \mathrm{E}\{\mathbf{X}^*(\mathbf{X}^*)'\} - \sigma^2\mathbf{I}_{p_1+r}$, then it is evident that $\mathbf{M}^*$ and $\mathbf{M}_0 = \mathbf{U}_1\mathrm{E}(\mathbf{Z}\mathbf{Z}')\mathbf{U}_1'$ are of the same rank and also have the same positive eigenvalues.

Denote next the eigenvalues of $\mathrm{E}(\mathbf{Z}\mathbf{Z}')$ by $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{d_1} > 0$ and let the $(p_1 + r)$-dimensional vector $\boldsymbol{\beta}_i^* = (\boldsymbol{\beta}_i', \boldsymbol{\beta}_{i,S}')'$, $i = 1, \ldots, p_1 + r$, be any eigenvector of $\mathbf{M}^*$ corresponding to its ith eigenvalue. We call the r-dimensional subvector $\boldsymbol{\beta}_{i,S}$ the augmented part (subvector) of the ith eigenvector. Then, for $i \leq d_1$, $\mathbf{M}^*\boldsymbol{\beta}_i^* = (\mathbf{U}_1\mathrm{E}(\mathbf{Z}\mathbf{Z}')\mathbf{U}_1'\boldsymbol{\beta}_i', \mathbf{0}')' = \lambda_i(\boldsymbol{\beta}_i', \boldsymbol{\beta}_{i,S}')'$, implying that $\boldsymbol{\beta}_{i,S} = \mathbf{0}$ for $i = 1, \ldots d_1$. Observe also that the same does not hold for the later eigenvectors. This specific structure of the augmentation parts will below be used to formulate the augmentation estimator. However, prior to that, we first discuss the estimation of the unknown noise variance σ^2 that plays a crucial part in the above construction.

B. Estimation of noise variance

In the vector setting, [10] used the median of the eigenvalues of the sample covariance matrix as an estimate for σ^2, under the assumption that at least half of the components are noise. A similar approach can be applied in our setting: Let $\hat{\sigma}_1^2 \geq \cdots \geq \hat{\sigma}_{p_1}^2$ be the eigenvalues of $(1/n)\sum_{i=1}^n \mathbf{X}_i\mathbf{X}_i'$ and, analogously, denote the eigenvalues of $\mathrm{E}(\mathbf{X}\mathbf{X}')$ by $\sigma_1^2 \geq \cdots \geq \sigma_{p_1}^2$. Then $(\sigma_1^2, \ldots, \sigma_{p_1}^2) = (\lambda_1 + \sigma^2, \ldots, \lambda_{d_1} + \sigma^2, \sigma^2, \ldots, \sigma^2)$ which, together with the fact that $\hat{\sigma}_i^2$ serve as estimators of σ_i^2, implies that we can estimate σ^2 as the median $\hat{\sigma}^2 := \mathrm{med}\{\hat{\sigma}_1^2, \ldots, \hat{\sigma}_p^2\}$ as long as the assumption $d_1 < p_1/2$ is fulfilled. However, since our overall objective is to estimate both d_1 and d_2, under this approach one would have to assume both $d_1 < p_1/2$ and $d_2 < p_2/2$, which can be seen as somewhat strict, restricting the core image to be at most one-fourth of the original image in size (in terms of the number of pixels). We next weaken this assumption by using simultaneously information from both the rows and the columns of the noise matrix $\boldsymbol{\varepsilon}$.

Observe that as $\boldsymbol{\varepsilon}$ follows a matrix spherical distribution, so does $\boldsymbol{\varepsilon}'$, implying that $\mathrm{E}(\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}) = (\sigma')^2\mathbf{I}_{p_2}$, for some $(\sigma')^2 > 0$. More precisely, for any $i = 1, \ldots, p_1$ and $j = 1, \ldots, p_2$,

$$\sum_{k=1}^{p_2} \mathbb{E}(\varepsilon_{ik}^2) = \sigma^2, \quad \sum_{k=1}^{p_1} \mathbb{E}(\varepsilon_{kj}^2) = (\sigma')^2. \tag{2}$$

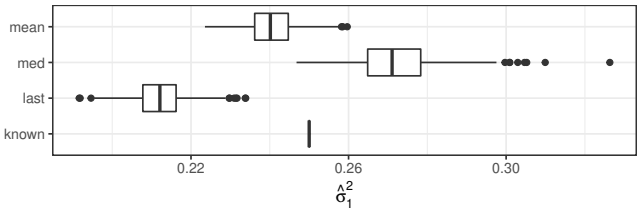


Fig. 3. Boxplot of σ_1^2 estimates in Model 3; $r = 10$, $s = 50$.

By summing the first expression of (2) over $i = 1, \ldots, p_1$ and the second one over $j = 1, \ldots, p_2$, one obtains $p_1\sigma^2 = p_2(\sigma')^2$, implying that

$$\sigma^2 = \frac{p_2}{p_1}(\sigma')^2. \tag{3}$$

We next use the identity (3) to obtain a pooled estimator for the variance σ^2. For the ordered eigenvalues $\sigma_1^2, \ldots, \sigma_{p_1}^2$ and $(\sigma_1')^2, \ldots, (\sigma_{p_2}')^2$ of the matrices $\mathrm{E}(\mathbf{X}\mathbf{X}')$ and $\mathrm{E}(\mathbf{X}'\mathbf{X})$, respectively, we define the set $S := \{\sigma_1^2, \ldots, \sigma_{p_1}^2, \frac{p_2}{p_1}(\sigma_1')^2, \ldots, \frac{p_2}{p_1}(\sigma_{p_2}')^2\}$. We also define its sample counterpart $\hat{S} := \{\hat{\sigma}_1^2, \ldots, \hat{\sigma}_{p_1}^2, \frac{p_2}{p_1}(\hat{\sigma}_1')^2, \ldots, \frac{p_2}{p_1}(\hat{\sigma}_{p_2}')^2\}$, where $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_{p_1}^2$ and $(\hat{\sigma}_1')^2, \ldots, (\hat{\sigma}_{p_2}')^2$ are the eigenvalues of the matrices $(1/n)\sum_{i=1}^n \mathbf{X}_i\mathbf{X}_i'$ and $(1/n)\sum_{i=1}^n \mathbf{X}_i'\mathbf{X}_i$, respectively.

Remark 1. *To clarify the scaling constant p_2/p_1, consider a scenario where the entries of $\boldsymbol{\varepsilon}$ are uncorrelated and have zero mean and variance $\delta^2 > 0$. Then, $\mathrm{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}') = \sum_{i=1}^{p_2}\delta^2\boldsymbol{I}_{p_1} = p_2\delta^2\boldsymbol{I}_{p_1}$. Similarly, $\mathrm{E}(\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}) = p_1\delta^2\boldsymbol{I}_{p_2}$, showing that the noise variance accumulates with the number of columns.*

The median of the set $\hat{S}$ is now a natural estimator of σ^2 under the assumption that

$$d_1 + d_2 < \frac{p_1 + p_2}{2}. \tag{4}$$

Obviously, also other quantiles of the set $\hat{S}$ can be used to estimate σ^2 (assuming that suitable analogs for (4) hold); see the following lemma. For example, in the simulation study, we will use $\min\{\hat{S}\}$ which requires minimal assumptions but, as a downside, has a strong downward bias, meaning that for finite sample sizes it mostly underestimates the true noise variance.

Lemma 1. *Let $\hat{\sigma}_q^2$ be the qth quantile of $\hat{S}$ and $\bar{\sigma}_q^2$ be mean of those elements of $\hat{S}$ that are smaller than or equal to $\hat{\sigma}_q^2$.*

i) *If $d_1+d_2 < (1-q)(p_1+p_2)$, then $\hat{\sigma}_q^2$ and $\bar{\sigma}_q^2$ are consistent estimators of σ^2. Especially, under* (4)*, $\mathrm{med}\{\hat{S}\}$ and $\bar{\sigma}_{0.5}^2$ are consistent estimators of σ^2.*

ii) *If $d_1 + d_2 < p_1 + p_2$, then $\min\{\hat{S}\}$ is a consistent estimator of σ^2.*

Given that all estimates of the noise variance are consistent they might behave quite differently as illustrated in Figure 3. The effect of under- and overestimation of the noise variance for our procedure is discussed in the following remark.

Remark 2. *Assume that the augmented subvector $\boldsymbol{X}_S$ has independent $\mathcal{N}(0, \sigma_S^2/p_2)$-elements. Then*

$$\begin{aligned}\boldsymbol{M}^* &= \mathrm{E}\{\boldsymbol{X}^*(\boldsymbol{X}^*)'\} - \sigma_S^2 \boldsymbol{I}_{p_1+r} \\ &= \begin{pmatrix} \boldsymbol{U}_1(\mathrm{E}(\boldsymbol{Z}\boldsymbol{Z}') + (\sigma^2 - \sigma_S^2)\boldsymbol{I}_{p_1})\boldsymbol{U}_1' & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix},\end{aligned}$$

and first p_1 non-zero eigenvalues of $\boldsymbol{M}^*$ are $\lambda_i + (\sigma^2 - \sigma_S^2)$, $i = 1, \ldots, p_1$, where $\lambda_i = 0$ for $i > d_1$. In practice, as discussed in Section III-C, we replace $\lambda_i + (\sigma^2 - \sigma_S^2)$ with $\max\{0, \lambda_i + (\sigma^2 - \sigma_S^2)\}$, $i = 1, \ldots, p_1$, to compensate for overestimation of σ^2 and to avoid negative eigenvalues. Let now $\sigma_S^2 = \sigma^2 + \delta$, where $0 \leq \delta < \lambda_{d_1}$, where $\delta > 0$ corresponds to amount of overestimation of σ^2. Then, $\max\{0, \lambda_i + (\sigma^2 - \sigma_S^2)\} = \lambda_i - \delta$, $i = 1, \ldots, d_1$, and $\max\{0, \lambda_i + (\sigma^2 - \sigma_S^2)\} = 0$, for $i > d_1$, implying that the rank of $\boldsymbol{M}^*$ is d_1 and that the nontrivial eigenvalues have been shifted by $-\delta$.

Hence, the following method is robust towards slight overestimation of the noise variance, where such behavior is related to thresholding eigenvalues of $\hat{\mathbf{M}}^*$ below by 0. The "allowed" amount of overestimation depends on the smallest non-trivial eigenvalue of $\mathbb{E}(\mathbf{Z}\mathbf{Z}')$ and no such tolerance is allowed for underestimation of σ^2. Though Remark 2 explains behaviour of the method at the population level, approximation to the same phenomenon surely holds in the sample case.

C. Augmentation estimator

We are now equipped to define the augmentation estimator. Let $\mathbf{X}_{1,S}, \ldots, \mathbf{X}_{n,S}$ be a sample of i.i.d. $r \times p_2$ matrices with elements drawn from the standard normal distribution $\mathcal{N}(0, 1)$. Define the augmented observations as the $(p_1 + r) \times p_2$ matrices $\mathbf{X}_i^* := (\mathbf{X}_i', \hat{\sigma}\mathbf{X}_{i,S}')'$, $i = 1, \ldots, n$, where $\hat{\sigma}^2$ is one of the estimates of the noise variance σ^2 defined earlier. A sample estimate $\hat{\mathbf{M}}^*$ of the matrix $\mathbf{M}^*$ is then obtained as

$$\hat{\mathbf{M}}^* = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i^* \mathbf{X}_i^{*\prime} - \hat{\sigma}^2 \mathbf{I}_{p_1+r},$$

whose first p_1 eigenvectors we denote in the following by $\hat{\boldsymbol{\beta}}_1^*, \ldots, \hat{\boldsymbol{\beta}}_{p_1}^*$. Mimicking [10], we define the normalized scree plot curve,

$$\Phi_n : \{0, 1, \ldots, p_1\} \to \mathbb{R}, \quad \Phi_n(k) = \hat{\lambda}_{k+1} / \left(\sum_{i=1}^{k+1} \hat{\lambda}_i + 1 \right),$$

where $(\hat{\lambda}_1, \ldots, \hat{\lambda}_{p_1}) := (\hat{\sigma}_1^2 - \hat{\sigma}^2, \ldots, \hat{\sigma}_{p_1}^2 - \hat{\sigma}^2)$ are the eigenvalues of the matrix $(1/n)\sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i' - \hat{\sigma}^2 \mathbf{I}_{p_1}$ and we define $\hat{\lambda}_{p_1+1} := 0$. However, as the values $\hat{\sigma}_i^2 - \hat{\sigma}^2$ are not necessarily non-negative (unlike their population counterparts), we suggest instead using $\hat{\lambda}_i = \max\{\hat{\sigma}_i^2 - \hat{\sigma}^2, 0\}$, $i = 1, \ldots, p_1$, but with caution as very negative values of $\hat{\sigma}_i^2 - \hat{\sigma}^2$ can indicate that the noise variance σ^2 is not properly estimated, which can happen, e.g., if assumption (4) does not hold. The normalization adjustment in the eigenvalue function Φ_n is done, as in the bootstrap ladle estimator [12], to ensure robustness with respect to scaling of the data, whereas the constant 1 in the denominator is used for stabilization in the extreme case of noise only, that is, when $d_1 = 0$ [10]. The constant 1 also enhances the decreasing pattern of the eigenvalue function, especially in settings with small sample sizes where the $(d_1 + 1)$st eigenvalue might not be very small.

In order to stabilize the final estimate, we conduct the augmentation procedure independently s times and compute the eigenvectors of $\hat{\mathbf{M}}^*$ for each replicate. For $j = 1, \ldots, s$, we denote by $\hat{\boldsymbol{\beta}}_{k,S}^j$ the augmentation subvector of the kth eigenvector of the matrix $\hat{\mathbf{M}}^*$ in the jth replicate. The full eigenvector information is captured by the function,

$$f_n : \{0, 1, \ldots, p_1\} \to \mathbb{R}, \quad f_n(k) = \frac{1}{s}\sum_{j=1}^{s} \|\hat{\beta}_{k,S}^j\|^2,$$

where $\hat{\beta}_{0,S}^j := \mathbf{0}$. We then finally combine the eigenvalue information in Φ_n and the eigenvector information in f_n to form the function $g_n : \{0, 1, \ldots, p_1\} \to \mathbb{R}$,

$$g_n(k) = \sum_{i=0}^{k} \{f_n(k) + \Phi_n(k)\}, \tag{5}$$

and take our estimate $\hat{d}_1$ of the dimension d_1 to be the minimizer of g_n. This choice is intuitively clear as, assuming that $d_1 > 0$, for any $k < d_1$ the eigenvalue part $\Phi_n(k)$ of (5) is large while the eigenvector part $f_n(k)$ is small. For $k > d_1$, the opposite happens and the eigenvalue part is small while the eigenvector part is large. Whereas, at the correct dimension $k = d_1$ both parts are small, implying that the sum curve g_n is (at the population level) minimized precisely at $k = d_1$. Furthermore, in the extreme noise case where $d_1 = 0$, the eigenvalue part in (5) is always negligible, while the eigenvector part is always large, except in the case $k = 0$, in which case it vanishes, again causing the minimum to occur at $k = 0$.

IV. Simulation study

The following simulations and data analysis were conducted using R [13] together with the packages ICtest [14], MixMatrix [15] and tensorBSS [16]. Following [11], we denote by $\mathcal{N}_{p_1,p_2}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2)$ the matrix variate normal distribution with dimensions p_1 and p_2, $p_1 \times p_2$ location $\boldsymbol{\mu}$ and row and column shape matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, respectively. Similarly $\mathcal{T}_{p_1,p_2}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, df)$ denotes the matrix variate t-distribution with degrees of freedom df.

The four models considered in the simulation study are:

$$\begin{aligned} &1)\ \mathbf{X} = \mathbf{U}_1 \mathbf{Z}_t \mathbf{U}_2' + \boldsymbol{\varepsilon}_t, \quad 2)\ \mathbf{X} = \mathbf{U}_1 \mathbf{Z}_t \mathbf{U}_2' + \boldsymbol{\varepsilon}_N, \\ &3)\ \mathbf{X} = \mathbf{U}_1 \mathbf{Z}_N \mathbf{U}_2' + \boldsymbol{\varepsilon}_t, \quad 4)\ \mathbf{X} = \mathbf{U}_1 \mathbf{Z}_N \mathbf{U}_2' + \boldsymbol{\varepsilon}_N, \end{aligned} \tag{6}$$

where $\boldsymbol{Z}_t \sim \mathcal{T}_{3,5}(\mathbf{0}, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, 5)$, $\boldsymbol{\varepsilon}_t \sim \frac{1}{\sqrt{20}}\mathcal{T}_{5,15}(\mathbf{0}, \boldsymbol{I}_5, \boldsymbol{I}_{15}, 5)$, $\boldsymbol{Z}_N \sim \mathcal{N}_{3,5}(\mathbf{0}, \frac{1}{\sqrt{3}}\boldsymbol{\Sigma}_1, \frac{1}{\sqrt{3}}\boldsymbol{\Sigma}_2)$, $\boldsymbol{\varepsilon}_N \sim \frac{1}{\sqrt{20*3}}\mathcal{N}_{5,15}(\mathbf{0}, \boldsymbol{I}_5, \boldsymbol{I}_{15})$. The column shape matrix $\boldsymbol{\Sigma}_1$ is of the form $\boldsymbol{V}_1' \boldsymbol{D}_1 \boldsymbol{V}_1$, where $\boldsymbol{V}_1$ is a random orthogonal matrix and $\boldsymbol{D}_1 = \mathrm{diag}(10, 10, 3)$, and, similarly, $\boldsymbol{\Sigma}_2 = \boldsymbol{V}_2' \boldsymbol{D}_2 \boldsymbol{V}_2$, where $\boldsymbol{V}_2$ is a random orthogonal matrix and $\boldsymbol{D}_2 = \frac{1}{\sqrt{64*3}}\mathrm{diag}(1, 2, 3, 5, 5)$. The

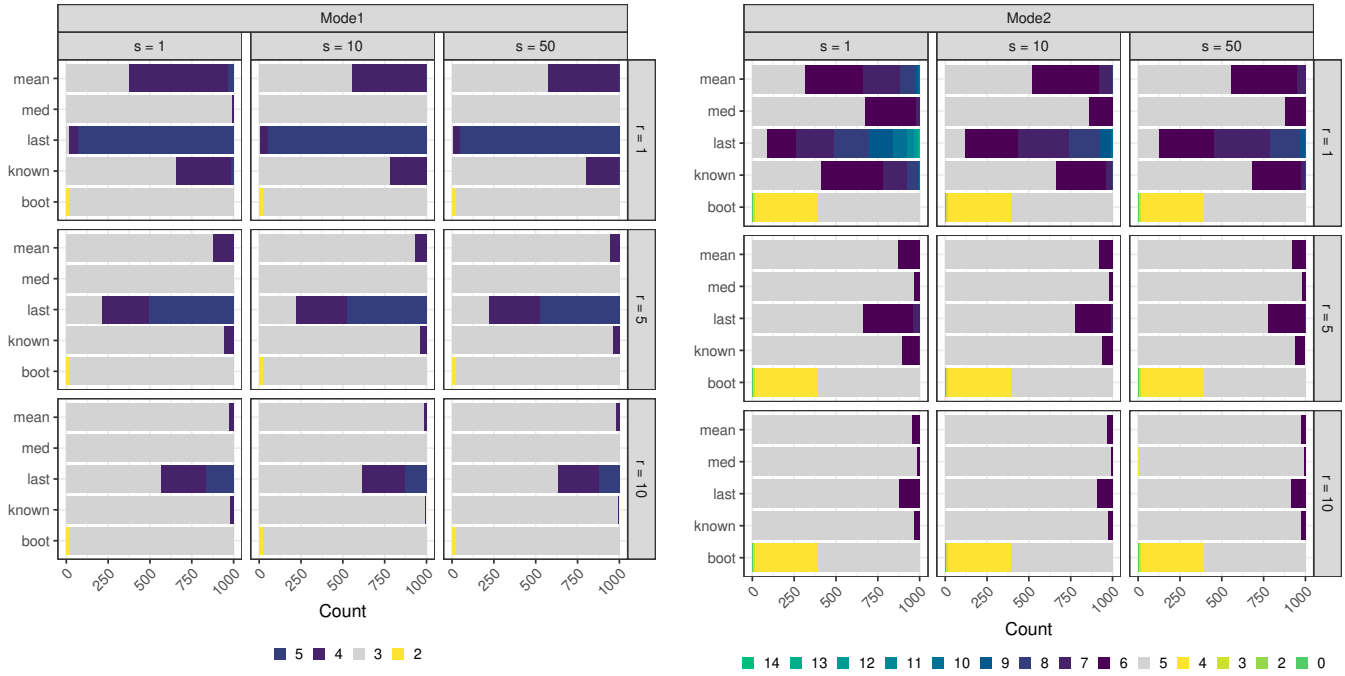


Fig. 4. Frequencies of estimated latent dimensions in Model 3 based on 1000 repetitions. The true latent dimensions are $d_1 = 3$, $d_2 = 5$ and are always marked as grey. Note that for ladle (method boot) the estimates are the same for all s.

mixing matrices $\mathbf{U}_1 \in \mathbb{R}^{5\times 3}$ and $\mathbf{U}_2 \in \mathbb{R}^{15\times 5}$ are taken to be the first 3 and 5 columns of randomly generated orthogonal matrices in $\mathbb{R}^{5\times 5}$ and $\mathbb{R}^{15\times 15}$, respectively. Thus in all four models, $E(\boldsymbol{Z}\boldsymbol{Z}') \approx \text{diag}(0.46, 0.46, 0.14)$, $E(\boldsymbol{Z}'\boldsymbol{Z}) \approx \text{diag}(0.33, 0.33, 0.20, 0.13, 0.07)$, $E(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}') = (1/4)\mathbf{I}_5$ and $E(\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}) = (1/12)\,\mathbf{I}_{15}$.

Then, for each of the four models, 1000 data sets of size $n = 1000$ were created and the dimensions d_1 and d_2 were estimated using all possible combinations of $s \in \{1, 10, 50\}$ and $r \in \{1, 5, 10\}$. To estimate σ^2, the following three approaches were used (i) mean rule: the mean of the 50% smallest values of $\hat{S}$, (ii) median rule: the median of $\hat{S}$ and (iii) last rule: the minimum of $\hat{S}$. To evaluate the cost of estimating σ^2, we also used (iv) the true value, i.e. treat it as known. As an alternative strategy to exploit the information in the variation of the eigenvectors, we also derived a matrix version of the so-called ladle estimator that was suggested for vector data in [12]. The method is based on bootstrapping and we refer to it in the following as the boot rule and the corresponding dimension estimates are based on $m = 200$ bootstrap estimates. Initially, we intended to use the SURE estimates of [9] as a competing method. However, due to its very high computational complexity (note that SURE has to go through all possible combinations of the dimensions d_1 and d_2), we calculated SURE estimates only for 200 independent samples from Model 3. To illustrate the computational complexity, a small timing comparison for 10 repetitions from Model 3 was performed on an i7-8565U processor with 1.80GHz and 16GB RAM. Table I contains the median computation time in seconds when the median rule was used to estimate σ^2 in the augmentation estimator. In this small scale example, SURE is already 50 times slower when compared to the bootstrap-based ladle with 200 bootstrap repetitions, which in turn is slower than any of the considered augmentation-based estimators. However, we have to point out that in all cases where we computed the SURE estimate (the 10 timing comparisons and a batch of 200 additional test runs), it returned the correct dimensions for the core matrix.

Due to space constraints, only the results for Model 3 are presented. However, the performance was in all four models quite similar, and the estimation appears to be, as expected, most difficult when the noise follows the spherical t-distribution. We first look at the performance of different noise estimators. Their estimates are summarized in boxplots in Figure 3 and show, as expected, that the "last" rule always underestimated the true value while the median rule tended to overestimate it. Figure 4 gives then the estimated row and column dimensions for Model 3 and shows that the row dimension is easier to estimate, which is due to the smallest eigenvalue of $\mathbb{E}(\mathbf{Z}\mathbf{Z}')$ being much larger than the smallest eigenvalue of $\mathbb{E}(\mathbf{Z}'\mathbf{Z})$. This is especially true for the median-based estimator of the noise variance, due to its tendency to overestimate the noise variance. As discussed in Remark 2, the larger the smallest signal eigenvalue of $\mathbb{E}(\mathbf{Z}\mathbf{Z}')$ $[\mathbb{E}(\mathbf{Z}'\mathbf{Z})]$ is the more we are allowed to overestimate noise variance. In general, all methods, except bootstrap ladle, seemed to overestimate rather than underestimate the latent dimensions. This is favorable when compared to the alternative, as no

Algorithm 1: Augmentation estimator for d_1.

Input: $\mathbf{X}_1, \dots \mathbf{X}_n \in \mathbb{R}^{p_1 \times p_2}$ centered realizations of a zero-mean matrix from Model (1);

Set the row dimension $r > 0$;
Set the number of augmented replicates $s > 0$;
Calculate $\hat{\mathbf{M}}_1 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i\mathbf{X}_i'$;
Calculate $\hat{\mathbf{M}}_2 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i'\mathbf{X}_i$;
Calculate the estimate of the noise variance based on $\{\hat{\sigma}_1^2, \dots, \hat{\sigma}_p^2, \frac{p_2}{p_1}(\hat{\sigma}_1')^2, \dots, \frac{p_2}{p_1}(\hat{\sigma}_{p_2}')^2)\}$, the pooled set of scaled eigenvalues of $\hat{\mathbf{M}}_1$ and $\hat{\mathbf{M}}_2$. E.g. $\hat{\sigma}^2 = \text{med}(\hat{\sigma}_1^2, \dots, \hat{\sigma}_p^2, \frac{p_2}{p_1}(\hat{\sigma}_1')^2, \dots, \frac{p_2}{p_1}(\hat{\sigma}_{p_2}')^2)$.
Compute $\hat{\lambda}_i = \max\{\hat{\sigma}_i^2 - \hat{\sigma}^2, 0\}$;
for $i \leftarrow 1$ **to** n, $j \leftarrow 1$ **to** s **do**
    Generate an $r \times p_2$ matrix $\mathbf{X}_{i,S}^j$, with entries drawn i.i.d. from $\mathcal{N}(0,1)$ and define the augmented ith observation as $\mathbf{X}_i^{j*} = (\mathbf{X}_i', \hat{\sigma}\mathbf{X}_{i,S}^{j\,\prime})'$;
for $j \leftarrow 1$ **to** s **do**
    Compute the eigendecomposition of the jth replicated matrix

$$\hat{\mathbf{M}}^{j*} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i^{j*}\mathbf{X}_i^{j*\prime} - \hat{\sigma}^2\mathbf{I}_{p_1+r}.$$

    Let $\hat{\boldsymbol{\beta}}_{k,S}^j$ be the augmentation subvector of $\hat{\mathbf{M}}^{j*}$ belonging to the k-th eigenvalue;
The objective function is $g_n(k) = \sum_{i=0}^{k}\{f_n(k) + \Phi_n(k)\}$, where $\hat{\boldsymbol{\beta}}_{0,S}^j = \mathbf{0}$ and $\hat{\lambda}_{p_1+1} = 0$;
Return $\hat{d}_1 = \text{argmin}\{g_n(k) : k = 0, \dots, p_1\}$;

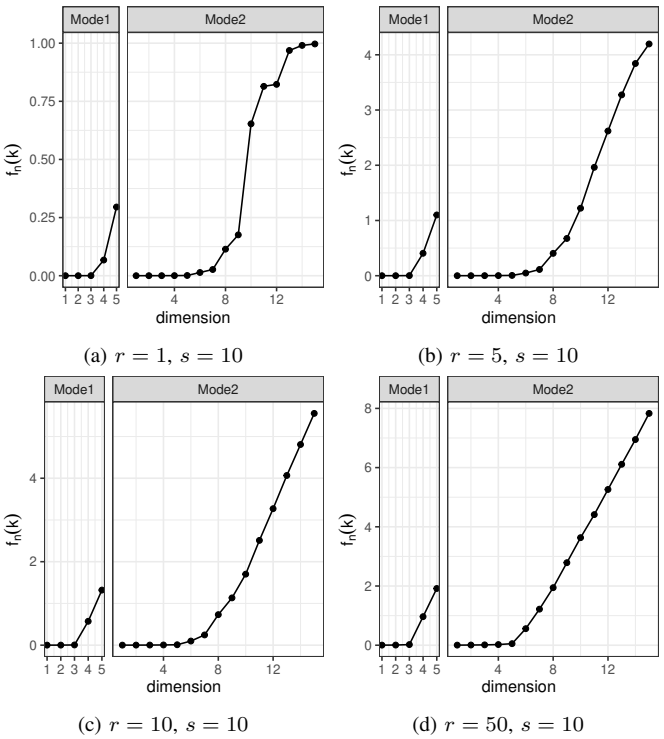
TABLE I
MEDIAN COMPUTATION TIME (SECONDS) OF 10 REPETITIONS FOR $\mathbb{R}^{5\times15}$-MATRICES.

Method	r	s	Time
Augmentation	1	1	0.09
	1	10	0.14
	10	1	0.44
	10	50	4.79
Boot			6.05
SURE			300.20

important signal information is lost. Moreover, using only the last element of $\hat{S}$ to estimate σ^2 is clearly the worst strategy whereas median seems to be the best, which is again in accordance with Remark 2 and the fact that tail eigenvalues tend to underestimate the noise level.

The fact that the median-based augmentation outperforms even the augmentation strategy where the true value of σ^2 is used can again be explained by Remark 2. Namely, the median eigenvalue still belongs to the noise eigenvalues while, as shown in Figure 3, it is mostly larger than the true σ^2. This again stands along with the recommendation that if Assumption (4) holds, we propose that one uses $\hat{\sigma}_{0.5}^2$ as the estimator of the variance. The same is true for the analogues of Assumption 4 and $\hat{\sigma}_{0.5}^2$, as discussed in Lemma 1. The fact that the median-based estimator outperforms the one in which the true noise variance is used should not come as a huge surprise, since the sample size considered in the simulation study is only moderately large implying that the median eigenvalue can still be significantly larger than the true noise variance. With the increase of sample size, differences between the latent dimension estimates obtained using various noise variance estimators will shrink.

Focusing next on the choice of the tuning parameters s and r, we can make, based on the simulation results, the following recommendations. The number of replications s should be as large as possible to reduce variation in the final estimate since the number of replications can be interpreted in the same way as the number of independent re-samples in bootstrapping procedures. The number of rows r of the augmentation sub-matrix seems to have a bigger impact on the final estimate than s. Since $\|\hat{\boldsymbol{\beta}}_{k,S}\|^2 = \sum_{i=1}^{r}\hat{\boldsymbol{\beta}}_{k,S,i}^2$, larger values of r give more emphasis to the non-negligible norm of the augmented subvector, implying that it is better to use larger values of r. This behavior is illustrated in Figure 5 and supported by the simulation study. Observe that scale in Figure 5 increase with r. However, choices of s and r are a trade-off between the computational cost and precision of the obtained estimate. E.g., in the vector case, [12] propose to use $r \approx p_1/5$. To conclude, based on simulations, the augmented estimator performs superbly, especially when considering its computational simplicity.



(a) $r = 1$, $s = 10$ (b) $r = 5$, $s = 10$

(c) $r = 10$, $s = 10$ (d) $r = 50$, $s = 10$

Fig. 5. Value of the function f_n for one data set generated from Model 3 with a sample size of $n = 1000$. The noise variance is assumed to be known.

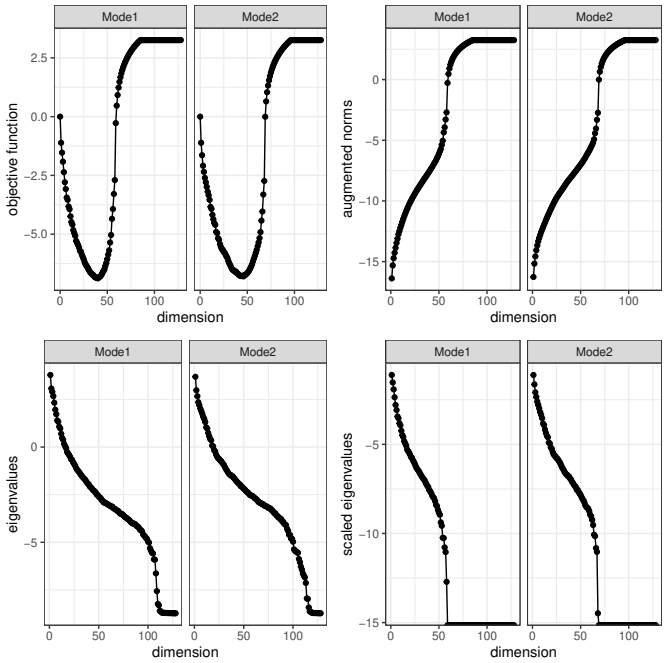


Fig. 6. Logarithmized objective function for the augmented ladle estimator using $r = 26$ augmented components. The objective function is essentially a combination of the augmented norms curve and the scaled eigenvalues curve.

V. Example

For the 3000 128×128 finger images, we decided to use $r = p_1/5 \approx 26$ as recommended in [10] and $s = 100$ and estimate σ^2 using the median rule. Figure 6 visualizes the different parts of the augmented estimator on a logarithmic scale. The figure clearly shows that the eigenvalues alone and the information from the eigenvectors alone are not very helpful in choosing the dimensions for each mode. However, combing the two criteria gives a clear minimum at (40,46) which can be easily picked in an automated way. To evaluate if these dimensions are reasonable, we randomly select a hand showing no fingers and a hand showing all fingers and reconstruct the images based on different numbers of latent components. The reconstructed images together with the original images are presented in Figure 7. These figures reveal that using fewer components than our optimal ones yields blurry images while larger numbers do not yield a significant improvement, indicating that (40,46) would be a good core dimension for the compression.

VI. Discussion

Estimating the number of latent components in matrix-valued PCA in an automated and computationally efficient way has not been possible so far. We extended the augmentation-based estimator from [10] to this setting and demonstrated its excellent performance for both simulated and real data. In future work, we will derive the theoretical properties of the augmented estimator and extend it to the general tensorial PCA case (known, for example, as tPCA [17]) to also cover, e.g., color images and video data.

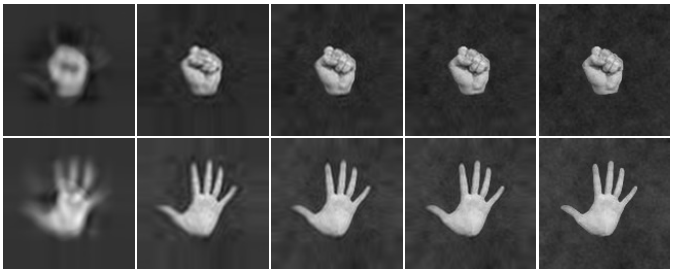


Fig. 7. Dimension reduction for two specific images. From left to right, the images have been reconstructed using (5,5), (25,25), (40,46) and (60,60) components. The rightmost hands correspond to the original images.

References

[1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*, 1991. doi: 10.1109/CVPR.1991.139758 pp. 586–587.

[2] P. Hancock, "Evolving faces from principal components," *Behav Res Methods Instrum Comput*, vol. 32, pp. 327–33, 06 2000. doi: 10.3758/BF03207802

[3] I. Jolliffe, *Principal Component Analysis*, ser. Springer Series in Statistics. Springer, New York, NY, 2002.

[4] J. R. Schott, "A high-dimensional test for the equality of the smallest eigenvalues of a covariance matrix," *J Multivar Anal*, vol. 97, no. 4, pp. 827–843, 2006. doi: 10.1016/j.jmva.2005.05.003

[5] K. Nordhausen, H. Oja, and D. E. Tyler, "Asymptotic and bootstrap tests for subspace dimension," *arXiv preprint arXiv:1611.04908*, 2016.

[6] D. Zhang and Z.-H. Zhou, "$(2D)^2$PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1-3, pp. 224–231, 2005. doi: 10.1016/j.neucom.2005.06.004

[7] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans Neural Netw*, vol. 19, no. 1, pp. 18–39, 2008. doi: 10.1109/TNN.2007.901277

[8] H. Hung, P. Wu, I. Tu, and S. Huang, "On multilinear principal component analysis of order-two tensors," *Biometrika*, vol. 99, no. 3, pp. 569–583, 2012. doi: 10.1093/biomet/ass019

[9] I.-P. Tu, S.-Y. Huang, and D.-N. Hsieh, "The generalized degrees of freedom of multilinear principal component analysis," *J Multivar Anal*, vol. 173, pp. 26–37, 2019. doi: 10.1016/j.jmva.2019.01.010

[10] W. Luo and B. Li, "On order determination by predictor augmentation," *Biometrika*, 2020. doi: 10.1093/biomet/asaa077

[11] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*, 1st ed. Chapman and Hall/CRC, 1999.

[12] W. Luo and B. Li, "Combining eigenvalues and variation of eigenvectors for order determination," *Biometrika*, vol. 103, pp. 875–887, 2016. doi: 10.1093/biomet/asw051

[13] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2020. [Online]. Available: https://www.R-project.org/

[14] K. Nordhausen, H. Oja, D. E. Tyler, and J. Virta, *ICtest: Estimating and Testing the Number of Interesting Components in Linear Dimension Reduction*, 2021, R package version 0.3-3. [Online]. Available: https://CRAN.R-project.org/package=ICtest

[15] G. Thompson, *MixMatrix: Classification with Matrix Variate Normal and t Distributions*, 2019, R package version 0.2.4. [Online]. Available: https://CRAN.R-project.org/package=MixMatrix

[16] J. Virta, C. L. Koesner, B. Li, K. Nordhausen, H. Oja, and U. Radojicic, *tensorBSS: Blind Source Separation Methods for Tensor-Valued Observations*, 2021, R package version 0.3.8. [Online]. Available: https://CRAN.R-project.org/package=tensorBSS

[17] J. Virta, S. Taskinen, and K. Nordhausen, "Applying fully tensorial ICA to fMRI data," in *2016 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2016. doi: 10.1109/SPMB.2016.7846858 pp. 1–6.