TECHNISCHE
UNIVERSITÄT
WIEN

DIPLOMARBEIT

On estimating the signal dimension in tensorial PCA

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Statistik-Wirtschaftsmathematik

eingereicht von

Christoph Kösner
Matrikelnummer: 01326860

ausgeführt am Institut für Stochastik und Wirtschaftsmathematik
der Fakultät für Mathematik und Geoinformation der Technischen Universität Wien

Betreuer: Prof. Klaus Nordhausen
Mitwirkung: Ass.-Prof. Joni Virta

Wien, am 8. Mai 2023

Kurzfassung

In Modellen mit latenten Variablen, in denen die Dimension der nicht-beobachtbaren Daten als niedriger angenommen wird als die Dimension der beobachtbaren, wird erstere oft per Faustregel festgelegt. In der Hauptkomponentenanalyse handelt es sich hierbei um die Anzahl der zu behaltenden Komponenten.

In dieser Arbeit wurde ein neuartiger Bootstrap-Test für die niedrigeren Dimensionen von latenten Tensoren in einem fehlerbehafteten Tensor Modell entwickelt. Dabei handelt es sich um eine Verallgemeinerung einer Vektor-Bootstrap Methode, die eine sphärische Verteilung für den Fehlervektor annimmt. Der Test kann pro Tensorstufe angewandt werden und seine Test-Statistik basiert auf Eigenwerten einer tensoriellen Version der Kovarianz, die sich ebenfalls für jede Stufe errechnen lässt. Damit ist er stark verknüpft mit der tensoriellen Hauptkomponentenanalyse.

Der Test wurde auf simulierten Daten angewandt, wo sich mittels eines Teile-und-herrsche-Verfahrens die korrekten Dimensionen der Tensoren bestimmen ließen, wenn die Varianz der Fehlerkomponenten nicht ein Vielfaches der Varianz der latenten Tensorkomponenten war. Außerdem wurde der Test auf einem Datensatz aus Graustufen-Bildern angewandt und damit die niedrigeren Dimensionen einer Komponentenmatrix errechnet.

Abstract

In latent variable models, where the unobserved data is assumed to be of lower dimension than the observed, the low rank dimension is often determined by a rule of thumb. In principal component analysis (PCA), this number corresponds to the number of retained principal components.

In this thesis, a new bootstrap test for the low rank dimension of latent tensorial data in a noisy tensor model is developed, which can be applied mode-wise. This test is a generalization of a vector bootstrap method, which assumes a spherical distribution for the random error vector. Its test statistic is based upon the eigenvalues of the tensor mode-covariances and is therefore related to tensorial PCA.

For simulated data, it was demonstrated, that a divide and conquer approach based on the bootstrap test can compute the correct low rank dimension, when the variance of the error tensor components is below the variance of the data components. The test was also applied on a dataset of greyscale pictures to compute the lower dimensions of a component matrix.

Danksagung

Ich möchte hier meinen Eltern Wolfgang und Sieglinde danken, deren lebenslange Unterstützung mir eine freie und vielseitige Entwicklung ermöglicht und mir auch während des Studiums viele Belastungen genommen hat.

Des Weiteren danke ich Professor Klaus Nordhausen und Joni Virta für ihre unendliche Geduld und ihre detailreichen Feedbacks, durch die ich die Qualität dieser Diplomarbeit in hohem Maße steigern und sie somit zu einem für mich selbst zufriedenstellenden Abschluss bringen konnte.

Große Dankbarkeit empfinde ich auch allen meinen Freunden gegenüber, die mir stets eine starke Stütze im Leben sind und durch die meine Studienzeit sehr unterhaltsam wurde.

Außerdem danke ich der Firma Phactum Softwareentwicklung GmbH und hier besonders Geschäftsführer Peter Queteschiner, die es mir ermöglicht haben, Job und das Verfassen der Diplomarbeit miteinander zu vereinbaren und deren Support ich mir während dieser gesamten Zeit sicher sein konnte.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 11.01.2021

Christoph Kösner

Contents

1 Introduction

The origins of data dimension reduction date back to the beginning of the twentieth century, when Pearson developed best fit methods for lines and planes in a system of points in a three dimensional space [?]. This work was continued by [?] in the 30s, which later resulted in the principal component analysis (PCA), one of todays most commonly used tools for reducing large datasets of vectors $\boldsymbol{x}_i \in \mathbb{R}^p, i = 1, \dots, n$ in dimension [?].

Mathematically, PCA makes use of a principal axis transformation, so that the components of the resulting vectors are uncorrelated and the variance held by their first components is maximized. Removing the last components then results in a dataset of vectors $\boldsymbol{y}_i \in \mathbb{R}^q, i = 1, \dots, n$ with $q < p$. If the original data is arranged in a matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ of dimension $p \times n$, PCA reduced the dimension to a $q \times n$ matrix.

PCA is often applied in a model free setting, i.e. without any assumptions on the data. In factor analysis (FA), the observed data is assumed to be the sum of a linear combination of unobserved factors and an additional error. While this is a slightly different philosophical approach, PCA is also often used in FA as a first step to retrieve the unknown factors [?]. Their number, however, is hereby often determined by a rule-of-thumb. We assume the dimension of the unobserved vectors in the following as lower than the dimension of the observed vectors and refer to it as low rank dimension. Probabilistic principal component analysis [?] adds additionally stochastic assumptions on the random error vector, under which an estimator of the low rank dimension can be given [?]. With the assumption of the error vector being spherically distributed, a bootstrap low rank dimension test has been developed [?], which builds the foundation for this thesis.

The wide-scale use of PCA and FA in different sciences including electrical engineering, image analysis, chemistry, psychometrics and many more was enabled by the increase of computational power in the 70s [?]. Around the same time, the problem of reducing datasets of matrices and higher order tensors in dimension without large loss of information attracted attention.

We use the common definition of (data) tensors in data science as elements of $\mathbb{R}^{p_1 \times p_2 \times \dots, p_k}$, we call k the order of the tensor and p_i the dimension of the i-th mode. This definition is closely tied to the definition of tensors in multilinear algebra [?], which are used to describe multilinear functions. For multilinear functions from finite dimensional vector spaces over $\mathbb{R}$, the corresponding tensors can be identified by data tensors in our sense.

A primitive approach to tensor dimension reduction is the transformation of the higher order data into vector form in order to use PCA or another dimension reduction method for vectors [?]. For matrices this is achieved by vectorization, while higher order tensors can be reduced mode-wise in several iterations. Because the dimensions of the original array are lost under such transformation, and with it, which elements were only an index apart, the dimension reduction method loses valuable information.

Considering grey-scale pictures, which are matrices or second order tensors by our definition, pixels, which are connected horizontally would in the resulting vectors be apart by the number of the vertical dimension of the picture.

To take the structural information of tensors into account, several more sophisticated approaches to tensor dimension reduction have been developed. A model-free generalization to PCA is given with the tensor decomposition method Tucker3 [?] and under additional assumptions, the higher order singular value decomposition [?]. More statistical approaches include the multilinear principal component analysis [?] and tensorial PCA [?].

Methods for dimension reductions or estimations in vector models will be explored in Section **??**, an introduction to tensor definitions and operations as well as their application to tensor dimension reductions in Section **??**. The main part of this thesis is Chapter **??**, where we develop a new bootstrap test for the low rank dimension in a tensor model, which is a generalization of the aforementioned vector bootstrap test [?] and can be applied mode-wise. For this, we use a generalized concept of sphericity applicable to random tensors and assume the random error tensor in our model to be tensor spherically distributed. A comparison to an asymptotic tensor dimension test in terms of performance will take place in Chapter **??**, for which data was simulated according to the model assumptions in several scenarios. The bootstrap test is then applied on a real dataset in order to evaluate the possible dimension reduction.

In the following work, vectors and matrices are denoted as lower $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\omega})$ and upper case bold letters $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Lambda})$. This notation accounts for elements from real number spaces and random vectors and matrices alike. Random tensors and elements from $\mathbb{R}^{p_1\times\cdots\times p_k}$ are denoted in calligraphic fonts (e.g. $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$).

We denote column and row vectors as $\boldsymbol{m}_i$ and $\boldsymbol{m}_{(j)}$, so that $\boldsymbol{M} = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_q) = \left(\boldsymbol{m}_{(1)}^T, \ldots, \boldsymbol{m}_{(p)}^T\right)^T$ for $\boldsymbol{M} \in \mathbb{R}^{p\times q}$. The vectorization of a matrix $\boldsymbol{M}$ is a stacked vector $\operatorname{vec}(\boldsymbol{M}) := \left(\boldsymbol{m}_1^T, \ldots, \boldsymbol{m}_q^T\right)^T \in \mathbb{R}^{pq}$ consisting of all column vectors. We sometimes denote parts of matrices as $\boldsymbol{M}_{:,c:d} := (\boldsymbol{m}_c, \ldots, \boldsymbol{m}_d)$ or $\boldsymbol{M}_{a:b,:} := \left(\boldsymbol{m}_{(a)}^T, \ldots, \boldsymbol{m}_{(b)}^T\right)^T$. The operation diag builds for a finite number of square matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_k$ the block diagonal matrix

$$\operatorname{diag}\left(\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_k\right) := \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \ldots & \boldsymbol{0} & \boldsymbol{A}_k \end{pmatrix}$$

or a simple diagonal matrix for elements from $\mathbb{R}$.

For a sample of vector valued data $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots\ldots, \boldsymbol{x}_n)$, the sample covariance matrix is denoted as

$$\boldsymbol{S}_X := \frac{1}{n-1}\sum_{i=1}^{n}\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)^T,$$

with the mean vector $\bar{\boldsymbol{x}} := \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$. The sample variance of n (random) variables $x_1, \ldots, x_n$ is denoted as

$$s^2(x_1, \ldots, x_n) := \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \,,$$

with $\bar{x} := \frac{1}{n}\sum_{i=1}^{n} x_i$.

When matrices are decomposed into eigenvalue and singular value decompositions, we usually denote the diagonal matrices containing the eigen- or singular values as $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$. The element $\lambda_i(\boldsymbol{M})$ is the i-th eigenvalue of $\boldsymbol{M}$, or the i-th diagonal value of $\boldsymbol{\Lambda}$ in the eigendecomposition $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$. $\overline{\lambda(\boldsymbol{M})}$ is the mean value of all eigenvalues of $\boldsymbol{M}$. We always assume that real eigenvalues are ordered, so that $\lambda_1 \geq \cdots \geq \lambda_k$. Additional notations for tensor operations will be introduced in Section **??**.

2 State of the art

This chapter lays the theoretical foundation, necessary for the method development in this thesis. First, the meaning of dimension reduction in the context of multivariate data is explained, for which the principal component analysis is shown from a deterministic point of view. Then, PCA is generalized to stochastic and multilinear models. Tensor notations and theory of random tensors are introduced in the process.

2.1 Dimension estimation for random vector models

2.1.1 Model-free principal component analysis

Principal component analysis (PCA, [?]) is an unsupervised learning method, which, given a centered set of data points $\boldsymbol{x}_i$, $i = 1, \dots, n$, in the Euclidean vector space $\mathbb{R}^p$, finds an orthonormal basis $\boldsymbol{W} = (\boldsymbol{w}_1, \dots, \boldsymbol{w}_p)$ that maximizes the sample variance of the new coordinate representations of the data in each component starting from the first one. As a result, the transformed data has also uncorrelated components.

One method for finding the desired basis elements, which are called the principal component vectors, is an iterative search for linear transformation vectors with $i-1$ orthogonality constraints for the i-th vector. In the first step, where no constraint needs to be considered, the problem amounts to

$$\boldsymbol{w}_1 = \underset{\|\boldsymbol{w}\|=1}{\arg\max} \left\{ \sum_i (\boldsymbol{x}_i \cdot \boldsymbol{w})^2 \right\},$$

where the objective function is proportional to the sample variance of the first components. In matrix notation, we arrange $\boldsymbol{x}_i$ as row vectors of the matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^T \in \mathbb{R}^{n \times p}$, leading to the equivalent formulation

$$\boldsymbol{w}_1 = \underset{\|\boldsymbol{w}\|=1}{\arg\max} \left\{ \|\boldsymbol{X}\boldsymbol{w}\|^2 \right\} = \underset{\|\boldsymbol{w}\|=1}{\arg\max} \left\{ \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{w} \right\}.$$

In the following iterations, the remaining axes can be found by subtraction of the already computed projections

$$\boldsymbol{X}^{(k)} = \boldsymbol{X} - \sum_{s=1}^{k-1} \boldsymbol{X} \boldsymbol{w}_s \boldsymbol{w}_s^T$$

and application of the same optimization problem to the newly created data

$$\boldsymbol{w}_{(k)} = \underset{\|\boldsymbol{w}\|=1}{\arg\max} \left\{ \left\| \boldsymbol{X}^{(k)} \boldsymbol{w} \right\|^2 \right\}.$$

It can be shown, that the resulting matrix $\boldsymbol{W} = (\boldsymbol{w}_1, \dots, \boldsymbol{w}_p)$ can be obtained from the eigendecomposition

$$(n-1)\,\boldsymbol{S}_{\boldsymbol{X}} = \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^T,$$

where $\boldsymbol{S}_{\boldsymbol{X}}$ is the sample covariance of $\boldsymbol{X}$. Because we assumed orthonormality, the eigenvector basis $(\boldsymbol{w}_1, \dots, \boldsymbol{w}_p)$ is unique up to orthogonal transformations under which the eigenspaces are invariant. Another way of obtaining $\boldsymbol{W}$ is the singular value decomposition of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{W}^T$, which follows from the fact, that the squared singular values in

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{W}\boldsymbol{\Gamma}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{W}^T = \boldsymbol{W}\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\boldsymbol{W}^T,$$

resulting from the diagonal matrix $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$ are equal to the eigenvalues $\boldsymbol{\Lambda}$. After the coordinate transformation $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}$, we see by inspection of its empirical covariance matrix

$$(n-1)\boldsymbol{S}_{\boldsymbol{Y}} = \boldsymbol{Y}^T\boldsymbol{Y} = \boldsymbol{W}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{W} = \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{\Lambda},$$

that the new variables are uncorrelated and the empirical variances of the principal components $\boldsymbol{y}_i$ are proportional to the eigenvalues λ_j for $j = 1, \dots, p$.

A common use of PCA is dimension reduction of high dimensional data. Since components with small variance are assumed not to contain much information of the data, the reduction can be achieved by truncation of the transformation matrices, i.e. by removal of the eigenvectors corresponding to small eigenvalues as in

$$\boldsymbol{Y}_{:,1:q} = \boldsymbol{X}\boldsymbol{W}_{:,1:q}.$$

The importance of a single principal component is given by its variance in proportion to the total variance

$$\pi_j = \frac{\lambda_j}{\sum_{i=1}^p \lambda_i}.$$

Often a percentage π is set, which defines how much of the total variance should be contained in the remaining components. The smallest q with $\sum_{i=1}^q \pi_i \geq \pi$ is the number of retained components and therefore the low rank dimension. After the dimension reduction,

$$\hat{\boldsymbol{X}} = \boldsymbol{Y}_{:,1:q}\boldsymbol{W}_{:,1:q}^T$$

is a low rank approximation to the original data.

2.1.2 Probabilistic PCA

Note, that the earlier description of PCA was given in a model-free setting. An approach for a stochastic model that includes an additional error term was presented by Tipping and Bishop [**?**]. In the probabilistic principal component analysis (PPCA), the distribution of the observed p-dimensional random vector $\boldsymbol{t}$ is given by

$$\boldsymbol{t} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{2.1}$$

for a q-dimensional random vector $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_q)$ with $q \leq p$, the mean vector $\boldsymbol{\mu} \in \mathbb{R}^p$, $\boldsymbol{A} \in \mathbb{R}^{p \times q}$ and an error term $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_p)$ which is independent of $\boldsymbol{x}$. Under these assumptions, $\boldsymbol{t}$ is distributed as

$$\boldsymbol{t} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{C}\right)$$

with a covariance matrix $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I}_p$. Given a sample $\boldsymbol{T} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$ from model (**??**), [**?**] gave the ML estimators

$$\begin{aligned} \boldsymbol{A}_{\mathrm{ML}} &= \boldsymbol{U}_q \left(\boldsymbol{\Lambda}_q - \sigma^2 \boldsymbol{I}_q\right)^{1/2} \boldsymbol{R}, \qquad (2.2) \\ \sigma^2_{\mathrm{ML}} &= \frac{1}{p-q} \sum_{j=q+1}^{p} \lambda_j(\boldsymbol{S}_{\boldsymbol{T}}), \end{aligned}$$

where $\boldsymbol{U}_q$ is the matrix consisting of the first q eigenvectors of the sample covariance matrix $\boldsymbol{S}_T$, $\boldsymbol{\Lambda}_q$ the diagonal matrix of the corresponding eigenvalues and $\boldsymbol{R}$ an arbitrary $q \times q$ rotation matrix. In practice, first σ^2_{ML} is estimated and then plugged into equation (**??**) to compute $\boldsymbol{A}_{\mathrm{ML}}$. Note that the low rank dimension q needs to be known for the ML estimators.

An approach for dimension selection was given by Minka [**?**], who used Bayesian model selection, in which the probability of data D, given a set of parameters M can be computed by

$$p\left(D|M\right) = \int_{\Phi} p(D|\theta) p\left(\theta|M\right)\, d\theta,$$

where Φ is the space of unknown parameters and $p(\theta|M)$ the a-priori density given M. In order to find an estimator $\hat{q}$ in the PPCA model, they approximated the resulting integral after careful selection of a-priori densities for $\boldsymbol{U}_q, \boldsymbol{L}, \boldsymbol{R}$ from (**??**) with M set to the possible subspace dimensions k. This resulted in the posteriori distribution

$$\begin{aligned} p(D|k) \approx{}& p(\boldsymbol{U}_q) \left(\prod_{j=1}^{k} \lambda_j(\boldsymbol{S}_T) \right)^{-n/2} \hat{v}^{-n(p-k)/2} \left(2\pi\right)^{(m+k)/2} \left|\boldsymbol{M}\right|^{-1/2} n^{-k/2} \\ \text{with} \quad & m = pk - k(k+1)/2, \quad \hat{v} = \frac{\sum_{j=k+1}^{p} \lambda_j(\boldsymbol{S}_T)}{p-k}, \\ & p(\boldsymbol{U}_q) = 2^{-k} \prod_{i=1}^{k} \Gamma\left(\left(p-i+1\right)/2\right) \pi^{-(p-i+1)/2}, \\ & |\boldsymbol{M}| = \prod_{i=1}^{k} \prod_{j=i+1}^{p} \left(\hat{\lambda}_j^{-1} - \hat{\lambda}_i^{-1}\right) \left(\lambda_i(\boldsymbol{S}_T) - \lambda_j(\boldsymbol{S}_T)\right) n, \end{aligned}$$

where $\hat{\lambda}_i$ are

$$\hat{\lambda}_i = \begin{cases} \lambda_i(\boldsymbol{S}_T) & \text{for } i = 1, \ldots, k \\ \hat{v} & \text{for } i = k+1, \ldots, p \end{cases}.$$

The k, which maximizes $p(D|k)$ is then selected as the low rank dimension estimator $\hat{q}$.

2.1.3 Asymptotic and bootstrap tests for subspace dimension

In the following, we go through statistical tests for dimension selection that are based on asymptotic distributions under more general assumptions. The main focus of this thesis is the extension of the bootstrap methods of [?], which are shown at the end of this section, to random matrices and tensors of higher order.

Asymptotic dimension tests were developed by Nordhausen et. al [?] for the model

$$\boldsymbol{t} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\mu}, \tag{2.3}$$

where $\boldsymbol{x}$ is a centered p-dimensional spherically distributed random vector, $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{A} \in \mathbb{R}^{p\times p}$ is non-singular.

A spherically distributed k-dimensional random vector $\boldsymbol{z}$ is invariant to orthogonal transformations, i.e. $\boldsymbol{O}\boldsymbol{z} \sim \boldsymbol{z}$ for each matrix $\boldsymbol{O}$ of the Stiefel manifold

$$\mathfrak{O}_k := \left\{ \boldsymbol{O} \in \mathbb{R}^{k\times k} : \boldsymbol{O}^T\boldsymbol{O} = \boldsymbol{I}_k \right\}.$$

The covariance matrix of a spherical random vector is a multiple of the identity, since

$$\boldsymbol{\Sigma} = \mathrm{Cov}\left[\boldsymbol{z}\right] = \mathrm{Cov}\left[\boldsymbol{O}\boldsymbol{z}\right] = \boldsymbol{O}\boldsymbol{\Sigma}\boldsymbol{O}^T \quad \forall \boldsymbol{O} \in \mathfrak{O}_k$$

holds. Eigendecomposition of the matrix $\boldsymbol{A} = \boldsymbol{O}\boldsymbol{\Lambda}\boldsymbol{O}^T$ from model (**??**) and sphericity leads to

$$\boldsymbol{t} = \boldsymbol{O}\boldsymbol{\Lambda}\boldsymbol{x} + \boldsymbol{\mu}. \tag{2.4}$$

If the last $p-k$ eigenvalues $\lambda_{k+1}(\boldsymbol{A}), \ldots, \lambda_p(\boldsymbol{A})$ are equal, then the subvector consisting of the last $p-k$ components of $\boldsymbol{\Lambda}\boldsymbol{x}$ is also spherically distributed. In this case, $\boldsymbol{t}$ is called subspherically distributed.

Based on (**??**), the null hypothesis of the low rank dimension test can be formulated as

$$H_{0\hat{q}} : \lambda_1(\boldsymbol{A}) \geq \lambda_2(\boldsymbol{A}) \geq \cdots \geq \lambda_{\hat{q}}(\boldsymbol{A}) > \lambda_{\hat{q}+1}(\boldsymbol{A}) = \lambda_{\hat{q}+2}(\boldsymbol{A}) = \cdots = \lambda_p(\boldsymbol{A}). \tag{2.5}$$

Given an iid sample $\boldsymbol{T} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$ of model (**??**), the covariance matrix $\boldsymbol{S_T}$ is now partitioned into

$$\boldsymbol{S_T} = \begin{pmatrix} \boldsymbol{S}_{11} & \boldsymbol{S}_{12} \\ \boldsymbol{S}_{21} & \boldsymbol{S}_{22} \end{pmatrix} = \begin{pmatrix} \hat{\boldsymbol{U}}_1 & \hat{\boldsymbol{U}}_2 \end{pmatrix} \boldsymbol{\Lambda} \begin{pmatrix} \hat{\boldsymbol{U}}_1^T \\ \hat{\boldsymbol{U}}_2^T \end{pmatrix}$$

so that $\hat{\boldsymbol{U}}_1$ and $\hat{\boldsymbol{U}}_2$ are the eigenvector matrices corresponding to the first $\hat{q}$ and last $p-\hat{q}$ eigenvalues respectively. Using the test statistic

$$T_{\hat{q}} = s^2(\lambda_{\hat{q}+1}(\boldsymbol{S_T}), \ldots, \lambda_p(\boldsymbol{S_T})),$$

[?] derived the asymptotic distribution under the null hypothesis $H_{0\hat{q}}$

$$\frac{n\,(p-\hat{q})\,T_{\hat{q}}}{2d^2\sigma_1} \to \chi^2_{\frac{1}{2}(p-\hat{q}-1)(p-\hat{q}+2)},$$

where d, the average of the last $p-\hat{q}$ eigenvalues and σ_1 can be consistently estimated by

$$\hat{d} = \frac{1}{p-\hat{q}} \sum_{i=\hat{q}+1}^{p} \lambda_i(\boldsymbol{S_T}),$$
$$\hat{\sigma}_1 = \frac{1}{(p-\hat{q})(p-\hat{q}+2)} \frac{1}{n} \sum_{i=1}^{n} \left[\left(\boldsymbol{\epsilon}_{2(i)} - \bar{\boldsymbol{\epsilon}_2}\right)^T \boldsymbol{S}_{22}^{-1} \left(\boldsymbol{\epsilon}_{2(i)} - \bar{\boldsymbol{\epsilon}_2}\right) \right].$$

with $\boldsymbol{\epsilon}_{2(i)} = \hat{\boldsymbol{U}}_2^T \boldsymbol{x}_{(i)}$ and $\bar{\boldsymbol{\epsilon}_2} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\epsilon}_{2(i)}$. Under the assumption of normally distributed noise, the value $\hat{\sigma}_1 = 1$ can be used instead.

Virta and Nordhausen [?] extended the ideas of [?] to the model

$$\boldsymbol{t} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon} \tag{2.6}$$

where $\boldsymbol{\mu} \in \mathbb{R}^p$, $\boldsymbol{A} \in \mathbb{R}^{p\times q}$ has full rank, $\boldsymbol{x}$ is a centered q-dimensional random vector with the covariance matrix $\boldsymbol{\Sigma}$ and $\boldsymbol{\epsilon}$ is a p-dimensional spherical random noise vector, that has finite fourth moments and is independent of $\boldsymbol{x}$.

Similar to PPCA, the covariance matrix of $\boldsymbol{t}$ is

$$\operatorname{Cov}[\boldsymbol{t}] = \operatorname{Cov}[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}] = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I}_p,$$

so that the projection matrix $\boldsymbol{U}$ for the PCA components can be computed from the eigendecomposition of $\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T$. Since the last $p-q$ eigenvalues of this low rank matrix are zero, the eigenvalues of the covariance matrix $\operatorname{Cov}[\boldsymbol{t}]$ are $\left(\lambda_1^2+\sigma^2, \ldots, \lambda_q^2+\sigma^2, \sigma^2, \ldots, \sigma^2\right)$, which allows the test from (??) to be applied. Due to this structure, multiplication by the matrix consisting of the last $p-q$ eigenvectors $\boldsymbol{U}_2$ of $\operatorname{Cov}[\boldsymbol{t}]$ leads to components, solely consisting of the noise elements $\boldsymbol{\epsilon}_2 = \boldsymbol{U}_2^T \boldsymbol{t}$.

Under this model, [?] further developed a bootstrap test, for which the distribution of the data under the hypothesis $H_{0\hat{q}}$ needed to be forced on the bootstrap samples. First, for a given sample $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n$ from model (??), the principal components are computed and arranged in $\boldsymbol{Z}$. From this, bootstrap samples $\boldsymbol{Z}^{(j)*}$ for $j = 1, \ldots, m$ are then drawn, which do not necessary have a spherical distribution in their last components as the noise elements $\boldsymbol{\epsilon}_2$ in the model have. This problem can be solved by either replacing these elements with new values drawn from a spherical normal distribution, which is called parametric bootstrapping or by multiplication with random orthogonal matrices. They additionally suggested the use of permutation matrices as third strategy, since they are a subgroup of the orthogonal group. The original projection matrix is used to backtransform the adapted bootstrap samples of the principal components, allowing the bootstrap statistics to be computed.

The pseudo code of the bootstrap method is shown in Algorithm ??. More general algorithms will be developed in Sections ?? and ??.

```
Data: The data $\boldsymbol{T} = (\boldsymbol{t}_1, \dots, \boldsymbol{t}_n)^T$ consisting of $n$ elements with $p$ components,
      $m$ = the number of bootstrap samples
      $strat$ = the name of the bootstrap strategy
      $\hat{q}$ = the low rank dimension of the null hypothesis
Result: The p-value of the test
begin
    Estimate the sample mean $\bar{\boldsymbol{t}}$ and covariance $\boldsymbol{S_T}$ from $\boldsymbol{T}$ ;
    Compute the eigendecomposition of the sample covariance $\boldsymbol{S_T} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ ;
    Compute the test statistic $T_{\hat{q}} = s^2(\lambda_{\hat{q}+1}(\boldsymbol{S_T}), \dots, \lambda_p(\boldsymbol{S_T}))$;
    Compute the principal components $\boldsymbol{z}_i = \boldsymbol{U}^T(\boldsymbol{t}_i - \bar{\boldsymbol{t}})$ ;
    for $j = 1, \dots, m$ do
        Take a bootstrap sample $(\boldsymbol{z}_1^{(j)}, \dots, \boldsymbol{z}_n^{(j)})$ from $(\boldsymbol{z}_1, \dots, \boldsymbol{z}_n)$ ;
        for $i \in \{1, \dots, n\}$ do
            if $strat$ == "orthogonal" then
                $\boldsymbol{z}_i^{(j)} \leftarrow \operatorname{diag}(\boldsymbol{I}_{\hat{q}}, \boldsymbol{O}_i)\, \boldsymbol{z}_i^{(j)}$ where $\boldsymbol{O}_i$ is a random $(p - \hat{q}) \times (p - \hat{q})$
                  orthogonal matrix;
            end
            if $strat$ == "permutation" then
                $\boldsymbol{z}_i^{(j)} \leftarrow \operatorname{diag}(\boldsymbol{I}_{\hat{q}}, \boldsymbol{P}_i)\, \boldsymbol{z}_i^{(j)}$ where $\boldsymbol{P}_i$ is a random $(p - \hat{q}) \times (p - \hat{q})$
                  permutation matrix;
            end
            if $strat$ == "parametric" then
                Compute $\hat{\sigma}^2 = \frac{1}{p-\hat{q}} \sum_{i=\hat{q}+1}^{p} \lambda_i(\boldsymbol{S}_T)$ ;
                Replace the last $p - \hat{q}$ values of $\boldsymbol{z}_i^{(j)}$ with values from $N(\mathbf{0}, \hat{\sigma}^2 \boldsymbol{I}_{p-\hat{q}})$ ;
            end
            Compute the backtransformation $\boldsymbol{t}_i^{(j)} = \boldsymbol{U}\boldsymbol{z}_i^{(j)} + \bar{\boldsymbol{t}}$;
        end
        Compute the bootstrap test statistic $\hat{t}_j$ from $\boldsymbol{T}^{(j)} = \left(\boldsymbol{t}_1^{(j)}, \dots, \boldsymbol{t}_n^{(j)}\right)^T$;
    end
    return the p-value $\frac{\#\{\hat{t}_j : \hat{t}_j > T_{\hat{q}}\} + 1}{m+1}$;
end
```

Algorithm 1: The bootstrap algorithm for signal dimension estimation by [?].

2.2 Generalization of PCA to higher order tensors

2.2.1 PCA for matrix valued data

In 2DPCA [?] a $p_1 \times p_2$-dimensional random matrix $\boldsymbol{X}$ is transformed into a p_1-dimensional random vector $\boldsymbol{y}$ by multiplication with a projection vector

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{v}.$$

Its goal is the discovery of several projections, which together contain most information of the data. As in PCA, this is measured by the variance the projections have. The covariance matrix of $\boldsymbol{y}$ is

$$\mathrm{Cov}\left[\boldsymbol{y}\right] = \mathbb{E}\left[\left((\boldsymbol{X} - \mathbb{E}\left[\boldsymbol{X}\right])\boldsymbol{v}\right)\left((\boldsymbol{X} - \mathbb{E}\left[\boldsymbol{X}\right])\boldsymbol{v}\right)^T\right],$$

where $\mathbb{E}\left[\boldsymbol{X}\right]$ is the componentwise expected value $(\mathbb{E}\left[\boldsymbol{X}\right])_{ij} := \mathbb{E}\left[x_{ij}\right]$. The trace of $\mathrm{Cov}\left[\boldsymbol{y}\right]$ is therefore

$$\mathrm{tr}\left(\mathrm{Cov}\left[\boldsymbol{y}\right]\right) = \boldsymbol{v}^T \mathbb{E}\left[(\boldsymbol{X} - \mathbb{E}\left[\boldsymbol{X}\right])^T(\boldsymbol{X} - \mathbb{E}\left[\boldsymbol{X}\right])\right]\boldsymbol{v}$$

and can be used in the optimization problem

$$\underset{\|\boldsymbol{v}\|=1}{\arg\max}\left(\mathrm{tr}(\mathrm{Cov}\left[\boldsymbol{y}\right])\right)$$

for the maximization of the sum of the variances of the components of $\boldsymbol{y}$. A set of maximizing vectors with orthogonal restrictions as in PCA is given by the eigenvectors of the scatter matrix $\boldsymbol{\Sigma}_2 := \mathbb{E}\left[(\boldsymbol{X} - \mathbb{E}\left[\boldsymbol{X}\right])^T(\boldsymbol{X} - \mathbb{E}\left[\boldsymbol{X}\right])\right]$. The transformation matrix $\boldsymbol{V} \in \mathbb{R}^{p_2 \times q_2}$ is again obtained by the selection of the first q_2 eigenvectors, i.e. the eigenvectors corresponding to the q_2 largest eigenvalues of $\boldsymbol{\Sigma}_2$. Multiplication $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{V}$ leads to the desired reduction of the number of columns.

To reduce the dimension p_1, the same method can be applied to the transposed random matrix $\boldsymbol{X}^T$, which independently results in a second projection matrix $\boldsymbol{U} \in \mathbb{R}^{p_1 \times q_1}$. In $(2\mathrm{D})^2$PCA [?], both projection matrices are computed and used to obtain the so-called feature matrix

$$\boldsymbol{Z} = \boldsymbol{U}^T\boldsymbol{X}\boldsymbol{V},$$

which now amounts to a dimension reduction in both dimensions.

For a sample of iid random matrices $(\boldsymbol{X}_i, i = 1, \dots, n)$, the two scatter matrices are estimated by $\boldsymbol{S}_1 = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^T$ and $\boldsymbol{S}_2 = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{X}_i - \bar{\boldsymbol{X}})^T(\boldsymbol{X}_i - \bar{\boldsymbol{X}})$ with the matrix mean $\bar{\boldsymbol{X}} := \frac{1}{n}\sum_{i=1}^n \boldsymbol{X}_i$. The projections are computed by

$$\boldsymbol{Z}_i = \hat{\boldsymbol{U}}^T\boldsymbol{X}_i\hat{\boldsymbol{V}}$$

with $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$ consisting of the first eigenvectors from $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ respectively. An approximation of the original data is obtained with

$$\hat{\boldsymbol{X}}_i = \hat{\boldsymbol{U}}\boldsymbol{Z}_i\hat{\boldsymbol{V}}^T.$$

We will see a direct generalization of this method in the following sections.

2.2.2 Tensor definitions and operations

In this section, the tensor definitions and operations necessary for the following work will be defined. We thereby follow the widely established notations from Kolda and Bader [**?**].

A tensor of dimension $p_1 \times \cdots \times p_k$ is called tensor of order k or a tensor with k modes. Vectors and matrices are therefore tensors of order one and two respectively. A fiber is a vector resulting from a tensor, when indices for each mode except one are fixed. A slice is the resulting matrix, when one index of a tensor with three modes is fixed. Slices can be horizontal ($\boldsymbol{X}_{i_1::}$), lateral ($\boldsymbol{X}_{:i_2:}$) or frontal ($\boldsymbol{X}_{::i_3}$).

For tensors of higher order, we use the notation $\mathcal{X}_{i_n=\alpha}$ when we fix the index corresponding to the n-th mode to α. A sample of tensor valued data $(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n)$ with $\mathcal{X}_i \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ for $i = 1, \dots, n$, can be arranged in a tensor $\mathcal{X} \in \mathbb{R}^{p_1 \times \cdots \times p_k \times n}$ of order $k+1$, so that $\mathcal{X}_{i_{k+1}=i} = \mathcal{X}_i$.

The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{p_1 \times p_2 \times \cdots \times p_k}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{p_1} \sum_{i_2=1}^{p_2} \cdots \sum_{i_k=1}^{p_k} x_{i_1 i_2 \dots i_k} \, y_{i_1 i_2 \dots i_k}.$$

The induced tensor norm, which generalizes both Euclidean and Frobenius norm is

$$\|\mathcal{X}\| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{i_1=1}^{p_1} \sum_{i_2=1}^{p_2} \cdots \sum_{i_k=1}^{p_k} x_{i_1 i_2 \dots i_k}^2}.$$

The *m-mode product* of a $p_1 \times p_2 \cdots \times p_k$-dimensional tensor $\mathcal{X}$ and a $q_m \times p_m$-dimensional matrix $\boldsymbol{U}$ results in the $p_1 \times p_2 \times \cdots \times p_{m-1} \times q_m \times p_{m+1} \times p_{m+2} \times \cdots \times p_k$ tensor

$$(\mathcal{X} \times_m \boldsymbol{U})_{i_1 i_2 \dots i_{m-1} j_m i_{m+1} \dots i_k} := \sum_{i_m=1}^{p_m} x_{i_1 i_2 \dots i_{m-1} i_m i_{m+1} \dots i_k} u_{j_m i_m}.$$

This product represents the multiplication of the matrix $\boldsymbol{U}$ with each m-mode fiber of $\mathcal{X}$. The 1-mode and 2-mode product for a tensor of order three is schematically shown in Figure **??**.

The *mode-m unfolding* $\boldsymbol{X}_{(m)}$ reorders the elements of a tensor $\mathcal{X}$ in the $p_m \times (\prod_{j \neq m} p_j)$-dimensional matrix

$$(\boldsymbol{X}_{(m)})_{i,j} := x_{i_1 i_2 \dots i_{m-1}, i, i_{m+1} \dots i_k}$$

with $j = 1 + \sum_{\substack{h=1 \\ h \neq m}}^{k} (i_h - 1) \prod_{\substack{l=1 \\ l \neq m}}^{h-1} p_l$. The mode-1 and mode-2 unfoldings for a third order tensor are represented in Figure **??**.

The outer product of two tensors $\mathcal{X} \in \mathbb{R}^{p_1 \times p_2 \times \cdots \times p_k}, \mathcal{Y} \in \mathbb{R}^{q_1 \times q_2 \times \cdots \times q_h}$ is the $p_1 \times p_2 \times \cdots \times p_k \times q_1 \times q_2 \times \cdots \times q_h$-dimensional tensor

$$(\mathcal{X} \circ \mathcal{Y})_{i_1 i_2 \dots i_k j_1 j_2 \dots j_h} = x_{i_1 i_2 \dots i_k} y_{j_1 j_2 \dots j_h}.$$

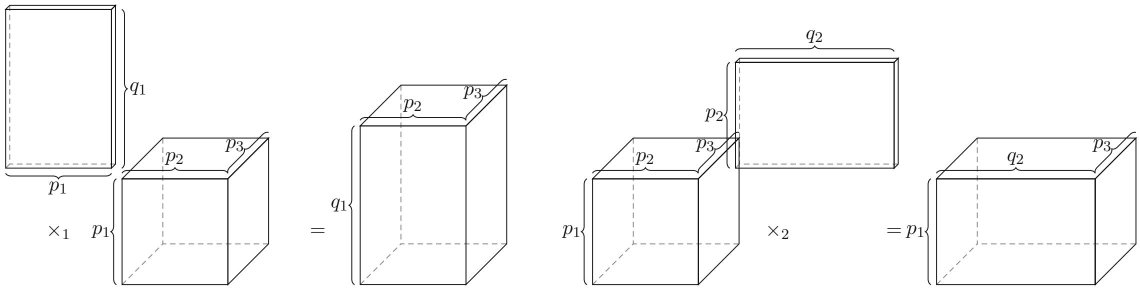

Figure 2.1: Schematic representation of the m-mode product of a third order tensor and matrices on mode $m = 1, 2$

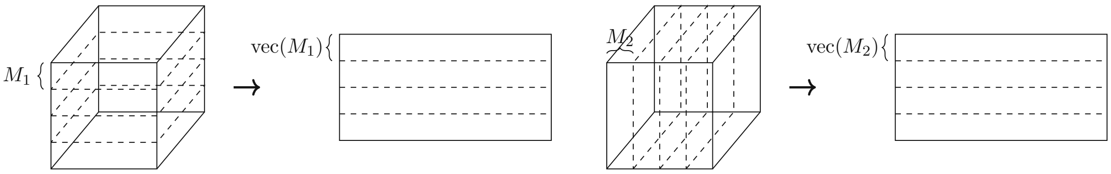

Figure 2.2: Unfolding of a third order tensor in mode $m = 1, 2$

A $p_1 \times p_2 \times \cdots \times p_k$ tensor $\mathcal{X}$ of rank-1 is the outer product of k vectors $\boldsymbol{v}^{(i)} \in \mathbb{R}^{p_i}$

$$\mathcal{X} = \boldsymbol{v}^{(1)} \circ \boldsymbol{v}^{(2)} \circ \ldots \circ \boldsymbol{v}^{(k)}$$
$$x_{i_1 i_2 \ldots i_k} = v^{(1)}_{i_1} v^{(2)}_{i_2} \ldots v^{(k)}_{i_k}.$$

We say a tensor is of rank-n, if it can be written as a sum of not fewer than n rank-1 tensors. The complexity of finding the tensor rank is shown to be NP-hard [**?**]. The canonical decomposition (CANDECOMP, [**?**]) and parallel factors model (PARAFAC, [**?**]) focussed on finding rank-1 decompositions of tensors.

The Kronecker product of two matrices $\boldsymbol{A} \in \mathbb{R}^{p_1 \times q_1}$ and $\boldsymbol{B} \in \mathbb{R}^{p_2 \times q_2}$ is

$$\boldsymbol{A} \otimes \boldsymbol{B} := \begin{pmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \ldots & a_{1q_1}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \ldots & a_{2q_1}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p_1 1}\boldsymbol{B} & a_{p_1 2}\boldsymbol{B} & \ldots & a_{p_1 q_1}\boldsymbol{B} \end{pmatrix} \in \mathbb{R}^{p_1 p_2 \times q_1 q_2} \tag{2.7}$$

and can be helpful in the description of tensor unfoldings.

2.2.3 Tucker- and higher order singular value decompositions

We will now use the shown tensor notations and operations to introduce tensor decompositions, which will then be used in a tensorial version of PCA. In 1966, Tucker [**?**] used a decomposition of third order tensors $\mathcal{X} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$ in the form of

$$\mathcal{X} = \mathcal{Z} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \boldsymbol{A}^{(3)}$$

with square matrices $\boldsymbol{A}^{(1)} \in \mathbb{R}^{p_1 \times p_1}$ $\boldsymbol{A}^{(2)} \in \mathbb{R}^{p_2 \times p_2}$ $\boldsymbol{A}^{(3)} \in \mathbb{R}^{p_3 \times p_3}$ and the so-called core tensor $\mathcal{Z} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$ for development of a generalization of SVD.

Since this decomposition can naturally be generalized to tensors of higher order and might be helpful for dimension reduction, the Tucker decomposition in today's literature is often seen as

$$\mathcal{X} = \mathcal{Z} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times \cdots \times_k \boldsymbol{A}^{(k)}, \tag{2.8}$$

with a core tensor $\mathcal{Z} \in \mathbb{R}^{q_1 \times q_2 \times \cdots \times q_k}$ and matrices $\boldsymbol{A}^{(i)} \in \mathbb{R}^{p_i \times q_i}$ and $q_i \leq p_i$ for $i = 1, \ldots, k$. Note that each tensor has a trivial Tucker decomposition $\mathcal{X} = \mathcal{X} \times_1 \boldsymbol{I}_{p_1} \times \cdots \times_k \boldsymbol{I}_{p_k}$ and can be brought into a different form by linear transformations, as for (**??**),

$$\begin{aligned} \mathcal{X} &= \mathcal{Z} \times_1 \boldsymbol{A}^{(1)} \times \ldots \times_{m-1} \boldsymbol{A}^{(m-1)} \times_m \boldsymbol{A}^{(m)} \boldsymbol{M} \boldsymbol{M}^{-1} \times_{m+1} \boldsymbol{A}^{(m+1)} \times \ldots \times_k \boldsymbol{A}^{(k)} \\ &= \hat{\mathcal{Z}} \times_1 \boldsymbol{A}^{(1)} \times \ldots \times_{m-1} \boldsymbol{A}^{(m-1)} \times_m \hat{\boldsymbol{A}}^{(m)} \times_{m+1} \boldsymbol{A}^{(m+1)} \times \ldots \times_k \boldsymbol{A}^{(k)} \end{aligned}$$

holds for any full rank matrix $\boldsymbol{M} \in \mathbb{R}^{q_m \times q_m}$ with $\hat{\boldsymbol{A}}^{(m)} = \boldsymbol{A}^{(m)} \boldsymbol{M}$ and $\hat{\mathcal{Z}} = \mathcal{Z} \times_m \boldsymbol{M}^{-1}$.

The relationship between the mode-m unfolding of tensor $\mathcal{X}$ and $\mathcal{Z}$ of a Tucker decomposition can be described with the Kronecker product from (**??**) as

$$\boldsymbol{X}_{(m)} = \boldsymbol{A}^{(m)} \boldsymbol{Z}_{(m)} \left(\boldsymbol{A}^{(k)} \otimes \ldots \otimes \boldsymbol{A}^{(m+1)} \otimes \boldsymbol{A}^{(m-1)} \otimes \ldots \otimes \boldsymbol{A}^{(1)} \right)^T. \tag{2.9}$$

A special Tucker decomposition, which is a multi-mode analogue of SVD, gained popularity under the name of higher order singular value decomposition (HOSVD) with the work of De Lathauwer et. al [**?**], who presented a complex version of the following theorem:

Theorem 1. *Every tensor $\mathcal{X} \in \mathbb{R}^{p_1 \times p_2 \times \cdots \times p_k}$ can be written as the product*

$$\mathcal{X} = \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times \cdots \times_k \boldsymbol{U}^{(k)},$$

in which

1. *$\boldsymbol{U}^{(i)}$ is an orthogonal matrix for all $i = 1, \ldots, k$*
2. *$\mathcal{S}$ is a $(p_1 \times p_2 \times \cdots \times p_k)$-dimensional real tensor of which the subtensors $\mathcal{S}_{i_n=\alpha}$ have the properties of*
   - a) *all-orthogonality: two subtensors $\mathcal{S}_{i_n=\alpha}$ and $\mathcal{S}_{i_n=\beta}$ are orthogonal for all possible values of n, α and β subject to $\alpha \neq \beta$*

     $$\langle \mathcal{S}_{i_n=\alpha}, \mathcal{S}_{i_n=\beta} \rangle = 0.$$

   - b) *ordering:*

     $$\|\mathcal{S}_{i_n=1}\| \geq \|\mathcal{S}_{i_n=2}\| \geq \cdots \geq \|\mathcal{S}_{i_n=p_n}\| \geq 0$$

     for all possible values of n.

It was shown by [**?**], that the matrices $\boldsymbol{U}^{(i)}$ of the decomposition can be obtained from the singular value decompositions of the tensor unfoldings

$$\boldsymbol{X}_{(i)} = \boldsymbol{U}^{(i)} \boldsymbol{\Gamma}^{(i)} \boldsymbol{V}^{(i)^T}.$$

As explained in Section **??**, the eigendecompositions

$$\boldsymbol{X}_{(i)}\boldsymbol{X}_{(i)}^T = \boldsymbol{U}^{(i)}\boldsymbol{\Lambda}^{(i)}\boldsymbol{U}^{(i)^T}$$

offer another way of obtaining $\boldsymbol{U}^{(i)}$. The core tensor is then computed by

$$\mathcal{S} = \mathcal{X} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times \cdots \times_k \boldsymbol{U}^{(k)}.$$

2.2.4 Random tensors

We define a $p_1 \times \cdots \times p_k$-dimensional random tensor as array of random variables $x_{i_1 i_2 \ldots i_k}$. Since all tensor operations from Section **??** can be broken down to rearrangements or primitive variable manipulations which can be also applied to random variables, they are equally valid for random tensors.

As in the vector and matrix case, the expected value of a $p_1 \times \cdots \times p_k$ random tensor $\mathcal{X}$ is defined componentwise as

$$(\mathbb{E}[\mathcal{X}])_{i_1,i_2,\ldots i_k} := \mathbb{E}[x_{i_1,i_2,\ldots,i_k}].$$

The covariances between all elements of $\mathcal{X}$ can be expressed in

$$(\operatorname{Cov}[\mathcal{X}])_{i_1 i_2,\ldots,i_k,j_1,j_2,\ldots,j_k} := \operatorname{Cov}(x_{i_1 i_2 \ldots i_k}, x_{j_1 j_2 \ldots j_k}),$$

a tensor of order $2k$. For a random tensor $\mathcal{Z}$ with uncorrelated elements of zero mean and a variance of one, [**?**] stated the following theorem:

Theorem 2. *Let* $\boldsymbol{A}^{(i)} \in \mathbb{R}^{p_i \times p_i}$, $\boldsymbol{\Sigma}_i := \boldsymbol{A}^{(i)}\boldsymbol{A}^{(i)^T}$ *for* $i = 1, \ldots, k$ *and*

$$\mathcal{X} = \mathcal{Z} \times_1 \boldsymbol{A}^{(1)} \times \cdots \times \boldsymbol{A}^{(k)}.$$

Then

1. $\operatorname{Cov}[\mathcal{X}] = \boldsymbol{\Sigma}_1 \circ \boldsymbol{\Sigma}_2 \circ \ldots \circ \boldsymbol{\Sigma}_k$
2. $\operatorname{Cov}[\operatorname{vec}(\mathcal{X})] = \boldsymbol{\Sigma}_k \otimes \boldsymbol{\Sigma}_{k-1} \otimes \cdots \otimes \boldsymbol{\Sigma}_1$
3. $\mathbb{E}\left[\boldsymbol{X}_{(m)}\boldsymbol{X}_{(m)}^T\right] = \boldsymbol{\Sigma}_m \times \prod_{j:j\neq m} \operatorname{tr}(\boldsymbol{\Sigma}_j)$

The third part of the theorem gives information about the covariance structure of a given mode. Consequently, for an uncentered tensor, we define the m-mode covariance

$$\operatorname{Cov}_m[\mathcal{X}] = \rho_m^{-1}\mathbb{E}\left[\left(\boldsymbol{X}_{(m)} - \mathbb{E}\left[\boldsymbol{X}_{(m)}\right]\right)\left(\boldsymbol{X}_{(m)} - \mathbb{E}\left[\boldsymbol{X}_{(m)}\right]\right)^T\right] \tag{2.10}$$

with a normalization constant $\rho_m = \prod_{i\neq m}^k p_i$.

2.2.5 Tensor PCA and multilinear principal component analysis

The tensor PCA (tPCA) from [?] provides a natural way to extend the mathematical methods used in PCA and (2D)^{2}PCA to tensors of higher order. There, a random $p_1 \times p_2 \times \cdots \times p_k$-dimensional tensor $\mathcal{X}$ is transformed by

$$\mathcal{Y} = \mathcal{X} \times_1 (\boldsymbol{U}^{(1)})^T \times_2 (\boldsymbol{U}^{(2)})^T \times_3 \cdots \times_k (\boldsymbol{U}^{(k)})^T$$

with matrices $\boldsymbol{U}^{(i)}$ from the Stiefel manifold $\mathfrak{O}_{q_i,p_i} = \left\{\boldsymbol{O} \in \mathbb{R}^{p_i \times q_i} : \boldsymbol{O}^T\boldsymbol{O} = \boldsymbol{I}_{q_i}\right\}$ for $i = 1, \ldots, k$, which consist of the first q_i eigenvectors of the i-mode covariances $\mathrm{Cov}_i\left[\mathcal{X}\right]$.

The HOSVD method shown in Section **??** is closely related to tPCA for centered tensors (i.e. $\mathbb{E}\left[\mathcal{X}\right] = 0$). In HOSVD, the same operations are applied on a single real-valued tensor, while in tPCA the expected value in (**??**) is needed due to its stochastic nature.

Given a sample of iid distributed random tensors $(\mathcal{X}_1, \ldots, \mathcal{X}_n)$ of dimension $p_1 \times p_2 \times \cdots \times p_k$, the m-mode covariances are estimated by the matrices

$$\boldsymbol{S}_m := \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{X}_{i(m)} - \bar{\boldsymbol{X}}_{(m)}\right)\left(\boldsymbol{X}_{i(m)} - \bar{\boldsymbol{X}}_{(m)}\right)^T \tag{2.11}$$

and decomposed to $\boldsymbol{S}_m = \hat{\boldsymbol{U}}^{(m)}\hat{\boldsymbol{\Lambda}}^{(m)}(\hat{\boldsymbol{U}}^{(m)})^T$ for $m = 1, \ldots, k$, to find a single transformation for all samples.

A different non-stochastic approach for tensor dimension reduction is given with MPCA [?] that shows, how similar transformations can be obtained by formulation and solving of a numerical optimization problem. Given a set of higher order data tensors $\{\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \ldots \mathcal{X}_n\}$ from a tensor space $\mathbb{R}^{p_1 \times p_2 \times \cdots \times p_k}$, its goal is the dimension reduction of the tensors in each mode with a multilinear transformation in the form of

$$\mathcal{Y}_i = \mathcal{X}_i \times_1 (\boldsymbol{U}^{(1)})^T \times_2 (\boldsymbol{U}^{(2)})^T \times_3 \cdots \times_k (\boldsymbol{U}^{(k)})^T$$

with matrices $\boldsymbol{U}^{(i)} \in \mathbb{R}^{p_i \times q_i}$. MPCA was formulated by the problem

$$\left\{\tilde{\boldsymbol{U}}^{(i)}, i = 1, \ldots, k\right\} = \underset{\boldsymbol{U}^{(1)},\ldots,\boldsymbol{U}^{(k)}}{\arg\max}\ \Phi_Y,$$

where we denote by

$$\Phi_Y = \sum_{i=1}^{n}\left\|\mathcal{Y}_i - \bar{\mathcal{Y}}\right\|^2$$

the tensor scatter of the projected samples $\mathcal{Y}_i$. To approach a solution, [?] gave the following theorem:

Theorem 3. *Let $\left\{\tilde{\boldsymbol{U}}^{(i)}, i = 1, \ldots, k\right\}$ be the solution. Then, given all the other projection matrices $\tilde{\boldsymbol{U}}^{(1)}, \tilde{\boldsymbol{U}}^{(2)}, \ldots, \tilde{\boldsymbol{U}}^{(m-1)}, \tilde{\boldsymbol{U}}^{(m+1)}\tilde{\boldsymbol{U}}^{(m+2)}, \ldots, \tilde{\boldsymbol{U}}^{(k)}$ the matrix $\tilde{\boldsymbol{U}}^{(m)}$ consists of the eigenvectors corresponding to the largest q_m eigenvalues of the matrix*

$$\boldsymbol{\Phi}^{(m)} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{X}_{i(m)} - \bar{\boldsymbol{X}}_{(m)}\right)\tilde{\boldsymbol{U}}_{\boldsymbol{\Phi}^{(m)}}\tilde{\boldsymbol{U}}_{\boldsymbol{\Phi}^{(m)}}^T\left(\boldsymbol{X}_{i(m)} - \bar{\boldsymbol{X}}_{(m)}\right)^T \tag{2.12}$$

with

$$\tilde{\boldsymbol{U}}_{\boldsymbol{\Phi}^{(m)}} := \boldsymbol{U}^{(m+1)} \otimes \boldsymbol{U}^{(m+2)} \otimes \cdots \otimes \boldsymbol{U}^{(k)} \otimes \boldsymbol{U}^{(1)} \otimes \boldsymbol{U}^{(2)} \otimes \cdots \otimes \boldsymbol{U}^{(m-1)}.$$

Because the computation of each $\tilde{\boldsymbol{U}}^{(m)}$ is dependent on all other $\tilde{\boldsymbol{U}}^{(i)}$ for $i \neq m$, this theorem does not provide a closed-form solution to the optimization problem, except in the case, where no dimension reduction is desired. Then the orthogonality of $\tilde{\boldsymbol{U}}_{\boldsymbol{\Phi}^{(m)}}$ leads to the so-called full projection and (**??**) reduces to the sample m-mode covariance (**??**) from the tPCA case.

To gradually reduce dimension in different modes, [**?**] used an iterative approach, in which the matrices $\tilde{\boldsymbol{U}}^{(i)}$ are initialized by the full projection and then truncated stepwise. A single truncation leads to a change of $\boldsymbol{\Phi}^{(m)}$, with eigenvalues decreasing in all other modes. Hence, they suggested the recomputation of the eigendecompositions of $\boldsymbol{\Phi}^{(m)}$ in each iteration with consideration of the interdependencies.

The transformation matrices we get from tPCA are used in the following work, which is equal to the full projection results when the iterative approach is discarded.

2.2.6 Spherical tensor distributions

Since the focus of this thesis is a generalization of the model (**??**) in [**?**], a multi-mode analogue of the concept of spherical distributions is necessary. Some theory was layed out for the matrix case.

A left spherical $p \times q$ random matrix has the property $\boldsymbol{O}_1\boldsymbol{X} \sim \boldsymbol{X} \quad \forall \boldsymbol{O}_1 \in \mathfrak{O}_p$. Consequently right spherical distributions can be defined through $\boldsymbol{X}\boldsymbol{O}_2 \sim \boldsymbol{X} \quad \forall \boldsymbol{O}_2 \in \mathfrak{O}_q$. We call a random matrix spherical, iff it is left and right spherical. The following examples of spherical distributions are given in [**?**]:

- The matrix-variate normal distribution $\boldsymbol{X} \sim N_{p,q}\left(\boldsymbol{0}, \boldsymbol{I}_p, \boldsymbol{I}_q\right)$ has the density

$$p(\boldsymbol{X}) = \frac{1}{(2\pi)^{pq/2}} \exp\left(-(1/2)\,\mathrm{tr}\left\{\boldsymbol{X}\boldsymbol{X}^T\right\}\right)$$

- The matrix-variate t-distribution: $\boldsymbol{X} \sim T_{p,q}\left(v, \boldsymbol{0}, \boldsymbol{I}_p, \boldsymbol{I}_q\right)$ has the density

$$p(\boldsymbol{X}) = \frac{\Gamma_p\left[\frac{v+q+p-1}{2}\right]}{(2\pi)^{qp/2}\,\Gamma_p\left[\frac{v+p-1}{2}\right]} \cdot \left|\boldsymbol{I}_p + \boldsymbol{X}\boldsymbol{X}^T\right|^{-(v+q+p-1)/2}$$

  where Γ_p denotes the multivariate Gamma function and v the degree of freedom.

- The uniform distribution on the Stiefel Manifold $\mathfrak{O}_{q,p} = \left\{\boldsymbol{O} \in \mathbb{R}^{p\times q} : \boldsymbol{O}^T\boldsymbol{O} = \boldsymbol{I}_q\right\}$ is given by a probability measure with density

$$p(\boldsymbol{X}) = \left[\mathrm{vol}(\mathfrak{O}_{q,p})^{-1}\right] = \Gamma_q\left(p/2\right)\frac{\pi^{pq/2}}{2^q}$$

  with respect to the Haar measure.

We say a random tensor $\mathcal{X}$ has spherical distribution, if

$$\mathcal{X} \times_m \boldsymbol{O} \sim \mathcal{X} \quad \forall \boldsymbol{O} \in \mathfrak{O}_{p_m}$$

holds for each mode m. This also means, that all possible fibers of the random tensor must be spherically distributed in the vector sense, due to the property of the m-mode multiplication shown in Section **??**. An example of a spherical random tensor can be given by one that consists of iid normal distributed variables.

3 Methodology

3.1 Low rank dimension tests for second-order tensors

The following random matrix model will be a generalization of (**??**). For testing the low rank dimension, we will present a bootstrap test similar to [**?**] and additionally show an asymptotic test provided by Virta in personal communication, which works under similar assumptions. The performance of both tests will compared in the next chapter.

3.1.1 Noisy second order model

The $p_1 \times p_2$ random matrix $\boldsymbol{T}$ obeys the noisy second-order (NS) model, if it can be written as

$$\boldsymbol{T} = \boldsymbol{M} + \boldsymbol{O}^{(1)} \boldsymbol{X} (\boldsymbol{O}^{(2)})^T + \boldsymbol{E} \tag{3.1}$$

where $\boldsymbol{M} \in \mathbb{R}^{p_1 \times p_2}$ is the mean value, $\boldsymbol{O}^{(1)} \in \mathfrak{O}_{q_1,p_1}$, $\boldsymbol{O}^{(2)} \in \mathfrak{O}_{q_2,p_2}$, $\boldsymbol{X}$ is a random $q_1 \times q_2$-dimensional, zero-mean core matrix with $q_1 \leq p_1$, $q_2 \leq p_2$ and the $p_1 \times p_2$-dimensional error $\boldsymbol{E}$ is also random and independent of $\boldsymbol{X}$. We additionally assume, that both $\mathbb{E}\left[\|\boldsymbol{X}\|^2\right]$ and $\mathbb{E}\left[\|\boldsymbol{E}\|^2\right]$ are finite, $\mathbb{E}[\boldsymbol{E}] = \boldsymbol{0}$ and $\boldsymbol{E}$ has spherical matrix-variate distribution.

3.1.2 Bootstrap tests for second-order tensors

In the following, ideas from the bootstrap tests shown in Section **??** are combined with the two-dimensional PCA from Section **??**. For that, we first analyze the scatter covariance matrix of model (**??**). From the zero mean assumption, the independence of $\boldsymbol{E}$ to $\boldsymbol{X}$ and the orthogonality of $\boldsymbol{O}^{(2)}$, we get the representation

$$\begin{aligned} \operatorname{Cov}[\boldsymbol{T}] &= \mathbb{E}\left[\left(\boldsymbol{O}^{(1)} \boldsymbol{X} (\boldsymbol{O}^{(2)})^T + \boldsymbol{E}\right)\left(\boldsymbol{O}^{(1)} \boldsymbol{X} (\boldsymbol{O}^{(2)})^T + \boldsymbol{E}\right)^T\right] \\ &= \boldsymbol{O}^{(1)} \operatorname{Cov}[\boldsymbol{X}] (\boldsymbol{O}^{(1)})^T + \operatorname{Cov}[\boldsymbol{E}]. \end{aligned} \tag{3.2}$$

For matrix spherically distributed $\boldsymbol{E}$, it can be seen by

$$\operatorname{Cov}[\boldsymbol{E}] = \operatorname{Cov}[\boldsymbol{O}\boldsymbol{E}] = \boldsymbol{O} \operatorname{Cov}[\boldsymbol{E}] \boldsymbol{O}^T \quad \forall \boldsymbol{O} \in \mathfrak{O}_{p_1},$$

that the matrix covariance has a similar property as in the vector case and we can conclude, that it is also a multiple of the unit matrix $e\boldsymbol{I}_{p_1}$ with $e \in \mathbb{R}$. The matrix covariance $\operatorname{Cov}[\boldsymbol{X}]$ is of dimension $q_1 \times q_1$, so we can diagonalize $\boldsymbol{O}^{(1)} \operatorname{Cov}[\boldsymbol{X}] (\boldsymbol{O}^{(1)})^T = \boldsymbol{U}\boldsymbol{\Lambda}_1\boldsymbol{U}^T$ with at least

$p_1 - q_1$ eigenvalues that are zero. Using the covariance representation (**??**) and the sphericity of $\boldsymbol{E}$, we obtain the diagonalization

$$\begin{aligned}\operatorname{Cov}\left[\boldsymbol{U}^T\boldsymbol{T}\right] &= \boldsymbol{U}^T\boldsymbol{O}^{(1)}\operatorname{Cov}\left[\boldsymbol{X}\right](\boldsymbol{O}^{(1)})^T\boldsymbol{U} + \operatorname{Cov}\left[\boldsymbol{E}\right] \\ &= \begin{pmatrix}\operatorname{diag}(\lambda_1,\dots,\lambda_{q_1}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix} + e\boldsymbol{I}_{p_1} && (3.3) \\ &= \begin{pmatrix}\operatorname{diag}(\lambda_1+e,\dots,\lambda_{q_1}+e) & \boldsymbol{0} \\ \boldsymbol{0} & e\boldsymbol{I}_{p_1-q_1}\end{pmatrix}. && (3.4)\end{aligned}$$

It follows, that the sample variance of the last $p_1 - q_1$ eigenvalues of $\operatorname{Cov}[\boldsymbol{T}]$ is zero, on which the following low rank dimension bootstrap test will be based on.

Given an iid sample $(\boldsymbol{T}_1,\dots,\boldsymbol{T}_n)$ from model (**??**), we first compute scatter covariance matrices

$$\boldsymbol{S}_1 = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{T}_i - \bar{\boldsymbol{T}}\right)\left(\boldsymbol{T}_i - \bar{\boldsymbol{T}}\right)^T \quad \boldsymbol{S}_2 = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{T}_i - \bar{\boldsymbol{T}}\right)^T\left(\boldsymbol{T}_i - \bar{\boldsymbol{T}}\right)$$

as in the two-dimensional PCA cases. In order to create a test for the null hypothesis

$$\boldsymbol{H}_{0\hat{q}_1} : q_1 = \hat{q}_1, \tag{3.5}$$

we compute the sample variance of the last eigenvalues of $\boldsymbol{S}_1$ using the test statistic

$$t^2_{\hat{q}_1} := \frac{1}{\hat{d}}s^2\left(\lambda_{\hat{q}_1+1}(\boldsymbol{S}_1),\dots,\lambda_{p_1}(\boldsymbol{S}_1)\right),$$

where $\hat{d} = \frac{1}{p_1-\hat{q}_1}\sum_{i=\hat{q}_1+1}^{p_1}\lambda_i(\boldsymbol{S}_1)$ is an estimator of the last eigenvalues and used for normalization. Under the null and by (**??**), this value should now be approximately zero. To obtain bootstrap samples with spherical error in the given mode, three strategies similar to the ones explained in Section **??** are implemented. From the eigendecompositions $\boldsymbol{S}_1 = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}_1\hat{\boldsymbol{U}}^T$ and $\boldsymbol{S}_2 = \hat{\boldsymbol{V}}\hat{\boldsymbol{\Lambda}}_2\hat{\boldsymbol{V}}^T$, we use $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$ to obtain the matrix version of the principal components $\boldsymbol{Z}_i = \hat{\boldsymbol{U}}^T\left(\boldsymbol{T}_i - \bar{\boldsymbol{T}}\right)\hat{\boldsymbol{V}}$ for $i = 1,\dots,n$. The bootstrap samples $(\boldsymbol{Z}_1^{(j)},\boldsymbol{Z}_2^{(j)},\dots,\boldsymbol{Z}_n^{(j)})$ for $j = 1,\dots,m$, are then drawn from the results. By (**??**), the submatrices $(\boldsymbol{Z}_1^{(j)})_{\hat{q}_1+1:p_1,:},\dots,(\boldsymbol{Z}_n^{(j)})_{\hat{q}_1+1:p_1,:}$, now approximately only contain bootstrapped error under the null-hypothesis, which needs to be transformed, so that it fulfills the sphericity assumption of the model. The orthogonal bootstrap strategy is the multiplication of these submatrices by randomly drawn orthogonal matrices

$$\boldsymbol{Z}_i^{*(j)} = \operatorname{diag}\left(\boldsymbol{I}_{\hat{q}_1},\boldsymbol{O}_i\right)^T\boldsymbol{Z}_i^{(j)} \quad \boldsymbol{O}_i \in \mathfrak{O}_{p_1-\hat{q}_1}. \tag{3.6}$$

As in **??**, the second strategy is multiplication with random $(p_1 - \hat{q}_1)\times(p_1 - \hat{q}_1)$ dimensional permutation matrices $\boldsymbol{P}_i$ instead of $\boldsymbol{O}_i$ in (**??**), since they build a subgroup of the Stiefel manifold. Finally, the parametric bootstrap method is the complete replacement of the submatrices $(\boldsymbol{Z}_1^{(j)})_{\hat{q}_1+1:p_1,:},\dots,(\boldsymbol{Z}_n^{(j)})_{\hat{q}_1+1:p_1,:}$ by iid normally distributed values, which are by Section **??** spherical. The resulting data $\left(\boldsymbol{Z}_1^{*(j)}, \dots, \boldsymbol{Z}_n^{*(j)}\right)$ gets transformed back in all strategies, resulting in the bootstrap samples

$$\boldsymbol{T}_i^{(j)} = \hat{\boldsymbol{U}} \boldsymbol{Z}_i^{*(j)} \hat{\boldsymbol{V}}^T + \bar{\boldsymbol{T}}$$

from which now the test statistics $t_{\hat{q}_1}^{(j)}$ can be computed.

The same reasoning can be applied on the transposed data, for which we analogously to (**??**) get the matrix covariance

$$\operatorname{Cov}\left[\boldsymbol{T}^T\right] = \boldsymbol{O}^{(2)} \operatorname{Cov}\left[\boldsymbol{X}^T\right] (\boldsymbol{O}^{(2)})^T + \operatorname{Cov}\left[\boldsymbol{E}^T\right].$$

Here, the left term of (**??**) contains at least $p_2 - q_2$ zero eigenvalues. A test for the hypothesis

$$H_{0\hat{q}_2} : q_2 = \hat{q}_2$$

can therefore be established in the same way.

3.1.3 Asymptotic tests for second-order tensors

The following results for asymptotic tests for low rank dimension in model (**??**) were communicated in personal communication by J. Virta. Given a sample of $p_1 \times p_2$-dimensional matrices $(\boldsymbol{T}_1, \dots, \boldsymbol{T}_n)$, a limiting distribution was found for a test statistic based on $t_{\hat{q}_1}^2(\boldsymbol{S}_1)$ shown in the last section.

The following notations need to be introduced to give the asymptotic test statistic. The matrix $\hat{\boldsymbol{U}}_2$ consists of the last $p_1 - \hat{q}_1$ eigenvectors of $\boldsymbol{S}_1$, $\hat{\boldsymbol{\epsilon}}_i := (\hat{\boldsymbol{U}}_2)^T \left(\boldsymbol{T}_i - \bar{\boldsymbol{T}}\right)$ are therefore the last rows of the error matrices and $\boldsymbol{S}_\epsilon := \frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{\epsilon}}_i \hat{\boldsymbol{\epsilon}}_i^T$ is their estimated matrix covariance.

The limiting distribution of the asympotic test statistic under the null hypothesis $H_{0\hat{q}_1}$ from (**??**) is then given by

$$\frac{n(p_1 - \hat{q}_1) t_{\hat{q}_1}^2(\boldsymbol{S}_1)}{2\theta^2\psi} \to \chi^2_{\frac{1}{2}(p_1 - \hat{q}_1 - 1)(p_1 - \hat{q}_1 + 2)}$$

where θ^2 is the value of the last eigenvalues of $\boldsymbol{\Sigma}_1$, estimated by $\hat{\theta}^2 := \frac{1}{p_1 - \hat{q}_1} \sum_{i=\hat{q}_1+1}^{p_1} \lambda_i(\boldsymbol{S}_1)$ and ψ is the asymptotic variance of any off-diagonal value of $\boldsymbol{S}_1$, which can be estimated by

$$\hat{\psi} = \frac{1}{(p_1 - \hat{q}_1 - 1)(p_1 - \hat{q}_1 + 2)} \frac{1}{n} \sum_{i=1}^n \left\{ \operatorname{tr}\left[\left(\hat{\boldsymbol{\epsilon}}^T \boldsymbol{S}_\epsilon^{-1} \hat{\boldsymbol{\epsilon}}_i\right)^2\right] - \frac{1}{p_1 - \hat{q}_1} \left(\operatorname{tr}\left[\hat{\boldsymbol{\epsilon}}_i^T \boldsymbol{S}_\epsilon^{-1} \hat{\boldsymbol{\epsilon}}_i\right]\right)^2 \right\}$$

For normally distributed noise, the value $\psi = p_2^{-1}$ can be used instead. The test can also be applied on the transposed data to test for the low rank dimension $H_{0\hat{q}_2} : q_2 = \hat{q}_2$.

3.2 Generalization to higher-order tensors

3.2.1 Tensor model

For the random $q_1 \times q_2 \times \cdots \times q_k$-dimensional core tensor $\mathcal{X}$ with $\mathbb{E}[\mathcal{X}] = \mathbf{0}$, a set of orthogonal matrices $\boldsymbol{O}^{(i)} \in \mathfrak{O}_{q_i,p_i}$ with $q_i \leq p_i$ for $i = 1, \ldots, k$, a spherical random tensor $\mathcal{E}$ of dimension $p_1 \times p_2 \times \cdots \times p_k$, which is independent of $\mathcal{X}$ and the mean value tensor $\mathcal{M} \in \mathbb{R}^{p_1 \times p_2 \times \cdots \times p_k}$, we define the noisy tensor model

$$\mathcal{T} = \mathcal{M} + \mathcal{X} \times_1 \boldsymbol{O}^{(1)} \times \cdots \times_k \boldsymbol{O}^{(k)} + \mathcal{E}, \tag{3.7}$$

a tensorial version of (**??**). The m-mode covariance $\boldsymbol{\Sigma}_m := \operatorname{Cov}_m[\mathcal{E}]$ of the error tensor fulfills

$$\begin{aligned}
\operatorname{Cov}_m[\mathcal{E}] &= \operatorname{Cov}_m[\mathcal{E} \times_m \boldsymbol{O}] \\
&= \mathbb{E}\big[\big((\mathcal{E} \times_m \boldsymbol{O})_{(m)} - \mathbb{E}\left[(\mathcal{E} \times_m \boldsymbol{O})_{(m)}\right]\big) \\
&\qquad \big((\mathcal{E} \times_m \boldsymbol{O})_{(m)} - \mathbb{E}\left[(\mathcal{E} \times_m \boldsymbol{O})_{(m)}\right]\big)^T \big] \\
&= \boldsymbol{O}\ \mathbb{E}\left[(\mathcal{E}_m - \mathbb{E}[\mathcal{E}]_m)(\mathcal{E}_m - \mathbb{E}[\mathcal{E}]_m)^T\right]\ \boldsymbol{O}^T \quad \forall \boldsymbol{O} \in \mathfrak{O}_{p_m}
\end{aligned}$$

for each mode m, where the third equation follows from property (**??**). Therefore each $\boldsymbol{\Sigma}_m$ of the spherical tensor $\mathcal{E}$ is a multiple of a p_m-dimensional unit matrix $e_m \boldsymbol{I}_{p_m}$ with $e_m \in \mathbb{R}$. By the same property, the second equation in

$$\begin{aligned}
&\operatorname{Cov}_m\left[\mathcal{X} \times_1 \boldsymbol{O}^{(1)} \times \cdots \times_k \boldsymbol{O}^{(k)}\right] \\
&= \mathbb{E}\left[\left(\mathcal{X} \times_1 \boldsymbol{O}^{(1)} \times \cdots \times_k \boldsymbol{O}^{(k)}\right)_{(m)} \left(\mathcal{X} \times_1 \boldsymbol{O}^{(1)} \times \cdots \times_k \boldsymbol{O}^{(k)}\right)^T_{(m)}\right] \\
&= \boldsymbol{O}^{(m)} \mathbb{E}\Big[\boldsymbol{X}_{(m)} \left(\boldsymbol{O}^{(k)} \otimes \ldots \otimes \boldsymbol{O}^{(m+1)} \otimes \boldsymbol{O}^{(m-1)} \otimes \ldots \otimes \boldsymbol{O}^{(1)}\right) \cdot \\
&\quad \left(\boldsymbol{O}^{(k)} \otimes \ldots \otimes \boldsymbol{O}^{(m+1)} \otimes \boldsymbol{O}^{(m-1)} \otimes \ldots \otimes \boldsymbol{O}^{(1)}\right)^T \boldsymbol{X}^T_{(m)}\Big] (\boldsymbol{O}^{(m)})^T
\end{aligned}$$

follows, and we get, that this matrix is of rank q_m, so that last $p_m - q_m$ eigenvalues are zero. It follows from the independence of $\mathcal{E}$ to $\mathcal{X}$, that

$$\operatorname{Cov}_m[\mathcal{T}] = \operatorname{Cov}_m\left[\mathcal{X} \times_1 \boldsymbol{O}^{(1)} \times \cdots \times_k \boldsymbol{O}^{(k)}\right] + \operatorname{Cov}_m[\mathcal{E}].$$

Since $\operatorname{Cov}_m[\mathcal{E}]$ commutes, the eigenvalues of $\operatorname{Cov}_m[\mathcal{T}]$ are the sum of the eigenvalues of both summands and has the same form as (**??**). We can therefore use the variance of the last eigenvalues of each tensor covariance matrix to implement a low rank bootstrap test.

3.2.2 Bootstrap test for higher order tensors

In the following, we create a test for model (**??**) for the hypothesis $H_{0\hat{q}_m} : q_m = \hat{q}_m$ or

$$\begin{aligned}
H_{0\hat{q}_m} :& \lambda_1(\boldsymbol{\Sigma}_m) \geq \lambda_2(\boldsymbol{\Sigma}_m) \geq \cdots \geq \lambda_{\hat{q}_m}(\boldsymbol{\Sigma}_m) \\
&\geq \lambda_{\hat{q}_m+1}(\boldsymbol{\Sigma}_m) = \lambda_{\hat{q}_m+2}(\boldsymbol{\Sigma}_m) = \cdots = \lambda_{p_m}(\boldsymbol{\Sigma}_m),
\end{aligned}$$

Given a sample of tensors $(\mathcal{X}_1, \dots, \mathcal{X}_n)$, we first compute the sample m-mode covariance $\boldsymbol{S}_m$ and choose the variance of its last eigenvalues as the bootstrap statistic

$$T_{\hat{q}_m} = \frac{1}{\hat{d}} s^2(\lambda_{\hat{q}_m+1}(\boldsymbol{S}_m), \dots \dots, \lambda_{p_m}(\boldsymbol{S}_m))$$

where $\hat{d} = \frac{1}{p_m - \hat{q}_m} \sum_{i=\hat{q}_m+1}^{p_m} \lambda_i(\boldsymbol{S}_m)$ is again used for normalization.

We then compute the tPCA components $(\mathcal{Z}_1, \dots, \mathcal{Z}_n)$ from Section **??** without dimension reduction from the centered sample $(\mathcal{T}_i - \bar{\mathcal{T}}, \dots \mathcal{T}_n - \bar{\mathcal{T}})$, from which we draw M tensor bootstrap samples $\left(\mathcal{Z}_1^{(j)}, \dots, \mathcal{Z}_n^{(j)}\right)$. To achieve a spherical distribution in the error of the tested mode, we apply the three strategies from Sections **??** and **??** in a more general form. In the orthogonal bootstrap strategy, we compute

$$\mathcal{Z}_i^{*(j)} = \mathcal{Z}_i^{(j)} \times_m \operatorname{diag}\left(\boldsymbol{I}_{\hat{q}_m}, \boldsymbol{U}_i\right) \quad \text{with } \boldsymbol{U}_i \in \mathfrak{O}_{p_m - \hat{q}_m} \text{ for } i = 1, \dots, n$$

for each bootstrap sample. In the permutation bootstrap strategy, we use $(p_m - \hat{q}_m) \times (p_m - \hat{q}_m)$-dimensional permutation matrices $\boldsymbol{P}_i$ instead of $\boldsymbol{U}_i$ and in the parametric bootstrap, we replace all elements $(\mathcal{Z}_i^{(j)})_{i_1, i_2, \dots, i_k}$ with $i_j = 1, \dots, p_j$ for all $j \neq m$ and $i_m = \hat{q}_m + 1, \dots, p_m$ by normal distributed values with a variance of $\hat{d}$, which form a spherical subtensor. We then apply the backtransformation

$$\mathcal{T}_i^{(j)} = (\mathcal{Z}_i^{*(j)}) \times_1 \boldsymbol{U}^{(1)} \times \dots \times_k \boldsymbol{U}^{(k)} + \bar{\mathcal{T}}$$

with the matrices $\boldsymbol{U}^{(i)}$ returned from tPCA. The bootstrap test statistic is then computed from each bootstrap sample $(\mathcal{T}_1^{(j)}, \dots, \mathcal{T}_n^{(j)})$, $j = 1, \dots, M$ and compared with $T_{\hat{q}_m}$ from the original data. Algorithm **??** gives a pseudo code representation of the introduced methods.

```
Data: The data $(\mathcal{T}_1, \ldots, \mathcal{T}_n)$ consisting of $n$ elements of dimension
      $p_1 \times p_2 \times \cdots \times p_k$,
      $M$ = the number of bootstrap samples
      $m$ = the tested mode
      $strat$ = the name of the bootstrap strategy
      $\hat{q}_m$ = the low rank dimension of the null hypothesis
Result: The p-value of the test
begin
    Estimate the sample mean tensor $\bar{\mathcal{T}}$
    and the sample $m$-mode covariances $\boldsymbol{S}_j$ from $\mathcal{T}_1, \ldots, \mathcal{T}_n$ for $j = 1, \ldots, k$ ;
    Compute all eigendecompositions $\boldsymbol{S}_j = \boldsymbol{U}^{(j)} \boldsymbol{\Lambda}^{(j)} (\boldsymbol{U}^{(j)})^T$ for $j = 1, \ldots, k$;
    Compute the test statistic $T_{\hat{q}_m} = 1/\hat{d} \cdot s^2(\lambda_{\hat{q}_m+1}(\boldsymbol{S}_m), \ldots\ldots, \lambda_{p_m}(\boldsymbol{S}_m))$ ;
    Compute the tensors $\mathcal{Z}_i = \left(\mathcal{T}_i - \bar{\mathcal{T}}\right) \times_1 (\boldsymbol{U}^{(1)})^T \times \cdots \times_k (\boldsymbol{U}^{(k)})^T$ ;
    for $j = 1, \ldots, M$ do
        Take a bootstrap sample $(\mathcal{Z}_1^{(j)}, \ldots, \mathcal{Z}_n^{(j)})$ from $(\mathcal{Z}_1, \ldots, \mathcal{Z}_n)$ ;
        for $i \in \{1, \ldots, n\}$ do
            if $strat$ == "orthogonal" then
                $\mathcal{Z}_i^{(j)} \leftarrow \mathcal{Z}_i^{(j)} \times_m \operatorname{diag}(\boldsymbol{I}_{\hat{q}_m}, \boldsymbol{O}_i)$ with
                $\boldsymbol{O}_i$ randomly chosen from $\mathfrak{O}_{p_m - \hat{q}_m}$
            end
            if $strat$ == "permutation" then
                $\mathcal{Z}_i^{(j)} \leftarrow \mathcal{Z}_i^{(j)} \times_m \operatorname{diag}(\boldsymbol{I}_{\hat{q}_m}, \boldsymbol{P}_i)$ with $\boldsymbol{P}_i$ randomly chosen from the
                  group of $(p_m - \hat{q}_m)$-dimensional permutation matrices.
            end
            if $strat$ == "parametric" then
                Compute $\sigma^2 = \frac{1}{p_m - \hat{q}_m} \sum_{i=\hat{q}_m+1}^{p_m} \lambda_i(\boldsymbol{S}_m)$. ;
                Replace the values $(\mathcal{Z})_{i_1, \ldots, i_k}$ with $i_j = 1, \ldots, p_j$ for all $j \neq m$ and
                  $i_m = \hat{q}_m + 1, \ldots, p_m$ with $N(0, \sigma^2)$ distributed elements
            end
            Compute the backtransformation $\hat{\mathcal{T}}_i^{(j)} = \mathcal{Z}_i^{(j)} \times_1 \boldsymbol{U}^{(1)} \times \cdots \times_k \boldsymbol{U}^{(k)} + \bar{\mathcal{T}}$ ;
        end
        Compute the bootstrap test statistic $\hat{t}_j$ from $(\hat{\mathcal{T}}_1^{(j)}, \hat{\mathcal{T}}_2^{(j)}, \ldots, \hat{\mathcal{T}}_n^{(j)})$;
    end
    return the p-value $\frac{\#\{\hat{t}_j : \hat{t}_j > T_{\hat{q}_m}\} + 1}{M+1}$;
end
```

Algorithm 2: The bootstrap algorithm for tensorial signal dimension estimation

4 Numerical analysis

In this chapter we present the results of two simulation studies, in which the asymptotic and bootstrap tests for matrix valued data from Sections **??** and **??** were compared by their rejection probabilities and their ability to estimate the correct dimension of low rank data. The orthogonal, permutation and parametric bootstrap methods and the asymptotic test with both normality and non-normality assumption will be compared. In the last section, both asymptotic tests and the orthogonal bootstrap will be applied on a picture dataset.

4.1 Rejection probabilities of simulated data

In the first simulation study, samples of the model (**??**) with dimensions fixed to $q_1 = 3$, $q_2 = 4$, $p_1 = 7$, $p_2 = 8$ were simulated to estimate the rejection probabilities of the mentioned tests.

The matrix normal and the matrix t-distributions with degree of freedom $v = 5$ were used for generation of either data or error matrices in all four combinations. Both matrix distributions were set up, so they have the same variance in all components. In order to compare different proportions of data and error variances, a normalized variance $\sigma^2 = 1$ was set for the data generating distributions and varying values of $\sigma^2 = 0.5, 1, 1.5, 2$ were set for the error.

Random orthogonal matrices from the Stiefel manifolds $\mathfrak{O}_{q_1,p_1}$ and $\mathfrak{O}_{q_2,p_2}$ were generated from an uniform distribution w.r.t. the Haar measure and then truncated to the required size. Random permutation matrices were generated by reordering the columns of a unit matrix by a resampled index set. More implementation details about the model generation are given in the Appendix **??**.

All scenarios were simulated with sample sizes of $n = 1000, 2000, 4000$. For the generated data, tests with hypotheses $\hat{q}_1 = 2, 3, 4$ were applied on the first mode with real dimension $q_1 = 3$ and $M = 200$ resamples were used in the bootstrap tests. For all parameter combinations, the simulation was conducted $N = 1000$ times to achieve accurate results. No differences in the results were visible, when using either the normal or the t-matrix distribution as generators of the data matrices $\boldsymbol{X}_i$. We therefore only distinguish between the error distributions in the following.

Figure **??** shows the rejection probabilities for all of the mentioned scenarios in the case, where the error distribution is normal. When the correct dimension is tested, it can be seen, that the rejection probabilities for all tests except the permutation method are at the 0.05 level, which was chosen as threshold p-value. When setting the hypothesis to a lower dimension, it was correctly rejected in all cases.

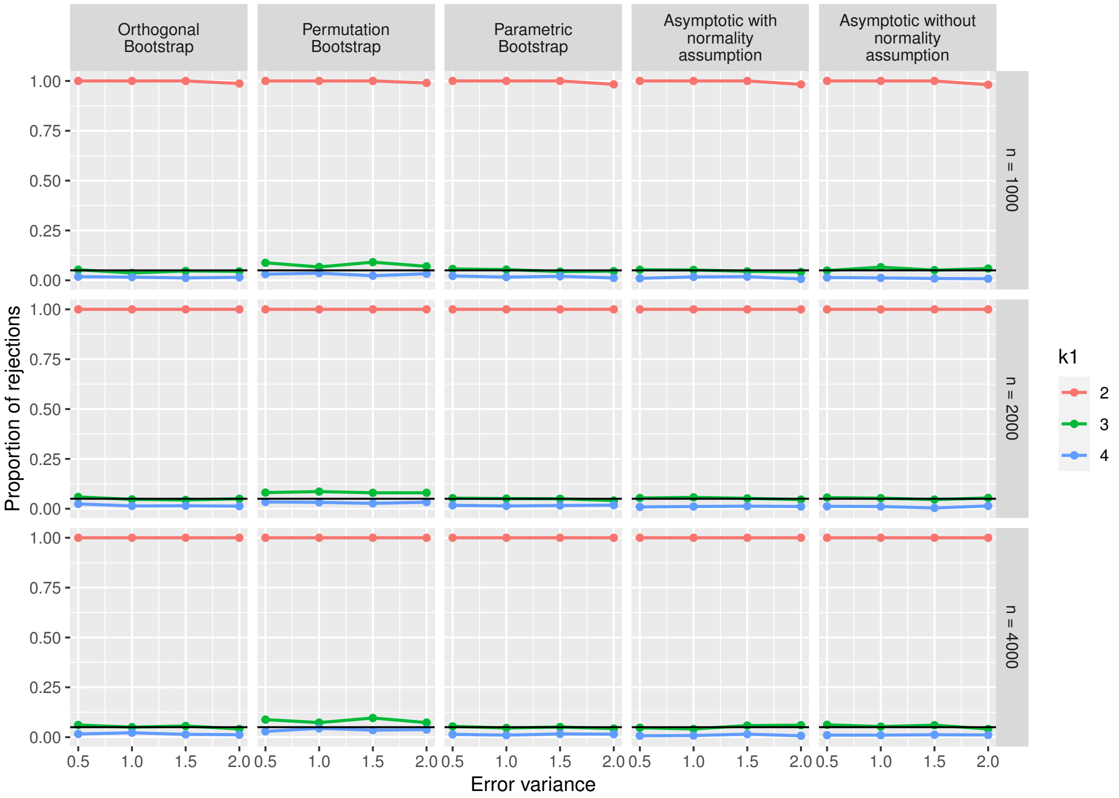

Figure 4.1: The rejection probabilities of the bootstrap and the asymptotic tests from Chapter **??**. The hypotheses $q_1 = 2, 3, 4$ are tested for simulated data of Model (**??**) with dimensions $q_1 = 3, q_2 = 4, p_1 = 7, p_2 = 8$. The error tensor is matrix-variate normal distributed with variances ranging from $\sigma^2 = 0.5, 1, 1.5, 2$ in each component and all tests were conducted with sample sizes of $n = 1000, 2000, 4000$.

Figure **??** shows the rejection probabilities, when the matrix-variate t is selected as error distribution. Expectedly, the parametric bootstrap method, where the error is substituted with normal distributed values and the asymptotic test, which also relies on normal distributed data, do not return meaningful results. For the other methods, larger variances in the error distributions lead to faster decreasing rejection probabilities in comparison to the normal error setups, when the hypothesis is set to a lower dimension than q_1. The rejection probability, when testing for the correct dimension, is at the correct 0.05 level for the asymptotic test without normality assumption, and slightly above in the orthogonal and parametric bootstrap tests.

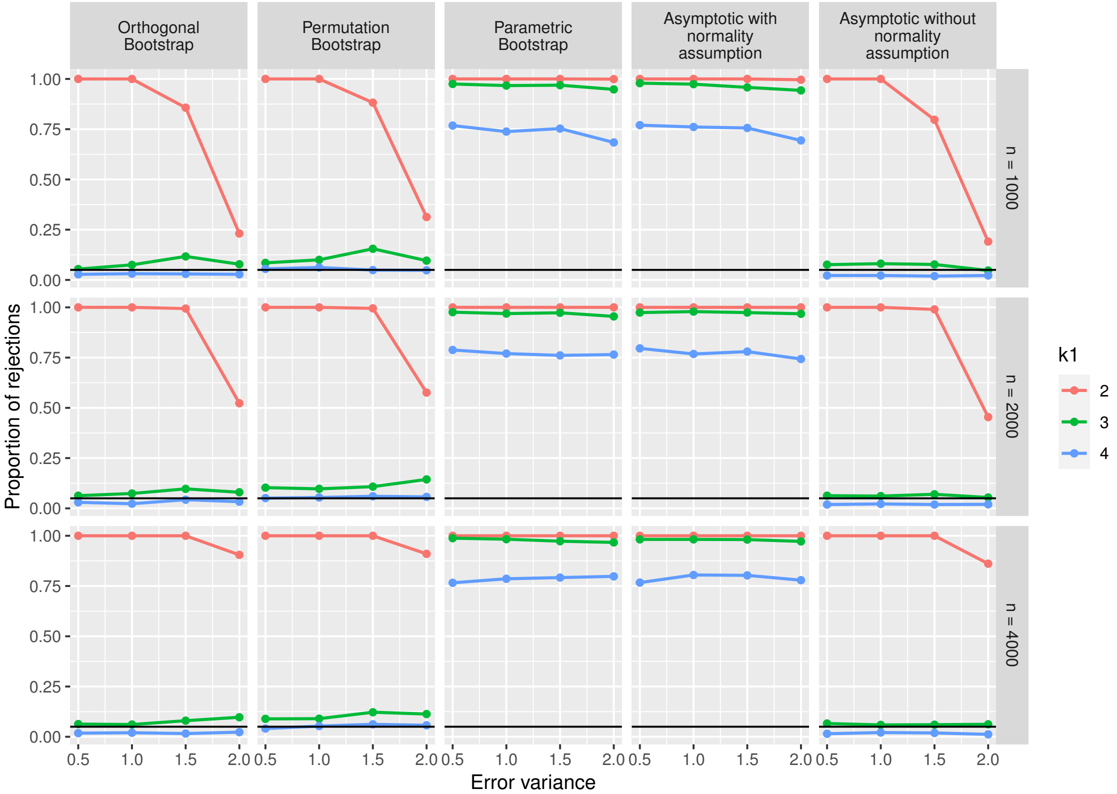

Figure 4.2: The rejection probabilities of the bootstrap and the asymptotic tests from Chapter **??**. The hypotheses $q_1 = 2, 3, 4$ are tested for simulated data of Model (**??**) with dimensions $q_1 = 3, q_2 = 4, p_1 = 7, p_2 = 8$. The error tensor is matrix-variate t-distributed with variances ranging from $\sigma^2 = 0.5, 1, 1.5, 2$ in each component and all tests were conducted with sample sizes of $n = 1000, 2000, 4000$.

4.2 Dimension estimation of simulated data

Both dimension tests can iteratively be used for the estimation of the low rank dimension in a divide and conquer procedure. For a given mode $i \in \{1, 2\}$, this is realized by first setting up an upper bound $b \leftarrow p_i$ and a lower bound $a \leftarrow 0$ and then applying the dimension test with $H_{0\hat{q}_i}$ where $\hat{q}_i = \lfloor (a + b)/2 \rfloor$ with $\lfloor \cdot \rfloor = \max \{n \in \mathbb{N} : n \leq x\}$. Depending on whether the null hypothesis is accepted or rejected, either a new upper bound $b \leftarrow \lfloor (a + b)/2 \rfloor$ or a new lower bound $a \leftarrow \lfloor (a + b)/2 \rfloor + 1$ is set. With the newly set bounds, this procedure is repeated until a and b are equal.

Since this method is computationally very expensive, the orthogonal bootstrap with $M = 200$ resamples and the asymptotic test with no normality assumption were selected from the methods compared in the last section, as they have performed well in all scenarios. In model (**??**), the dimensions $q_1 = 4$, $q_2 = 5$, $p_1 = 25$, $p_2 = 30$ were fixed with a sample size of $n = 4000$. The matrix-variate normal and t-distributions were used for data and error generation as described in the last section, but with error variances ranging from $\sigma_1^2 = 0.2$ to $\sigma_{10}^2 = 3.0$ in steps of equal size.

The divide and conquer approach was applied on the first mode, with $N = 1000$ repetitions. The resulting dimension estimates are shown in a percent stacked barchart in Figure **??**. In the case of normal error distribution, the correct low rank dimension could be reliable retrieved up to a variance of $\sigma^2 = 1.13$.

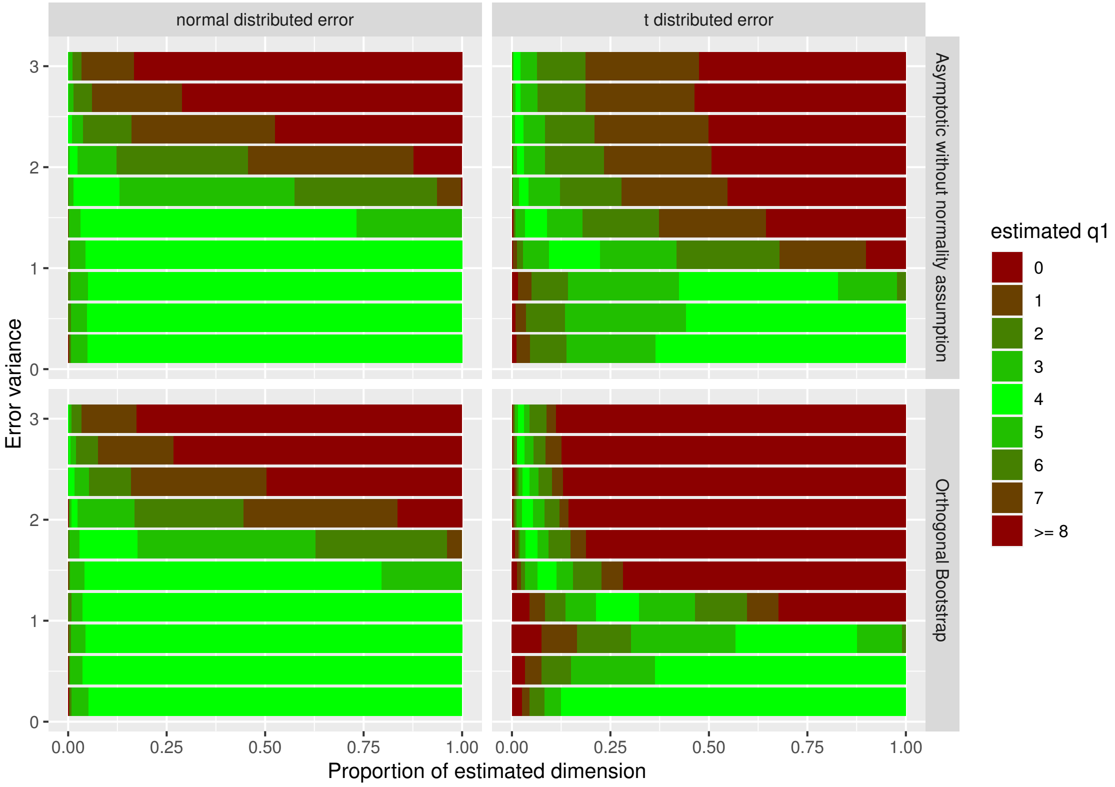

Figure 4.3: The proportions of the estimated low rank dimension of the first mode, computed by the asymptotic test without normality assumptions and the orthogonal bootstrap test using a divide and conquer approach. The dimensions of the simulated data of Model (**??**) are $q_1 = 4, q_2 = 5$ and $p_1 = 25, p_2 = 30$. Matrix-variate normal and t-distributions were both used in the simulation

For matrix t-distributed error, the estimates were only reliable for the bootstrap method with the lowest error variance. However, the percentage of correctly estimated dimension was slightly higher in the asymptotic test when the variance is slightly below 1. For larger error variance, wrong estimates were in almost all cases below the actual dimension, meaning that the hypothesis was often not rejected during the process.

4.3 Dimension estimation of a picture dataset

Both asymptotic and the orthogonal dimension tests were applied with the same divide and conquer approach as shown in Section **??** on the fingers dataset from [**?**], which contains 128 x 128 grayscale images of left and right hands with 0-5 fingers extended. For each group, there are 1500 pictures available and the two groups of left hands with no fingers extended (L0) and right hands with all fingers extended (R5) were chosen for closer analysis.

The extraction algorithm encoded the greyscale pixels in values between 0 and 1, which led to datasets consisting of $\boldsymbol{X}_i \in [0,1]^{128\times128}$, $i = 1, \ldots, 1500$, for both groups. All three tests computed the same low rank dimensions of 101 × 95 for L0 and 114 × 119 for R5. It is likely, that some of the differences between the two groups are caused by the smaller silhouette of the hands when no fingers are extended.

In Figure **??** and **??**, five pictures from each of the two groups are plotted respectively and compared with the pictures resulting from the reduced tensor components. In all cases, there were no visible differences to the original pictures. Very low rank tensor reconstructions were shown in the third row, where 20 × 20 components were used for comparison

A dimension reduction of less than 30 in both modes does not seem large. We attribute this to the fact, that the sphericity assumption on random matrix error is strong and probably does not apply very well to the chosen dataset. It is also possible, that methods based on eigenvalues alone do not have enough information to give a clear low rank dimension. This is supported by the scree plots based on the mode-covariances of both groups, which are shown in Figure **??**. These plots do not show an “elbow”, which can be usually seen as an indicator for a low rank dimension.

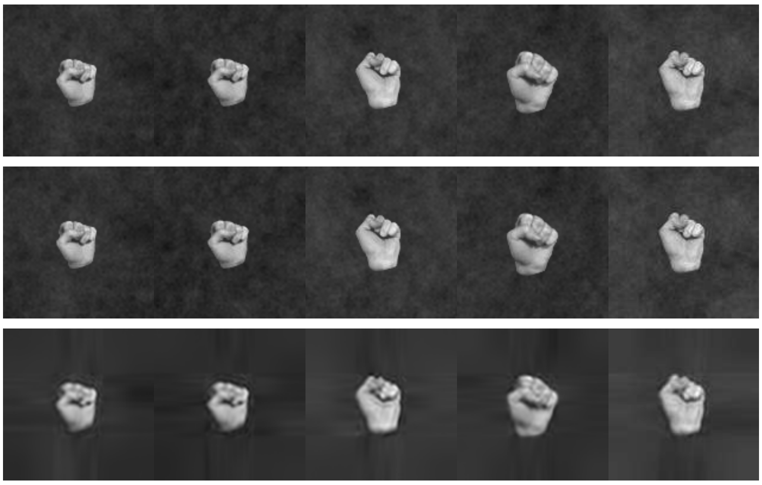

Figure 4.4: The first row shows five randomly chosen pictures from L0. In the second and third row, we see the same pictures after reconstruction from 101x95 and 20x20 components.

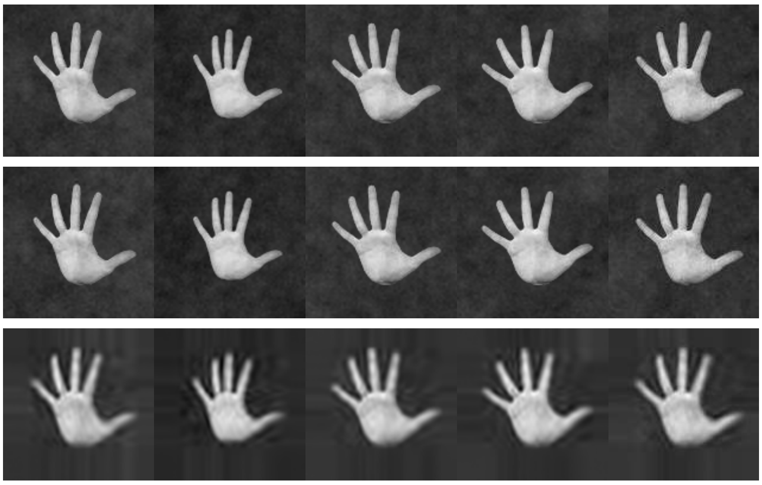

Figure 4.5: The first row shows five randomly chosen pictures from R5. In the second and third row, we see the same pictures after reconstruction from 114x119 and 20x20 components.

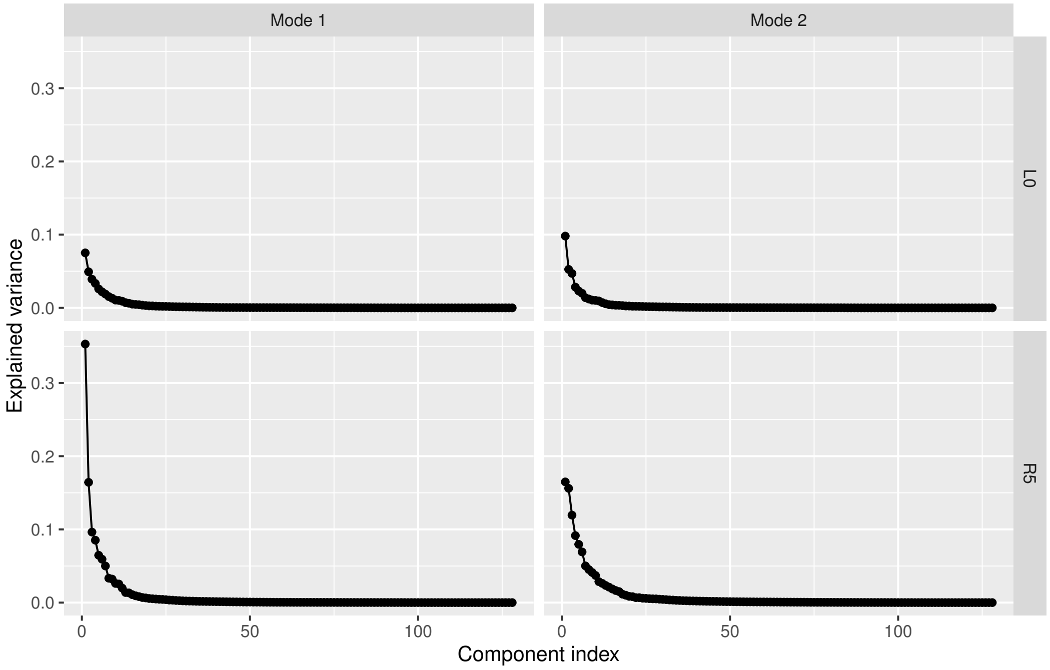

Figure 4.6: The scree plots for the row PCs (left) and column PCs (right) of the groups L0 and R5 from the fingers dataset

5 Summary and future work

In this thesis, bootstrap tests for the low rank dimensions of latent matrices and tensors in noisy models were developed, which can be applied per row or column for matrices or mode-wise for higher order tensors. These tests are based upon a bootstrap method for a vector model, which assumes the random error vector to have a spherical distribution. Three versions of bootstrap tests were given, namely the orthogonal, permutation and parametric bootstrap, which differ only in the bootstrap transformation step. The matrix versions of these tests were compared by their rejection probabilites with an existing test based on asymptotic distributions, which works under similar assumptions. For this, data was simulated according to the model with different error distributions, error variances and sample sizes. The orthogonal bootstrap and the asymptotic test performed best in all scenarios and were then also additionally compared by their ability to estimate the true low rank dimension in a divide and conquer approach. Here the asymptotic test gave slightly better results, when the error was not normal distributed.

The usefulness of such tests is mainly defined by its possible applications and is therefore largely dependent on the model and its assumptions. One of the main model assumptions in this thesis is the sphericity of the random error. This can be already considered strong in the case of spherical random vectors, since they can be defined by a single positive random variable. In the tensor model, the usage of a more general version of sphericity was necessary, which defines a spherical random tensor to have the same distribution as when multiplied by an orthogonal matrix in any mode. However, such definition was not found in literature and it is an open question, whether such distributions could play a significant role in real-world applications.

We saw in Chapter **??**, that the asymptotic and bootstrap tests were able to compute the correct low rank dimension in simulated data conforming to the model assumptions, when the variance of the error tensor components was below the variance of the data components. It was not possible to find real-world data, which was clearly fullfilling these assumptions. Application of the divide and conquer algorithm to the hands dataset in Section **??** resulted in very large number of components in both modes. This does not seem fitting, since the pictures of a single group differ only minimally and is probably caused by not fulfilled assumptions.

Both simulated and real-world data were chosen to be samples of matrices and only the matrix versions of the bootstrap tests were implemented and applied. The main reason for this was the computational effort, that the tests already caused when applied to matrix-valued data, especially when used in a divide and conquer approach. The simulations in Chapter **??**, in which the rejection probabilites and dimension estimates were computed, took several weeks on 50 CPU cores. The application of the divide and conquer algorithm for a single mode in one group of the hands pictures dataset took 6 hours on a single core.

It is likely that there lies optimization potential in the current implementations given in Appendix **??**. Also, the performance can be heavily improved by porting functions to a lower level programming language such as C or C++. Under such measures, the generalization of the bootstrap step and therefore the test could give reasonable times for tensorial data.

In general, the increase of computational power over time allows more and more sophisticated algorithms to be applied on tensor valued data. Many dimension reduction methods for tensor models are based on numerical and algebraic methods and do not take a probabilistic point of view. While there are probably little applications for tests with the model assumptions used in this thesis, the further research of different random tensor models has a lot of potential.

A Appendix: Implementation detail

In the following, code samples of the most important functions and algorithms developed for this thesis are presented. The programming language R [?] was used exclusively for all implementations.

The test methods assume the parameter x to contain the sample of tensors, where single elements can be selected by the last index of the array as described in Section **??**. The mode parameter sets the mode, on which the test shall be applied and k is the dimension of the hypothesis.

A.1 Data Simulation

The method used for generating the data in the simulations of Section **??** and **??** is given in Listing **??**. The four cases of data distribution and the error distribution being either normal or t-distributed can be seen in the if conditions. The method rmatrixt from package matrixsampling [?] was used for generating matrix-variate t-distributed data and the method rorth from package ICtest [?] for generation of the orthogonal transformation matrices. The low and high rank dimension are given by the parameters d and p, which both expect two dimensional vectors and the sample size is given by n.

```r
sim_model <- function(d, p, n, data_dist = "norm",
                      err_dist = "norm", err_sd = 1.0) {
  # data distribution
  if(data_dist == "norm") {
    z <- rnorm(prod(d)*n, mean = 0, sd = 1)
    dim(z) <- c(d, n)
  } else if(data_dist == "t") {
    sig2 <- 3
    UUU <- diag(sig2, ncol=d[1], nrow=d[1])
    z <- rmatrixt(n, 5, matrix(0,d[1],d[2]), UUU, diag(d[2]))
  } else {
    error("the distribution speficied is not implemented")
  }

  # error distribution
  if(err_dist == "norm") {
    err <- rnorm(prod(p)*n, mean = 0, sd  = err_sd)
    dim(err) <- c(p, n)
  } else if(err_dist == "t") {
    sig2 <- 3*err_sd^2
    UUU <- diag(sig2, ncol=p[1], nrow=p[1])
    err <- rmatrixt(n, 5, matrix(0,p[1],p[2]), UUU, diag(p[2]))
  } else {
    error("the distribution speficied is not implemented")
  }

  U1 <- ICtest::rorth(p[1])[, 1:d[1]]
  U2 <- ICtest::rorth(p[2])[, 1:d[2]]
  x <- tensorBSS::tensorTransform2(z, list(U1 = U1, U2 = U2), 1:2)

  # adds the error to the transformed data tensor
  x <- x + err

  return(list(z = z,
              x = x,
              err = err,
              U = list(U1, U2)))
}
```

Listing A.1: Data generation method for simulation study

A.2 Bootstrap test

In Listing **??**, **??** and **??**, the implementations of the bootstrap test statistic, the bootstrap transformation and the test itself are given. While the test method and the test statistic were implemented, so that they could possibly support higher order tensors, the transformation was only implemented for matrices, therefore rendering the test also only applicable to second order tensors. Methods of the R package tensorBSS [**?**] are used for computation of the m-mode covariances, m-mode multiplication and tensorial PCA.

```r
dimension_test_statistic <- function(x, mode, k) {
  mCov <- tensorBSS::mModeCovariance(x, mode, center = TRUE)
  mEig <- eigen(mCov, symmetric = TRUE)
  D <- mEig$values
  return(var(D[(k+1):length(D)]) / mean(D[(k+1):length(D)]))
}
```

Listing A.2: Test statistic of the bootstrap test

```r
bootstrap_transform <- function(x, mode, k, method = "orth") {
  x_tpca <- tensorBSS::tPCA(x = x)

  S <- x_tpca$S
  S_save <- S

  p <- dim(S)[-3]
  n <- dim(S)[3]


  if (method == "param") {
    S[(k + 1):p[1], , ] <-
      array(data = rnorm(
        n = (p[1] - k) * p[2] * n,
        mean = 0,
        sd = sqrt(mean(x_tpca$D[[1]][(k + 1):p[1]]))
      ),
      dim = c(p[1] - k, p[2], n))

  } else if (method == "orth") {
    for (i in 1:n) {
      S[(k + 1):p[1], , i] <-
        ICtest::rorth(p[1] - k) %*% S[(k + 1):p[1], , i]
    }
  } else if (method == "perm") {
    for (i in 1:n) {
      S[(k + 1):p[1], , i] <- S[sample((k + 1):p[1], replace = FALSE), , i]
    }
  } else{
    stop("the specified method is not available")
  }

  x_new <- tensorBSS::tensorTransform2(S, x_tpca$U)

  if(mode != 1){
    x_new <- aperm(x_new, c(2,1,3))
  }
  return(x_new)
}
```

Listing A.3: Data transformation step of the bootstrap test for second order tensors

```
bootstrap_test <- function(x, tmode, k, bootstrap_method, m) {
  tstat <- dimension_test_statistic(x, tmode, k)
  bootstrap_statistics_1 <- rep(0, m)

  for(i in 1:m) {
      y <- bootstrap_transform(x, tmode, k, bootstrap_method)
      bootstrap_statistics_1[i] <- dimension_test_statistic(y, tmode, k)
  }
  pval <- sum(bootstrap_statistics_1 > tstat) / m
  return(pval)
}
```

Listing A.4: The bootstrap test for second order tensors

A.3 Asymptotic test

The function for computing ψ from the asymptotic test in Section **??**, which is needed when normality is not assumed, is given in Listing **??**. The implementations for the test statistic and the general test method with and without normality assumption are given in Listing **??** and **??**. The test statistic is implemented for the first mode of a matrix, which means, that matrices need to be transposed beforehand depending on the mode given, as seen in Line 7 to 9 of Listing **??**.

```
calc_psi <- function(x, r1, cov_eigen) {
  tr <- function(m) {
    return(sum(diag(m)))
  }

  p1 <- dim(x)[1]
  p2 <- dim(x)[2]
  n <- dim(x)[3]

  U1 <- cov_eigen$vectors[,(p1-r1+1):p1]
  eps <- array(data = 0, dim = c(r1, p2, n))
  sigma_eps <- array(data = 0, dim = c(r1, r1))

  y <- center_data(x)
  for(i in 1:n) {
    eps[,,i] <- crossprod(U1, y[,,i])
    sigma_eps <- sigma_eps + tcrossprod(eps[,,i])
  }
  sigma_eps <- 1/n * sigma_eps

  sigma_eps_inv <- solve(sigma_eps)
  psi <- 0
  for(i in 1:(dim(eps)[3])) {
    mat <- crossprod(eps[,,i], sigma_eps_inv %*% eps[,,i])
    psi <- psi + tr(mat %*% mat)
    psi <- psi - 1/r1 * (tr(mat))^2
  }
  psi <- 1/((r1 - 1)*(r1 + 2)) * 1/n * psi
  return(psi)
}
```

Listing A.5: Method for computing ψ for the asymptotic test without normality assumption

```
analytical_tstat <- function(x, k1, normal_distributed) {
  p1 <- dim(x)[1]
  p2 <- dim(x)[2]
  n <- dim(x)[3]
  r1 <- p1 - k1
  x_cov <- matrix_cov(x)
  cov_eigen <- eigen(x_cov)


  psi <- 0
  if(normal_distributed) {
    psi <- 1/p2
  } else {
    psi <- calc_psi(x, r1 = r1, cov_eigen = cov_eigen)
  }

  theta <- mean(cov_eigen$values[(p1-r1+1):p1])
  sk1 <- var(cov_eigen$values[(p1-r1+1):p1])*(r1 - 1)/r1

  test_statistic <- n * r1 * sk1 / (2 * theta^2 * psi)
  return(test_statistic)
}
```

Listing A.6: Computation of the analytical test statistic

```
asymptotic_test <- function(x, k, tmode = 1, normal_distributed = FALSE){
  if(! tmode %in% c(1,2)){
    stop("can only transform on first or second mode")
  }
  p <- dim(x)[tmode]
  r <- p - k
  if(tmode == 2){
    x <- aperm(x, c(2,1,3))
  }

  test_statistic <- analytical_tstat(x, k, normal_distributed)
  pvalue <- pchisq(
      abs(test_statistic),
      df = 1 / 2 * (r - 1) * (r + 2),
      ncp = 0,
      lower.tail = FALSE
    )
  return(pvalue)
}
```

Listing A.7: Asymptotic test by J. Virta

A.4 Divide and Conquer

Listing ?? shows the divide and conquer implementation that makes use of the bootstrap test. The asymptotic counterpart only differs in line 1 and 10, because it has a boolean parameter for the normality assumption instead of the method parameter.

```
 1 divide_and_conquer_bootstrap <- function(x, tmode = 1,
 2     method = "orth", m = 100) {
 3   p <- dim(x)
 4   lower <- 0
 5   upper <- p[1]
 6   it <- 1
 7
 8   while(lower != upper) {
 9     k1 <-  floor((lower + upper)/2)
10     pval <- bootstrap_test(x, tmode, k1, method, m)
11
12     if(pval < 0.05) {
13       lower <- k1+1
14     } else {
15       upper <- k1
16     }
17     it <- it + 1
18     if(it == p[1]) {
19       break
20     }
21   }
22   return(list(
23     it = it,
24     p1_est = ifelse(lower == upper, yes = lower, no = 0)
```

```
  ))
}
```

Listing A.8: Divide and conquer algorithm with bootstrap test

List of Figures

Listings

Bibliography

[CC70] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition. *Psychometrika*, 35(3):283–319, 1970.

[DLDMV00] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[Har70] Richard A Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 1970.

[Hof11] Peter D Hoff. Separable covariance arrays via the Tucker product, with applications to multivariate relational data. *Bayesian Analysis*, 6(2):179–196, 2011.

[Hot33] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[HWTH12] Hung Hung, Peishien Wu, Iping Tu, and Suyun Huang. On multilinear principal component analysis of order-two tensors. *Biometrika*, 99(3):569–583, 2012.

[JC16] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[JM92] Ian T Joliffe and Byron JT Morgan. Principal component analysis and exploratory factor analysis. *Statistical Methods in Medical Research*, 1(1):69–95, 1992.

[Joh90] Hastad Johan. Tensor rank is NP-complete. *Journal of Algorithms*, 4(11):644–654, 1990.

[KB09] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[Kor19] Pavel Koryakin. Fingers dataset. `https://www.kaggle.com/koryakinp/fingers`, 2019. [Online; accessed 01-04-2023].

[Lan12] Serge Lang. *Algebra*, volume 211. Springer Science & Business Media, 2012.

[Lau19] Stéphane Laurent. *matrixsampling: Simulations of Matrix Variate Distributions*, 2019. R package version 2.0.0.

[LPV06] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos.

Multilinear principal component analysis of tensor objects for recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 776–779. IEEE, 2006.

[Min00] Thomas P Minka. Automatic choice of dimensionality for PCA. In *Advances in Neural Information Processing Systems*, volume 13, pages 598–604, 2000.

[NOT22] Klaus Nordhausen, Hannu Oja, and David E Tyler. Asymptotic and bootstrap tests for subspace dimension. *Journal of Multivariate Analysis*, 188:104830, 2022.

[NOTV22] Klaus Nordhausen, Hannu Oja, David E Tyler, and Joni Virta. *ICtest: Estimating and Testing the Number of Interesting Components in Linear Dimension Reduction*, 2022. R package version 0.3-5.

[Pea01] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[Phi89] Peter CB Phillips. Spherical matrix distributions and Cauchy quotients. *Statistics & Probability Letters*, 8(1):51–53, 1989.

[R C22] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022.

[TB99] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[Tuc66] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[VKL$^+$21] Joni Virta, Christoph L Kösner, Bing Li, Klaus Nordhausen, Hannu Oja, and Una Radojicic. *tensorBSS: Blind Source Separation Methods for Tensor-Valued Observations*, 2021. R package version 0.3.8.

[VN19] Joni Virta and Klaus Nordhausen. Estimating the number of signals using principal component analysis. *Stat*, 8(1):e231, 2019.

[VTN16] Joni Virta, Sara Taskinen, and Klaus Nordhausen. Applying fully tensorial ICA to fMRI data. In *2016 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6. IEEE, 2016.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2:37–52, 1987.

[YZFY04] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[ZZ05] Daoqiang Zhang and Zhi-Hua Zhou. (2D)^{2}PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing*, 69(1-3):224–231, 2005.