Principal component analysis based on spatial signs

Zwaan, F.C.

F.C. Zwaan

Principal component analysis based on spatial signs

Bachelor thesis

July 15, 2023

Thesis supervisor: dr. A. Dürre

Leiden University
Mathematical Institute

Contents

1 Introduction

Principal component analysis is a technique where the data is changed to a new set of variables, the principal components. The directions of the principal components are the eigenvectors of the covariance matrix. Principal component analysis could for example be used for dimension reduction of a data set or to analyze a high-dimensional data set [1], [2].

Firstly principal component analysis will be explained and then principal component analysis based on spatial signs will be explained. The reason to look at a different variant of principal component analysis is that the spatial sign covariance matrix is used, which works well with outliers [3]. Hence we would also expect principal component analysis based on spatial signs to work well with outliers.

Next, an elliptical distribution is explained. Then some asymptotic properties of the principal components, for an elliptical distribution, are given. This is first done for the classical case and then for the spatial sign case. For the classical case, all asymptotic results were already known and proven in [4]. For the spatial sign case, there is only an asymptotic result for the spatial sign covariance matrix, but there aren't any asymptotic properties known about the eigenvalues and eigenvectors of the spatial sign covariance matrix. So the goal is to find some asymptotic results for this. The next goal is to find an expression for the MSE of the principal components for both cases. The MSEs are then compared, to get an idea of which estimator is more precise. For dimension 2 some explicit results are found. For dimensions 3 and 4 the MSEs are compared numerically.

Lastly, confidence ellipsoids are used to simulate how well the convergence of the eigenvectors works and an example of how confidence ellipsoids can be used to analyze a data set is given.

2 Principal component analysis

In this chapter principal component analysis will be explained. The definitions and all the things that are explained about principal components in this chapter are based on chapters 3 and 9 of [1]. For more information about principal component analysis see chapter 9 of [1].
From now on we will assume that for a random p-variate vector X, the first and second moments are finite. So this means that $\mathbb{E}(X) < \infty$ and $\text{var}(X) < \infty$.

Let $X = (X_1, ..., X_p)^T$ be a random centered vector, so $\mathbb{E}(X) = 0$. Then for the direction of the first principal component we want to find a γ_1 such that

$$\gamma_1 = \underset{\{\gamma_1 : ||\gamma_1||=1\}}{\arg\max} \text{var}(\gamma_1^T X).$$

For $i = 2, .., p$ we want to find a γ_i such that

$$\gamma_i = \underset{\{\gamma_i : ||\gamma_i||=1, \gamma_1^T\gamma_i=...=\gamma_{i-1}^T\gamma_i=0\}}{\arg\max} \text{var}(\gamma_i^T X).$$

This is equivalent to taking γ_i as the eigenvector that belongs to the eigenvalue λ_i of the covariance matrix $\Sigma = \mathbb{E}[(X - \mathbb{E}(X))(X - \mathbb{E}(X))^T]$, such that $\lambda_1 \geq \cdots \geq \lambda_i \geq \cdots \geq \lambda_p$. Now this gives us the following definition of principal components.

From now on we will assume that all eigenvalues of the covariance matrix are distinct. Otherwise the following definition will not be unique.

Definition 2.1. *Let $X = (X_1, ..., X_p)^T$ be a random vector with covariance matrix Σ. Write $\Sigma = G\Lambda G^T$ as an eigenvalue decomposition, with $\lambda_1 > \lambda_2 > \cdots > \lambda_p \geq 0$ on the diagonal of Λ and G orthonormal with $g_{ii} \geq 0$, for $i = 1, ..., p$. Then the **k-th principal component** is defined as $y_k = g_k^T(X - \mathbb{E}(X))$.*

The principal components could also be computed for a $n \times p$ data matrix.

$$\mathcal{X} = (X_1, ..., X_n)^T = \begin{pmatrix} X_1^{(1)} & \cdots & X_1^{(p)} \\ \vdots & \ddots & \vdots \\ X_n^{(1)} & \cdots & X_n^{(p)} \end{pmatrix}.$$

Here n is the number of observations and p is the number of variables. First the expected value and the covariance matrix need to be estimated, before the principal components can be computed

In [4] it is shown that the maximum likelihood estimator of the mean vector μ of a multivariate normal distribution $N_p(\mu, \Sigma)$ is the sample mean and that it is an unbiased estimator. The sample mean will be used to estimate the expected value and is defined below.

Definition 2.2. *Let $\mathcal{X}$ be a $n \times p$ data matrix. The **sample mean** of $\mathcal{X}$ is*

$$\bar{X} = (\bar{X}^{(1)}, ..., \bar{X}^{(p)})^T = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^n X_i^{(1)} \\ \vdots \\ \frac{1}{n}\sum_{i=1}^n X_i^{(p)} \end{pmatrix}.$$

In [4] it is shown that the maximum likelihood estimator of the covariance matrix Σ of the multivariate normal distribution $N_p(\mu, \Sigma)$ is $\frac{1}{n}\sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})^T$. This estimator is biased, but by multiplying it by $n/(n-1)$ it is unbiased. This is called the sample covariance matrix and will be used to estimate the covariance matrix. The sample covariance matrix is defined below.

Definition 2.3. *Let $\mathcal{X}$ be a $n \times p$ data matrix. The **sample covariance matrix (SCM)** of $\mathcal{X}$ is*

$$\hat{\Sigma} = \frac{1}{n-1}\sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})^T =$$

$$\begin{pmatrix} \frac{1}{n-1}\sum_{i=1}^{n}(X_i^{(1)}-\bar{X}^{(1)})^2 & \cdots & \frac{1}{n-1}\sum_{i=1}^{n}(X_i^{(1)}-\bar{X}^{(1)})(X_i^{(p)}-\bar{X}^{(p)}) \\ \vdots & \ddots & \vdots \\ \frac{1}{n-1}\sum_{i=1}^{n}(X_i^{(1)}-\bar{X}^{(1)})(X_i^{(p)}-\bar{X}^{(p)}) & \cdots & \frac{1}{n-1}\sum_{i=1}^{n}(X_i^{(p)}-\bar{X}^{(p)})^2 \end{pmatrix}$$

Using these two estimators, gives us the following definition of principal components for a $n \times p$ data matrix. We will assume that the sample covariance matrix has distinct eigenvalues, otherwise the following definition is not unique.

Definition 2.4. *Let $\mathcal{X}$ be a $n \times p$ data matrix. Write $\hat{\Sigma} = \hat{G}\hat{\Lambda}\hat{G}^T$ as an eigenvalue decomposition of the sample covariance matrix of $\mathcal{X}$, with $\hat{\lambda}_1 > \hat{\lambda}_2 > \cdots > \hat{\lambda}_p \geq 0$ on the diagonal of $\hat{\Lambda}$ and $\hat{G}$ orthonormal with $\hat{g}_{ii} \geq 0$, for $i = 1, ..., p$. Then the **k-th principal component** is defined as $\hat{y}_k = (\mathcal{X} - 1_n\bar{X}^T)\hat{g}_k$, where 1_n is a column vector of ones, with n entries.*

3 Principal component analysis based on spatial signs

In this chapter principal component analysis based on spatial signs will be explained. The definitions and results in this chapter are based on [5].
A spatial sign maps a vector to itself divided by its norm, except zero which is mapped to itself. This is helpful when there are a lot of outliers, because then the effect of the outliers is bounded [3].

Definition 3.1. *Let $x \in \mathbb{R}^p$. The **spatial sign** of x is*

$$S(x) = \begin{cases} \frac{x}{||x||_2}, & \text{if } x \neq 0 \\ 0, & \text{if } x = 0 \end{cases}$$

Here $||.||_2$ denotes the euclidean norm.

For principal component analysis based on spatial signs another covariance matrix is needed to compute the principal components. This new covariance matrix is constructed by first centering a vector in a suitable way, taking the spatial sign and then computing the covariance. A suitable way to center it is by using the population spatial median, which is defined below [3].

Definition 3.2. *Let $X = (X_1, ..., X_p)^T$ be a random vector. The **population spatial median** of X is*

$$\mu_s = \arg\min_{\mu \in \mathbb{R}^p} \mathbb{E}(||X - \mu||_2 - ||X||_2).$$

If the first moments exist then the population spatial median can be written as [3]

$$\mu_s = \arg\min_{\mu \in \mathbb{R}^p} \mathbb{E}||X - \mu||_2.$$

Now by first centering the vector with the population spatial median, then taking the spatial sign of that and then computing the covariance gives the population spatial sign covariance matrix, which is defined below [3].

Definition 3.3. *Let $X = (X_1, ..., X_p)^T$ be a random vector. The **population spatial sign covariance matrix (population SSCM)** of X is*

$$\Sigma_s = \mathbb{E}[S(X - \mu_s)S(X - \mu_s)^T],$$

where μ_s is the population spatial median.

This could also be done for a $n \times p$ data matrix. Then the population spatial sign covariance matrix and the population spatial median need to be estimated. The population spatial sign covariance matrix will be estimated in almost the same way as the covariance matrix. The population spatial median will be estimated with the spatial median, which is defined below [3].

Definition 3.4. *Let $\mathcal{X}$ be a $n \times p$ data matrix. The **spatial median** of $\mathcal{X}$ is*

$$\hat{\mu}_s = \arg\min_{\mu \in \mathbb{R}^p} \sum_{i=1}^{n} ||X_i - \mu||_2.$$

Now by estimating the population SSCM in a similar way as the covariance matrix is estimated, the sample spatial sign covariance matrix follows, which is defined below.

Definition 3.5. *Let $\mathcal{X}$ be a $n \times p$ data matrix. Then the **sample spatial sign covariance matrix (sample SSCM)** of $\mathcal{X}$ is*

$$\hat{\Sigma}_s = \frac{1}{n-1} \sum_{i=1}^{n} S(X_i - \hat{\mu}_s)S(X_i - \hat{\mu}_s)^T,$$

where $\hat{\mu}_s$ is the spatial median.

For the principal components based on spatial signs, the population SSCM and the population spatial median will be used to compute the principal components. The spatial median and the sample SSCM are used to compute the principal components based on spatial signs for a data matrix. For example in [6], [7] principal components based on spatial signs is used and explained.

4 Asymptotic properties of the principal components

In this chapter some asymptotic properties of the principal components for an elliptical distribution will be given. This will be done for the classical principal components and the principal components based on spatial signs.

4.1 Elliptical distribution

For an elliptical distribution there are some nice asymptotic properties for the principal components and some nice results for the eigenvalues and eigenvectors of the population spatial sign covariance matrix. Before we look at that, an elliptical distribution will be explained and two examples will be given. The definitions and examples are based on chapter 2 of [8].

Definition 4.1. *Let ϵ be a random p-variate vector. Then ϵ is **spherically symmetrical** if for all orthogonal matrices Q it holds that $\epsilon \sim Q\epsilon$.*

An example of a distribution which is spherically symmetrical is $N_p(0, I_p)$.

Definition 4.2. *Let X be a p-variate random vector. X has an **elliptical distribution** if we can write it as $X = \mu + A\epsilon$. Where $\mu \in \mathbb{R}^p$, A a $p \times p$ matrix and $\epsilon = (\epsilon_1, ..., \epsilon_p)$ is spherically symmetric. If the probability density function exists, it is of the following form:*

$$f_X(x) = |\Psi|^{-\frac{1}{2}} g(\Psi^{-\frac{1}{2}}(x - \mu))$$

where $\Psi = AA^T$ is the scatter matrix which is positive definite and g is the density function of ϵ.

Next, two examples of an elliptical distribution will be given. These Examples are based on respectively example 2.1 and 2.2 of [8].

Example 4.1 (Multivariate normal distribution)**.** Let X be a random vector with a multivariate normal distribution $N_p(\mu, \Sigma)$. In example 2.1 of [8] it is shown that X is elliptical distributed and can be written as $X = \mu + A\epsilon$, with $\Sigma = AA^T$ and $\epsilon \sim N_p(0, I_p)$. X has density function

$$f_X(x) = \frac{\exp(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu))}{\sqrt{(2\pi)^p |\Sigma|}}.$$

Example 4.2 (Multivariate t-distribution)**.** Let X be a random vector with a multivariate t-distribution $t_p(\mu, \Psi, \nu)$. In example 2.2 of [8] it is shown that X is elliptical distributed and can be written as $X = \mu + A\epsilon$, with $\Psi = AA^T$ and $\epsilon \sim t_p(0, I_p, \nu)$. X has density function

$$f_X(x) = \frac{\Gamma((p + \nu)/2)}{\Gamma(\nu/2)\sqrt{(\pi\nu)^p |\Psi|}} [1 + \frac{1}{\nu}(x - \mu)^T \Psi^{-1}(x - \mu)]^{-(\nu+p)/2}.$$

In practice the multivariate t-distribution is for example used in censored data, see [9] and for irregularly observed longitudinal data, see [10].

In figure 1 and 2 the contour lines are plotted for the multivariate normal distribution and respectively the multivariate t-distribution. In figure 3 the contour lines of both are plotted in one plot. A few differences between the contour plots of the multivariate normal distribution and the multivariate t-distribution are that the multivariate normal distribution has lower values closer to the center $(0, 0)$. So the contour lines with value 0.11 to 0.02 are further away from the center for the multivariate normal distribution, then the multivariate t-distribution. But for the last contour at the value 0.01 we see that now the multivariate normal distribution is closer to the center then the multivariate t-distribution.

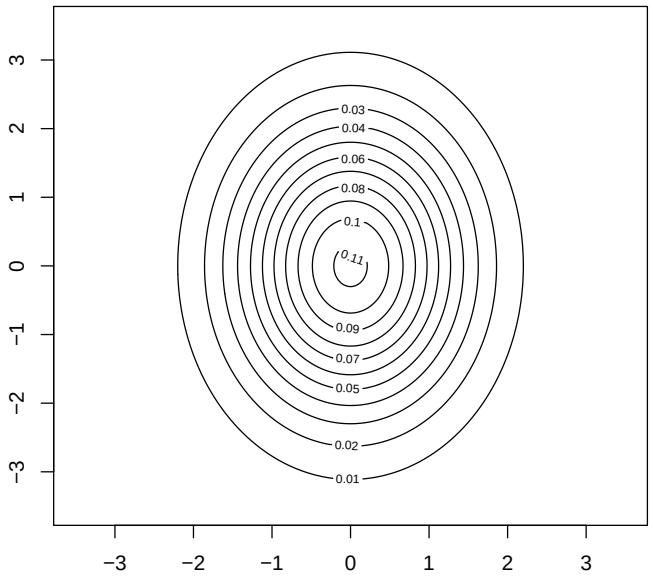


Figure 1: Contour lines of the multivariate normal distribution $N_p(\mu, \Sigma)$ with covariance matrix $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ and $\mu = (0, 0)$.

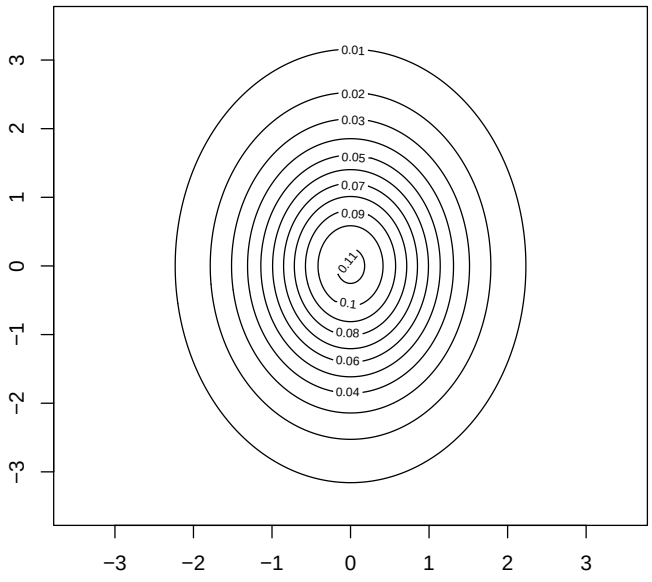


Figure 2: Contour lines of the multivariate t-distribution $t_p(\mu, \Psi, \nu)$ with scatter matrix $\Psi = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$, $\mu = (0, 0)$ and $\nu = 5$.

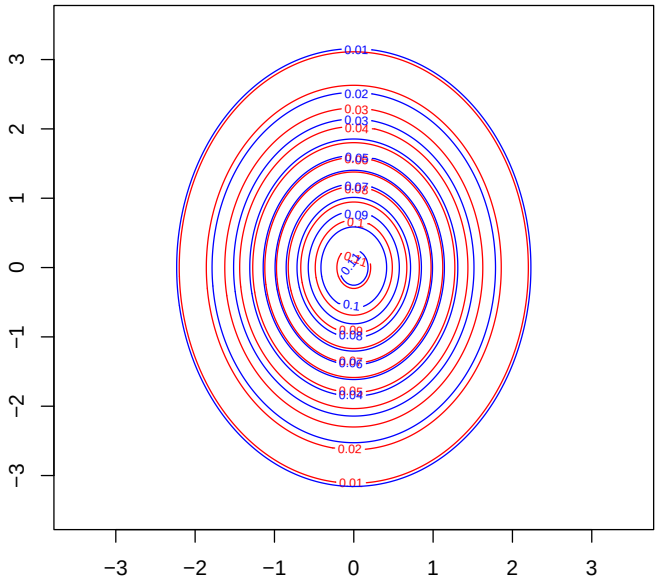


Figure 3: Contour lines of the multivariate normal distribution $N_p(\mu, \Sigma)$ in red and of the multivariate t-distribution $t_p(\mu, \Psi, \nu)$ in blue. $\Sigma = \Psi = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$, $\mu = (0, 0)$ and $\nu = 5$.

4.2 Classical principal components

In this subsection there will be some asymptotic results given about the classical principal components. Firstly there will be a definition given of the kurtosis, which is based on [4] (sections 2.7.4 and 13.8.1).

Definition 4.3. *Let X be a p-variate random vector which is elliptical distributed. Suppose X has positive definite scatter matrix Ψ and $\mathbb{E}[R^2] < \infty$, where $R^2 = (x-\mu)^T \Psi^{-1}(x-\mu)$. Then the* ***kurtosis*** *is*

$$\kappa = \frac{p\mathbb{E}[R^4]}{\mathbb{E}[R^2]^2(p+2)} - 1.$$

The kurtosis is the fourth standarized moment of X. So from now on we will assume that not only the first and second moments exist, but also the fourth moment. If the first and second moment exist for the multivariate normal distribution, then also the fourth moment about the mean exists.[4]. If the first and second moment exist for the multivariat normal distribution, then the fourth moment about the mean exists if the degrees of freedom $\nu > 4$. This is shown in example 4.4.

In [4] it is given that an estimator of κ is

$$\hat{\kappa} = \frac{1}{p(p+2)n} \sum_{i=1}^{n} \left((x_i - \bar{x})^T \hat{\Sigma}^{-1} (x_i - \bar{x}) \right)^2 - 1. \tag{1}$$

Below the kurtosis for the multivariate normal distribution and the multivariate t-distribution will be computed.

Example 4.3 (Kurtosis of a multivariate normal distribution)**.** Let a random vector $X = \mu + A\epsilon$ be distributed as $N_p(\mu, \Sigma)$. In Example 2.1 of [8] it is given that $R^2 = (X-\mu)^T \Sigma^{-1}(X-\mu)$ is distributed

as χ^2_p, because $||\epsilon||_2^2 = R^2$ and $\epsilon \sim N_p(0, I_p)$. From [11] we know the following for a Chi-squared distribution

$$\mathbb{E}[R^2] = p$$

and

$$\text{var}(R^2) = 2p.$$

This gives us that

$$\mathbb{E}[R^4] = \text{var}(R^2) + \mathbb{E}[R^2]^2 = 2p + p^2 = p(p+2).$$

Hence the kurtosis is

$$\kappa = \frac{p\mathbb{E}[R^4]}{\mathbb{E}[R^2]^2(p+2)} - 1 = \frac{p^2(p+2)}{p^2(p+2)} - 1 = 0.$$

Example 4.4 (Kurtosis of a multivariate t-distribution)**.** Let a random vector $X = \mu + A\epsilon$ be distributed as $t_p(\mu, \Psi, \nu)$. In Example 2.2 in [8] it is given that $R^2/p = ((X-\mu)^T\Psi^{-1}(X-\mu))/p$ is distributed as $F(p, \nu)$, because $||\epsilon||_2^2 = R^2$ and $\epsilon \sim t_p(0, I_p, \nu)$. From [12] we know the following holds for a F-distribution

$$\mathbb{E}[R^2/p] = \frac{\nu}{\nu - 2}, \text{ for } \nu > 2$$

and

$$\text{var}(R^2/p) = \frac{2\nu^2(p+\nu-2)}{p(\nu-2)^2(\nu-4)}, \text{ for } \nu > 4.$$

So we have

$$\begin{aligned}\mathbb{E}[R^2] =& p\frac{\nu}{\nu-2}, \text{ for } \nu > 2,\\ \text{var}(R^2) =& p^2\frac{2\nu^2(p+\nu-2)}{p(\nu-2)^2(\nu-4)}, \text{ for } \nu > 4.\end{aligned}$$

From this it follows that

$$\mathbb{E}[R^4] = \text{var}(R^2) + \mathbb{E}[R^2]^2 = p^2\frac{2\nu^2(p+\nu-2)}{p(\nu-2)^2(\nu-4)} + (p\frac{\nu}{\nu-2})^2, \text{ for } \nu > 4.$$

Hence the kurtosis is

$$\kappa = \frac{p\mathbb{E}[R^4]}{\mathbb{E}[R^2]^2(p+2)} - 1 = \frac{2(p+\nu-2)+p(\nu-4)}{(p+2)(\nu-4)} - 1 = \frac{\nu-2}{\nu-4} - 1, \text{ for } \nu > 4.$$

The following theorem gives some asymptotic results of the principal components. This theorem is theorem 13.8.1 from [4]. For the proof of this theorem also see [4]. The idea of this proof will later be used to proof a similar theorem for the principal components based on spatial signs.

Theorem 4.1 (Anderson [4], Theorem 13.8.1)**.** *Let $X_1, ..., X_n$ be n observations of a p-variate random vector X, where X is elliptical distributed. Write the covariance matrix and the sample covariance matrix as eigenvalue decompositions $\Sigma = G\Lambda G^T$ and $\hat{\Sigma} = \hat{G}\hat{\Lambda}\hat{G}^T$, with $\lambda_1 > ... > \lambda_p > 0$, $\hat{\lambda}_1 > ... > \hat{\lambda}_p > 0$ and $g_{1i} \geq 0, \hat{g}_{1i} \geq 0, i = 1, ..., p$. Define $B = \sqrt{n}(\hat{G} - G)$ and $D = \sqrt{n}(\hat{\Lambda} - \Lambda)$. Then the limiting distribution of D is normal with mean equal to zero and $\text{var}(d_i) = (2+3\kappa)\lambda_i^2$ and $\text{cov}(d_i, d_j) = \kappa\lambda_i\lambda_j$. The limiting distribution of B is normal with mean equal to zero and the covariance of b_j is*

$$\text{cov}(b_j) = (1+\kappa)\sum_{i=1, i\neq j}^{p} \frac{\lambda_j\lambda_i}{(\lambda_j - \lambda_i)^2} g_i g_i^T. \tag{2}$$

The covariance of b_i and b_j is

$$\text{cov}(b_i, b_j) = -(1+\kappa)\frac{\lambda_i\lambda_j}{(\lambda_i - \lambda_j)^2} g_j g_i^T. \tag{3}$$

We see that the covariance of the eigenvalues and eigenvectors both depend on the kurtosis and the real eigenvalues and eigenvectors. The covariance is linear dependent on the kurtosis. A large kurtosis means a large covariance. The covariance of the eigenvectors is also large, when the real eigenvalues are close to each other. The covariance of the eigenvalues is large, when the real eigenvalues are large.

4.3 Principal components based on spatial sign

In this subsection some results about the principal components based on spatial signs are given. We would like to proof something similar as in Theorem 4.1, but now for the spatial sign covariance matrix. To do this we need to know something about the eigenvalues of the population SSCM and the asymptotic covariance matrix of the sample SSCM.

The following lemma gives a result about the eigenvalues and eigenvectors of the population SSCM. The only thing needed to know for this is the eigenvalues and eigenvectors of the scatter matrix. The following lemma is a result from [5]. For more details about the lemma and the proof see [5].

Lemma 4.1 (Dürre, Tyler and Vogel [5]). *Let X be a p-variate random vector, which has an elliptical distribution. Then we can write $X = A\epsilon + \mu$. Define $\Psi_0 = \frac{AA^T}{\mathrm{tr}(AA^T)}$. Write $\Psi_0 = O\Lambda O^T$ as an eigenvalue decomposition, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ the eigenvalues of Ψ_0 on the diagonal of Λ. Then $\Sigma_s = O\Delta O^T$, with $\delta_1 \geq \delta_2 \geq ...\delta_p \geq 0$ on the diagonal of Δ, where*

$$\delta_i = \mathbb{E}\left[\lambda_i \epsilon_i^2 \left(\sum_{j=1}^{p} \lambda_j \epsilon_j^2\right)^{-1}\right] \tag{4}$$

$$= \frac{\lambda_i}{2} \int_0^\infty \frac{1}{(1+\lambda_i x)\prod_{k=1}^{p}(1+\lambda_k x)^{1/2}} dx, \tag{5}$$

for $1 \leq i \leq p$.

From this lemma it follows that for the population spatial sign covariance matrix and the covariance matrix the eigenvectors are the same and the eigenvalues have the same ordering. This will give the same directions of the classical principal components and the principal components based on spatial signs.

The following lemma gives a result about the asymptotic covariance matrix of the sample SSCM. This lemma is a result from [5]. For more details about the lemma also see [5]

Lemma 4.2 (Dürre, Tyler and Vogel [5]). *Let $X_1, ..., X_n$ be n observations of a p-variate random vector X, where X is elliptical distributed. Then we can write $X = A\epsilon + \mu$. Define $\Psi_0 = \frac{AA^T}{\mathrm{tr}(AA^T)}$. Write $\Psi_0 = O\Lambda O^T$ as an eigenvalue decomposition, with $\lambda_1 > \lambda_2 > \cdots > \lambda_p > 0$ the eigenvalues of Ψ_0 on the diagonal of Λ. Write the population spatial sign covariance matrix and the sample spatial sign covariance matrix as eigenvalue decompositions $\Sigma_s = O\Delta O^T$ and $\hat{\Sigma}_s = \hat{O}L\hat{O}^T$, with $\delta_1 > ... > \delta_p$, $l_1 > ... > l_p$ and $o_{i1} \geq 0, \hat{o}_{i1} \geq 0, i = 1, ..., p$. The limiting distribution of $\sqrt{n}(\hat{\Sigma}_s - \Sigma_s)$ is normally distributed with mean zero and covariance matrix*

$$W_s = (O \otimes O)\{\Gamma - \mathrm{vec}\Delta(\mathrm{vec}\Delta)^T\}(O \otimes O)^T,$$

where

$$\Gamma = \mathbb{E}\left(\mathrm{vec}\left(\frac{\Lambda^{1/2}\epsilon\epsilon^T\Lambda^{1/2}}{\epsilon^T\Lambda\epsilon}\right)\mathrm{vec}\left(\frac{\Lambda^{1/2}\epsilon\epsilon^T\Lambda^{1/2}}{\epsilon^T\Lambda\epsilon}\right)^T\right). \tag{6}$$

For $1 \leq i, j \leq p$, define

$$\eta_{ij} = \mathbb{E}\left[\lambda_i\epsilon_i^2\lambda_j\epsilon_j^2\left(\sum_{k=1}^{p}\lambda_k\epsilon_k^2\right)^{-2}\right]. \tag{7}$$

This can be written as the following integrals

$$\eta_{ij} = \frac{\lambda_i \lambda_j}{4} \int_0^\infty \frac{x}{(1+\lambda_i x)(1+\lambda_j x)\prod_{k=1}^{p}(1+\lambda_k x)^{1/2}} dx, \text{ for } i \neq j, \tag{8}$$

$$\eta_{ii} = \frac{3\lambda_i^2}{4} \int_0^\infty \frac{x}{(1+\lambda_i x)^2\prod_{k=1}^{p}(1+\lambda_k x)^{1/2}} dx. \tag{9}$$

η_{ij} appears in Γ at the positions $\{(i-1)p+j,(i-1)p+j\},\{(i-1)p+i,(j-1)p+j\},\{(i-1)p+j,(j-1)p+i\}$ and η_{ji} appears at the positions $\{(j-1)p+i,(j-1)p+i\},\{(j-1)p+j,(i-1)p+i\},\{(j-1)p+i,(i-1)p+j\}$, for $1 \leq i < j \leq p$. η_{ii} appears in Γ at the position $\{(i-1)p+i,(i-1)p+i\}$, for $1 \leq i \leq p$. All the other entries of Γ are zero.

The following two definitions is some explanation of the notation that is used in the previous theorem. These definitions are based on definition A.4.1 and respectively definition A.4.2 of [4].

Definition 4.4. *Let A be an $n \times m$ matrix, then*

$$\text{vec}(A) = (a_{11},...,a_{n1},a_{12},...,a_{n2},....,a_{1m},...a_{nm})^T.$$

Definition 4.5. *Let A be an $n \times p$ matrix and B and $m \times q$ matrix. The kronecker product of A and B is*

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1p}B \\ a_{21}B & a_{22}B & \cdots & a_{2p}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{np}B \end{pmatrix}.$$

Below some properties of the vec operator and the kronecker product [4], which we will need later.

$$\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B) \tag{10}$$

$$(O \otimes O)^T = O^T \otimes O^T \tag{11}$$

For dimensions higher then 2 the integrals (5), (8) and (9) cannot be solved explicitly, but then these integrals can be solved numerically, for example in R. For dimension 2 these integrals can be solved explicitly. In the following 2 propositions and corollary the solutions of these integrals are given [3]. Proposition 4.1 and 4.2 are a part of respectively proposition 1 and 2 of [3].

Proposition 4.1 (Dürre, Vogel and Fried [3], Proposition 1 (3))**.** *If the dimension is $p = 2$, then $\delta_i = \frac{\sqrt{\lambda_i}}{\sqrt{\lambda_1}+\sqrt{\lambda_2}}$, for $i = 1, 2$.*

Proposition 4.2 (Dürre, Vogel and Fried [3], Proposition 2 (3))**.** *If the dimension is $p = 2$ and $\lambda_1 \neq \lambda_2$, then*

$$W_s = \frac{-\lambda_1\lambda_2 + \frac{1}{2}\sqrt{\lambda_1\lambda_2}(\lambda_1+\lambda_2)}{(\lambda_1-\lambda_2)^2}(O \otimes O)W_0(O \otimes O)^T,$$

with

$$W_0 = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}.$$

Corollary 4.1. *If $p = 2$ and $\lambda_1 \neq \lambda_2$, then*

- $\eta_{12} = \eta_{21} = \frac{-\lambda_1\lambda_2 + \frac{1}{2}\sqrt{\lambda_1\lambda_2}(\lambda_1+\lambda_2)}{(\lambda_1-\lambda_2)^2}$

- $\eta_{ii} = \eta_{12} + \delta_i^2,\ i = 1, 2$

Proof. From Lemma 4.2 it follows that

$$\begin{aligned} W_s =& (O \otimes O)\{\Gamma - \text{vec}\Delta(\text{vec}\Delta)^T\}(O \otimes O)^T \\ =& (O \otimes O) \begin{pmatrix} \eta_{11} - \delta_1^2 & 0 & 0 & \eta_{12} - \delta_1\delta_2 \\ 0 & \eta_{12} & \eta_{12} & 0 \\ 0 & \eta_{21} & \eta_{21} & 0 \\ \eta_{21} - \delta_1\delta_2 & 0 & 0 & \eta_{22} - \delta_2^2 \end{pmatrix} (O \otimes O)^T \end{aligned}$$

From Proposition 4.2 it then follows that

$$\eta_{12} = \eta_{21} = \frac{-\lambda_1\lambda_2 + \frac{1}{2}\sqrt{\lambda_1\lambda_2}(\lambda_1 + \lambda_2)}{(\lambda_1 - \lambda_2)^2}$$

and

$$\eta_{ii} = \eta_{12} + \delta_i^2 \text{ for, } i = 1, 2$$

□

The following theorem gives some asymptotic results of the principal components based on spatial signs. It is similar to theorem 4.1, but now Σ_s and $\hat{\Sigma}_s$ are used instead of Σ and $\hat{\Sigma}$.

Theorem 4.2. *Let $X_1, ..., X_n$ be n observations of a p-variate random vector X, where X is elliptical distributed. Write the population spatial sign covariance matrix and the sample spatial sign covariance matrix as eigenvalue decompositions $\Sigma_s = O\Delta O^T$ and $\hat{\Sigma}_s = \hat{O}L\hat{O}^T$, with $\delta_1 > ... > \delta_p$, $l_1 > ... > l_p$ and $o_{1i} \geq 0, \hat{o}_{1i} \geq 0, i = 1, ..., p$. Define $G = \sqrt{n}(\hat{O} - O)$ and $D = \sqrt{n}(L - \Delta)$. Then the limiting distribution of D is normal with mean equal to zero and $\text{var}(d_i) = \eta_{ii} - \delta_i^2$ and $\text{cov}(d_i, d_j) = \eta_{ij} - \delta_i\delta_j$. The limiting distribution of G is normal with mean equal to zero and the covariance of g_j is*

$$\text{cov}(g_j) = \sum_{i=1, i\neq j}^{p} \frac{\eta_{ij}}{(\delta_j - \delta_i)^2} o_i o_i^T. \tag{12}$$

The covariance of g_i and g_j is

$$\text{cov}(g_i, g_j) = -\frac{\eta_{ij}}{(\delta_i - \delta_j)^2} o_j o_i^T. \tag{13}$$

Here δ_i is defined as in Lemma 4.1 and η_{ij} is defined as in Lemma 4.2.

The covariance of the eigenvalues and eigenvectors depends on the real eigenvalues and eigenvectors of the covariance matrix. If we compare these covariances with the covariances of Theorem 4.1. Then we see that now the covariances do not depend on the kurtosis, which it did in Theorem 4.1. So even if we have a really large kurtosis, then the covariances don't have to be large. The covariance of the eigenvectors is large, when the real eigenvalues of the spatial sign covariance matrix are really close to each other.

We will need to use the following Theorem to proof Theorem 4.2. This Theorem is Theorem 4.2.3 of [4]. This theorem tells us, when the transformation of a random variable, which has an asymptotic normal distribution, is also asymptotic multivariate normally distributed. And it also tells us which asymptotic covariance it has.

Theorem 4.3 (Anderson [4], Theorem 4.2.3)**.** *Let $\{X_n\}$ be a sequence of p-variate random vectors and θ. Let θ be a p-variate vector such that as $n \to \infty$, then the limiting distribution of $\sqrt{n}(X_n - \theta)$ is $N(0, \Sigma)$. Let $f : \mathbb{R}^p \to \mathbb{R}^m$ be a vector-valued function of x such that $\nabla f_j(x)$ is non-zero at $x = \theta$, for $1 \leq j \leq m$ and let $\nabla f(\theta) = (\nabla f_1(\theta), \nabla f_2(\theta), ..., \nabla f_m(\theta))$. Then $\sqrt{n}(f(x_n) - f(\theta))$ has the limiting distirbution $N(0, \nabla f(\theta)^T \Sigma \nabla f(\theta))$.*

We will use the idea of the proof of Theorem 13.5.1 in [4], to proof theorem 4.2.

Proof of Theorem 4.2. Define $D = \sqrt{n}(L - \Delta), G = \sqrt{n}(\hat{O} - O)$ and $T = O^T \hat{\Sigma}_s O$. Let

$$T = YLY^T, \tag{14}$$

where Y is orthogonal. We need $Y_{ii} \geq 0$, for $i = 1, ...p$, so that (14) determines Y uniquely. Here $Y = O^T \hat{O}$.

Let $\sqrt{n}(T - \Delta) = U$ and $\sqrt{n}(Y - I) = W$. Write $U = \sqrt{n}(T - \Delta)$ as $T = U/\sqrt{n} + \Delta$, $W = \sqrt{n}(Y - I)$ as $Y = W/\sqrt{n} + I$ and $D = \sqrt{n}(L - \Delta)$ as $L = \Delta + D/\sqrt{n}$. Then (14) can be written as

$$\begin{aligned}
&\frac{U}{\sqrt{n}} + \Delta = \left(I + \frac{W}{\sqrt{n}}\right)\left(\Delta + \frac{D}{\sqrt{n}}\right)\left(I + \frac{W}{\sqrt{n}}\right)^T \\
\iff &\frac{U}{\sqrt{n}} + \Delta = \left(\Delta + \frac{D}{\sqrt{n}} + \frac{W\Delta}{\sqrt{n}} + \frac{WD}{n}\right)\left(I + \frac{W}{\sqrt{n}}\right)^T \\
\iff &\frac{U}{\sqrt{n}} + \Delta = \Delta + \frac{D}{\sqrt{n}} + \frac{W\Delta}{\sqrt{n}} + \frac{WD}{n} + \frac{\Delta W^T}{\sqrt{n}} + \frac{DW^T}{n} + \frac{W\Delta W^T}{n} + \frac{WDW^T}{n\sqrt{n}} \\
\iff &U = D + W\Delta + \Delta W^T + \frac{WD}{\sqrt{n}} + \frac{DW^T}{\sqrt{n}} + \frac{W\Delta W^T}{\sqrt{n}} + \frac{WDW^T}{n}.
\end{aligned} \tag{15}$$

And we have that

$$\begin{aligned}
&I = YY^T\left(\frac{W}{\sqrt{n}} + I\right)\left(\frac{W}{\sqrt{n}} + I\right)^T \\
\iff &I = \frac{WW^T}{n} + \frac{W}{\sqrt{n}} + \frac{W^T}{\sqrt{n}} + I \\
\iff &0 = \frac{WW^T}{n} + \frac{W}{\sqrt{n}} + \frac{W^T}{\sqrt{n}} \\
\iff &0 = W + W^T + \frac{WW^T}{\sqrt{n}}
\end{aligned} \tag{16}$$

The limiting distribution of $\sqrt{n}(\hat{\Sigma}_s - \Sigma_s)$ is multivariate normal. From Theorem 4.2.3 of [4] it follows that the limiting distribution of $\sqrt{n}O^T(\hat{\Sigma}_s - \Sigma_s)O = \sqrt{n}(T - \Delta)$ is multivariate normal. Define the function $g(Y, L) = YLY^T$. This function is clearly continuous and differentiable, because it is a polynomial of the matrix elements. Now we can use Theorem 4.3, see the proof of Theorem 13.5.1 of [4] why it satisfies all the conditions of Theorem 4.3. This gives us that the limiting distributions of $W = \sqrt{n}(Y - I)$ and $D = \sqrt{n}(L - \Delta)$ are multivariate normal.

Now by applying the continuous mapping theorem and using the fact that the limiting distribution of W is multivariate normal we get that $\frac{WW^T}{\sqrt{n}} \xrightarrow{d} 0$ and $\frac{W\Delta W^T}{\sqrt{n}} \xrightarrow{d} 0$ as $n \to \infty$.

With the same arguments as before, there also exist a continuous one-to-one map between T, Y to L. So there exist a continuous function from Y to L, so also from W to D. By using the continuous mapping theorem and the fact that the limiting distribution of W is multivariate normal we get that $\frac{WD}{\sqrt{n}} \xrightarrow{d} 0$, $\frac{DW^T}{\sqrt{n}} \xrightarrow{d} 0$, $\frac{WDW^T}{n} \xrightarrow{d} 0$, as $n \to \infty$.

Now (15) and (16) become

$$U = W\Delta + D + \Delta W^T + O_p(1). \tag{17}$$

and

$$0 = W + W^T + O_p(1) \tag{18}$$

Here $O_p(1)$ is an error term, which goes to zero as $n \to 0$. From (18) we get $W = -W^T + O_p(1)$. Substituting this in (17) we get

$$\begin{aligned}
&U = W\Delta + D - \Delta W + O_p(1) \\
\iff &D = U - W\Delta + \Delta + O_p(1)
\end{aligned}$$

From this we get

$$d_{ii} = u_{ii} - w_{ii}\delta_i - \delta_i w_{ii} + O_p(1) = u_{ii} + O_p(1), i = 1, ..., p$$

And if $i \neq j, i, j = 1, ..., p$ we get

$$\begin{aligned} u_{ij} &= w_{ij}\delta_j - \delta_i w_{ij} + O_p(1) \\ \iff w_{ij} &= \frac{u_{ij}}{\delta_j - \delta_i} + O_p(1) \end{aligned}$$

Hence as $n \to \infty$,

$$w_{ij} \xrightarrow{d} \frac{u_{ij}}{\delta_j - \delta_i}$$

From $W = -W^T + O_p(1)$ it follows that $w_{ii} = -w_{ii} + O_p(1) \Rightarrow w_{ii} = O_p(1)$.

From this it follows that $d_{ii} = u_{ii} + O_p(1)$, hence as $n \to \infty$,

$$d_{ii} \xrightarrow{d} u_{ii} \tag{19}$$

Define $\tilde{W}_{ij} = (w_{1,i}, .., w_{i-1,i}, w_{i+1,i}, .., w_{p,i}, w_{1,j}, .., w_{j-1,j}, w_{j+1,j}, .., w_{p,j})^T$, for $1 \leq i < j \leq p$. Then we have that

$$\tilde{W}_{ij} \xrightarrow{d} (\frac{u_{1,i}}{\delta_i - \delta_1}, .., \frac{u_{i-1,i}}{\delta_i - \delta_{i-1}}, \frac{u_{i+1,i}}{\delta_i - \delta_{i+1}}, .., \frac{u_{p,i}}{\delta_i - \delta_p}, \frac{u_{1,j}}{\delta_j - \delta_1}, .., \frac{u_{j-1,j}}{\delta_j - \delta_{j-1}}, \frac{u_{j+1,j}}{\delta_j - \delta_{j+1}}, .., \frac{u_{p,j}}{\delta_j - \delta_p})^T \tag{20}$$

In Lemma 4.2 it is given that $\sqrt{n}\text{vec}(\hat{\Sigma}_s - \Sigma_s)$ has asymptotic covariance matrix:

$$(O \otimes O)\{\Gamma - \text{vec}\Delta(\text{vec}\Delta)^T\}(O \otimes O)^T \tag{21}$$

where Γ is defined as (6).

By using (10) and (11) we get

$$\begin{aligned} (O \otimes O)^T \sqrt{n}\text{vec}(\hat{\Sigma}_s - \Sigma_s) &= (O^T \otimes O^T)\sqrt{n}\text{vec}(\hat{\Sigma}_s - \Sigma_s) \\ &= \sqrt{n}\text{vec}(O^T(\hat{\Sigma}_s - \Sigma_s)O) \\ &= \sqrt{n}\text{vec}(T - \Delta) \end{aligned}$$

Now by Theorem 4.2.3 of [4], we get that the asymptotic covariance matrix of $\sqrt{n}\text{vec}(T - \Delta)$ is

$$(O \otimes O)^T(O \otimes O)\{\Gamma - \text{vec}\Delta(\text{vec}\Delta)^T\}(O \otimes O)^T(O \otimes O) = \{\Gamma - \text{vec}\Delta(\text{vec}\Delta)^T\}$$

Define $A = \{\Gamma - \text{vec}\Delta(\text{vec}\Delta)^T\}$. Define $\tilde{U}_{ij} = (u_{1,i}, .., u_{i-1,i}, u_{i+1,i}, .., u_{p,i}, u_{1,j}, .., u_{j-1,j}, u_{j+1,j}, .., u_{p,j})^T$. The asymptotic covariance matrix of $\tilde{U}_{ij}$ is the submatrix of A consisting of the following columns and rows: $(i-1)p+1, ...(i-1)p+i-1, (i-1)p+i+1, ..., (i-1)p+p, (j-1)p+1, ..., (j-1)p+j-1$, $(j-1)p+j+1, ..., (j-1)p+p$. Call this submatrix A_{ij}.

For $l, k \in \{1, ..., p\}$, $\text{vec}\Delta(\text{vec}\Delta)^T$ is $\delta_k\delta_l$ at the position $\{(k-1)+k, (l-1)+l\}$. At all the other positions of $\text{vec}\Delta(\text{vec}\Delta)^T$ the entries are zero.

By lemma 4.2 the following holds for Γ, for $l, k \in \{1, ..., p\}, l \neq k$. Γ is η_{lk} on the positions $\{(l-1)p+k, (l-1)p+k\}, \{(l-1)p+l, (k-1)p+k\}, \{(l-1)p+k, (k-1)p+l\}$ and η_{kk} at the position $\{(k-1)p+k, (k-1)p+k\}$. Hence the following holds

$$\begin{pmatrix} u_{kk} \\ u_{ll} \end{pmatrix} \xrightarrow{d} N\left(0, \begin{pmatrix} \eta_{kk} - \delta_k^2 & \eta_{kl} - \delta_k\delta_l \\ \eta_{lk} - \delta_l\delta_k & \eta_{ll} - \delta_l^2 \end{pmatrix}\right).$$

Now from (19) we get

$$\begin{pmatrix} d_k \\ d_l \end{pmatrix} \xrightarrow{d} N\left(0, \begin{pmatrix} \eta_{kk} - \delta_k^2 & \eta_{kl} - \delta_k\delta_l \\ \eta_{lk} - \delta_l\delta_k & \eta_{ll} - \delta_l^2 \end{pmatrix}\right).$$

Hence the limiting distribution of D is normal with mean zero and

$$\mathrm{var}(d_k) = \eta_{kk} - \delta_k^2$$

and

$$\mathrm{cov}(d_k, d_l) = \eta_{kl} - \delta_k \delta_l.$$

We have $A_{ij} =$

$$\begin{pmatrix}
\eta_{i1} & 0 & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & \ddots & & & & \vdots & & & & & \vdots & & & & & \vdots \\
\vdots & & \eta_{i(i-1)} & & & \vdots & & & & & \vdots & & & & & \vdots \\
\vdots & & & \eta_{i(i+1)} & & \vdots & & & & & \vdots & & & & & \vdots \\
\vdots & & & & \ddots & 0 & & & & & 0 & & & & & \vdots \\
0 & \cdots & \cdots & \cdots & 0 & \eta_{ij} & 0 & \cdots & \cdots & 0 & \eta_{ij} & 0 & \cdots & \cdots & \cdots & 0 \\
\vdots & & & & & 0 & \ddots & & & & 0 & & & & & \vdots \\
\vdots & & & & & \vdots & & \eta_{ip} & & & \vdots & & & & & \vdots \\
\vdots & & & & & \vdots & & & \eta_{j1} & & \vdots & & & & & \vdots \\
\vdots & & & & & 0 & & & & \ddots & 0 & & & & & \vdots \\
0 & \cdots & \cdots & \cdots & 0 & \eta_{ji} & 0 & \cdots & \cdots & 0 & \eta_{ji} & 0 & \cdots & \cdots & \cdots & 0 \\
\vdots & & & & & 0 & & & & & 0 & \ddots & & & & \vdots \\
\vdots & & & & & \vdots & & & & & \vdots & & \eta_{j(j-1)} & & & \vdots \\
\vdots & & & & & \vdots & & & & & \vdots & & & \eta_{j(j+1)} & & \vdots \\
\vdots & & & & & \vdots & & & & & \vdots & & & & \ddots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & 0 & \eta_{jp}
\end{pmatrix}$$

Define the function

$$f^{(ij)}(\tilde{U}_{ij}) = \left(\frac{u_{1,i}}{\delta_i - \delta_1}, .., \frac{u_{i-1,i}}{\delta_i - \delta_{i-1}}, \frac{u_{i+1,i}}{\delta_i - \delta_{i+1}}, .., \frac{u_{p,i}}{\delta_i - \delta_p}, \frac{u_{1,j}}{\delta_j - \delta_1}, .., \frac{u_{j-1,j}}{\delta_j - \delta_{j-1}}, \frac{u_{j+1,j}}{\delta_j - \delta_{j+1}}, .., \frac{u_{p,j}}{\delta_j - \delta_p} \right)^T$$

By theorem 4.2.3 of [4] we get

$$f^{(ij)}(\tilde{U}_{ij}) \xrightarrow{d} N(0, (\nabla f^{(ij)})^T A_{ij} \nabla f^{(ij)})$$

where

$$\nabla f^{(ij)} = \mathrm{diag}\left(\frac{1}{\delta_i - \delta_1}, ..., \frac{1}{\delta_i - \delta_{i-1}}, \frac{1}{\delta_i - \delta_{i+1}}, ..., \frac{1}{\delta_i - \delta_p}, \frac{1}{\delta_j - \delta_1}, ..., \frac{1}{\delta_j - \delta_{j-1}}, \frac{1}{\delta_j - \delta_{j+1}}, ..., \frac{1}{\delta_j - \delta_p} \right).$$

Now by (20) We get

$$\tilde{W}_{ij} \xrightarrow{d} N(0, (\nabla f^{(ij)})^T A_{ij} \nabla f^{(ij)})$$

We have that $(\nabla f^{(ij)})^T A_{ij} \nabla f^{(ij)} =$

$$\begin{pmatrix}
\frac{\eta_{i1}}{(\delta_i-\delta_1)^2} & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & 0 \\
0 & \ddots & & \vdots & & & & & \vdots & & & \vdots \\
\vdots & & \ddots & 0 & & & & & 0 & & & \\
0 & \cdots & 0 & \frac{\eta_{ij}}{(\delta_i-\delta_j)^2} & 0 & \cdots & \cdots & 0 & \frac{-\eta_{ij}}{(\delta_i-\delta_j)^2} & 0 & \cdots & 0 \\
\vdots & & & 0 & \ddots & & & & 0 & & & \vdots \\
\vdots & & & \vdots & & \frac{\eta_{ip}}{(\delta_i-\delta_p)^2} & & & \vdots & & & \vdots \\
\vdots & & & \vdots & & & \frac{\eta_{j1}}{(\delta_j-\delta_1)^2} & & \vdots & & & \vdots \\
\vdots & & & 0 & & & & \ddots & 0 & & & \vdots \\
0 & \cdots & 0 & \frac{-\eta_{ji}}{(\delta_j-\delta_i)^2} & 0 & \cdots & \cdots & 0 & \frac{\eta_{ji}}{(\delta_j-\delta_i)^2} & 0 & \cdots & 0 \\
\vdots & & & 0 & & & & & 0 & \ddots & & \\
\vdots & & & \vdots & & & & & \vdots & & \ddots & 0 \\
0 & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & 0 & \frac{\eta_{jp}}{(\delta_j-\delta_p)^2}
\end{pmatrix}$$

Define B_{ij} as the matrix $(\nabla f^{(ij)})^T A_{ij} \nabla f^{(ij)}$ where we add 2 rows and 2 columns of all zeros, such that they are i-th, $p+j$-th column and row. The covariance of w_{ii} and w_{jj} with something else is always zero because $w_{ii} = w_{jj} = 0$. Hence the covariance matrix of $\begin{pmatrix} w_i \\ w_j \end{pmatrix}$ is B_{ij}. (Here $W = (w_1, ..., w_p)$.) We also know that W is asymptotic normally distributed, hence we have that

$$\begin{pmatrix} w_i \\ w_j \end{pmatrix} \xrightarrow{d} N(0, B_{ij})$$

We have that $Y = O^T \hat{O}$, so $G = \sqrt{n}(\hat{O} - O)$ has the limiting distribution of

$$G = \sqrt{n}(\hat{O} - O) = \sqrt{n}(Oy - O) = O\sqrt{n}(Y - I) = OW.$$

Define the function $\tilde{f}\left(\begin{pmatrix} w_i \\ w_j \end{pmatrix}\right) = \begin{pmatrix} Ow_i \\ Ow_j \end{pmatrix} = \text{vec}\,(O(w_i, w_j))$. Then by theorem 4.2.3 of [4] we get

$$\text{vec}\,(O(w_i, w_j)) \xrightarrow{d} N(0, \nabla \tilde{f}^T B_{ij} \nabla \tilde{f}),$$

where

$$\nabla \tilde{f} = \left(\begin{array}{c|c} O^T & 0 \\ \hline 0 & O^T \end{array}\right).$$

We have that

$$\nabla \tilde{f}^T B_{ij} \nabla \tilde{f} = \left(\begin{array}{c|c} \displaystyle\sum_{k=1,k\neq i}^{p} \frac{\eta_{ki}}{(\delta_i - \delta_k)^2} o_k o_k^T & -\frac{\eta_{ij}}{(\delta_i - \delta_j)^2} o_j o_i^T \\ \hline -\frac{\eta_{ji}}{(\delta_j - \delta_i)^2} o_i o_j^T & \displaystyle\sum_{k=1,k\neq j}^{p} \frac{\eta_{kj}}{(\delta_j - \delta_k)^2} o_k o_k^T \end{array}\right).$$

Now from this it follows that the asymptotic covariance matrix of g_j is

$$\text{cov}(g_j) = \sum_{i=1,i\neq j}^{p} \frac{\eta_{ij}}{(\delta_j - \delta_i)^2} o_i o_i^T.$$

and the asymptotic covariance matrix of g_i and g_j is

$$\text{cov}(g_i, g_j) = -\frac{\eta_{ij}}{(\delta_i - \delta_j)^2} o_j o_i^T$$

□

5 Asymptotic MSE

The MSE is a measure for the accuracy of an estimator and we would like the MSE to be as small as possible [11]. It could be helpful to look at the MSE of the classical principal components and the principal components based on spatial signs, to see which has a smaller MSE. In other words to see which estimator is more accurate. Theorem 4.1 is used to find the MSE of the classical principal components.

Theorem 5.1. *Let $X_1, ..., X_n$ be n observations of a p-variate random vector X, where X is elliptical distributed. Write the covariance matrix and the sample covariance matrix as eigenvalue decompositions $\Sigma = \Gamma\Lambda\Gamma^T$ and $\hat{\Sigma} = GLG^T$, with $\lambda_1 > ... > \lambda_p$, $l_1 > ... > l_p$ and $\gamma_{i1} \geq 0, g_{i1} \geq 0, i = 1, ..., p$. As $n \to \infty$,*

$$MSE_c = \mathbb{E}\left(||vec(G) - vec(\Gamma)||_2^2\right) \xrightarrow{p} \frac{1}{n}(1+\kappa)\sum_{j=1}^{p}\sum_{i=1, i\neq j}^{p}\frac{\lambda_j\lambda_i}{(\lambda_j-\lambda_i)^2}$$

Proof. If we write $\Sigma = \Gamma\Lambda\Gamma^T$ and $\hat{\Sigma} = GLG^T$ as eigenvalue decompositions. Define $\hat{Y} = \text{vec}(G)$ and $Y = \text{vec}(\Gamma)$. Then we have that the MSE is

$$\begin{aligned}
\text{MSE}_c =& \mathbb{E}\left(||\hat{Y} - Y||_2^2\right) \\
=& \mathbb{E}[(g_{11} - \gamma_{11})^2] + ... + \mathbb{E}[(g_{p1} - \gamma_{p1})^2] + ... + \mathbb{E}[(g_{1p} - \gamma_{1p})^2] + ... + \mathbb{E}[(g_{pp} - \gamma_{pp})^2] \\
=& \sum_{i=1}^{p}\sum_{j=1}^{p}\mathbb{E}[(g_{ij} - \gamma_{ij})^2] \\
=& \sum_{i=1}^{p}\sum_{j=1}^{p}\left(\text{var}(g_{ij} - \gamma_{ij}) + \mathbb{E}[g_{ij} - \gamma_{ij}]^2\right)
\end{aligned}$$

Now by Theorem 4.1 as $n \to \infty$,

$$\begin{aligned}
\sum_{i=1}^{p}\sum_{j=1}^{p}\left(\text{var}(g_{ij} - \gamma_{ij}) + \mathbb{E}[g_{ij} - \gamma_{ij}]^2\right) \to& \frac{1}{n}\sum_{j=1}^{p}\text{tr}(\text{cov}(b_j)) \\
=& \frac{1}{n}\sum_{j=1}^{p}\text{tr}\left((1+\kappa)\sum_{i=1, i\neq j}^{p}\frac{\lambda_j\lambda_i}{(\lambda_j-\lambda_i)^2}\gamma_i\gamma_i^T\right) \\
=& \frac{1}{n}\sum_{j=1}^{p}\left((1+\kappa)\sum_{i=1, i\neq j}^{p}\frac{\lambda_j\lambda_i}{(\lambda_j-\lambda_i)^2}\right)
\end{aligned}$$

Hence as $n \to \infty$,

$$\text{MSE}_c \to \frac{1}{n}\sum_{j=1}^{p}\left((1+\kappa)\sum_{i=1, i\neq j}^{p}\frac{\lambda_j\lambda_i}{(\lambda_j-\lambda_i)^2}\right)$$

□

Note that it always holds that $\kappa \geq -1$, because $\mathbb{E}[R^4]$ and $[R^2]$ in definition 4.3 are always positive. This means that MSE_c is never negative, hence it is well-defined.

The MSE is depended on the kurtosis and the real eigenvalues of the covariance matrix. The MSE is linearly dependent on the kurtosis. The MSE is large, when the kurtosis is large. If the eigenvalues are close to each other, then the MSE is also large. It converges towards 0 with rate $\frac{1}{n}$.

Theorem 4.2 is used to find the MSE of the principal components based on spatial signs.

Theorem 5.2. *Let $X_1, ..., X_n$ be n observations of a p-variate random vector X, where X is elliptical distributed. Write the population spatial sign covariance matrix and the sample spatial sign covariance*

matrix as eigenvalue decompositions $\Sigma_s = O\Delta O^T$ *and* $\hat{\Sigma}_s = \hat{O}L\hat{O}^T$*, with* $\delta_1 > ... > \delta_p$*,* $l_1 > ... > l_p$ *and* $o_{i1} \geq 0, \hat{o}_{i1} \geq 0, i = 1, ..., p$*. Then as* $n \to \infty$*,*

$$MSE_{ss} = \mathbb{E}\left(||vec(\hat{O}) - vec(O)||_2^2\right) \to \frac{1}{n}\sum_{j=1}^{p}\sum_{i=1, i\neq j}^{p}\frac{\eta_{ij}}{(\delta_j - \delta_i)^2}.$$

Here δ_i *is defined as in Lemma 4.1 and* η_{ij} *is defined as in Lemma 4.2.*

Now the MSE is depended on the real eigenvalues of the covariance matrix and the spatial sign covariance matrix. This also converges towards 0 with rate $\frac{1}{n}$. One of the differences between this MSE and the MSE for the classical case is that now the MSE is not depended on the kurtosis.

Proof. The same as the proof of Theorem 5.1, but now using Theorem 4.2 instead of Theorem 4.1 □

In table 1 the MSE is simulated and compared with the asymptotic MSE from Theorem 5.1 and Theorem 5.2. For the classical principal components the simulation is done by computing the eigenvectors of the sample covariance matrix and repeating this 10000 times. Then the MSE is computed. For the spatial sign case the same is done but then the sample SSCM is used. It is done for a multivariate normal distribution and a multivariate t-distribution. These seem to converge towards each other as n increases, so the convergence seems to work well. In table 2 the same is done, but now for dimension 3 and dimension 12. This is only done for a multivariate normal distribution. For dimension 3 it takes longer to converge then for dimension 2. For dimension 12 it takes even longer to converge. At $n = 1000$ it still doesn't seem to converge. Hence as the dimension increases the convergence rate seems to decrease.

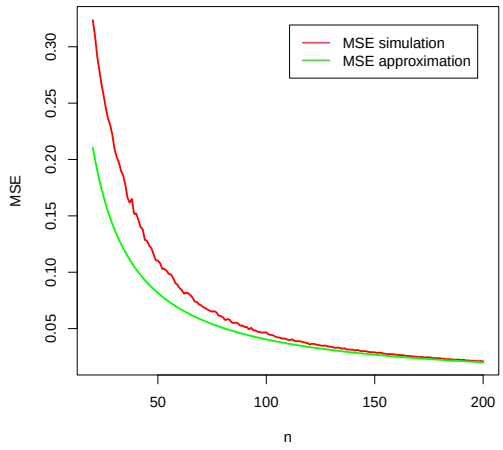
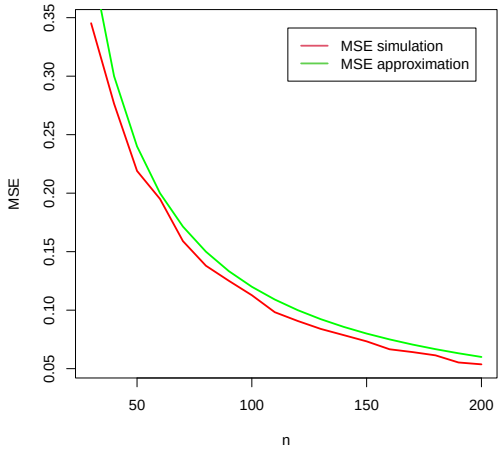
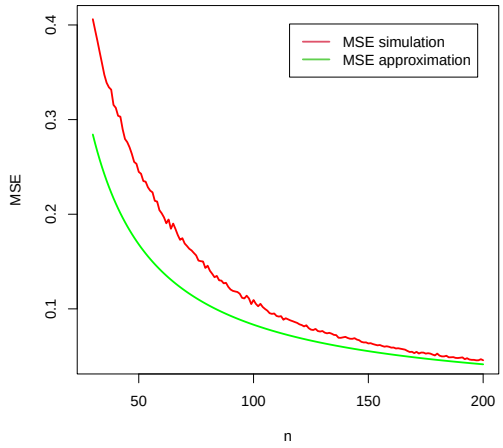
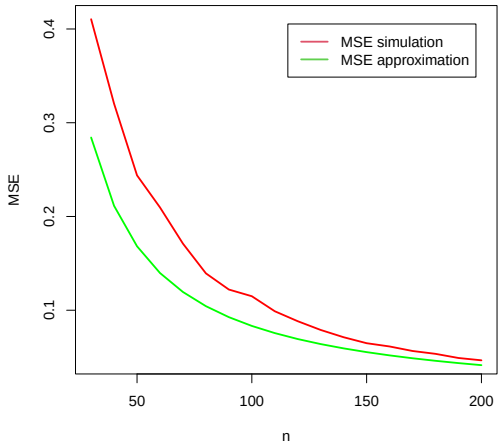
	Multivariate normal distribution	Multivariate t-distribution with $\nu = 5$
Classic		
Spatial signs		

Table 1: The scatter matrix is $\Psi = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$. The red line is the simulated MSE and the green line is the MSE approximated by using Theorem 5.1 for the classical case and Theorem 5.2 for the spatial sign case.

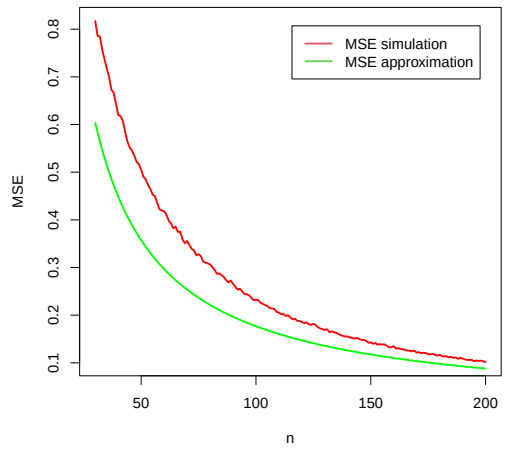
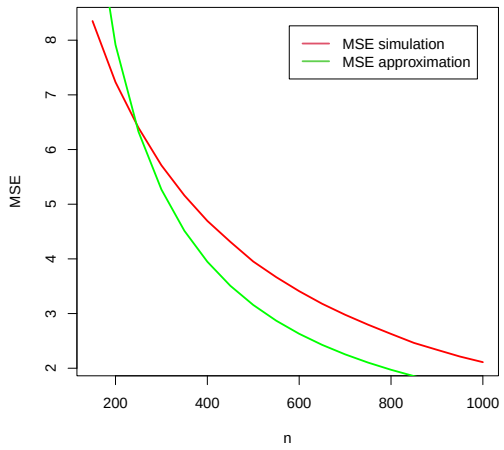
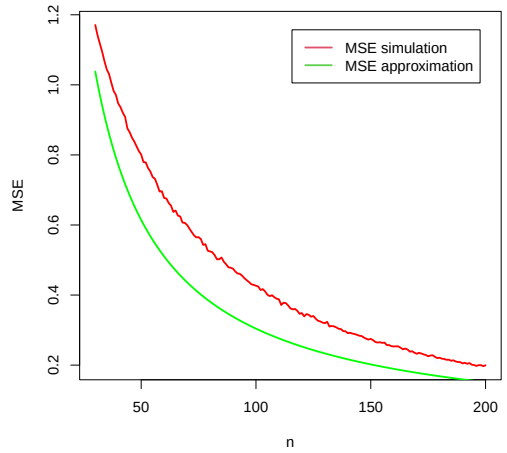
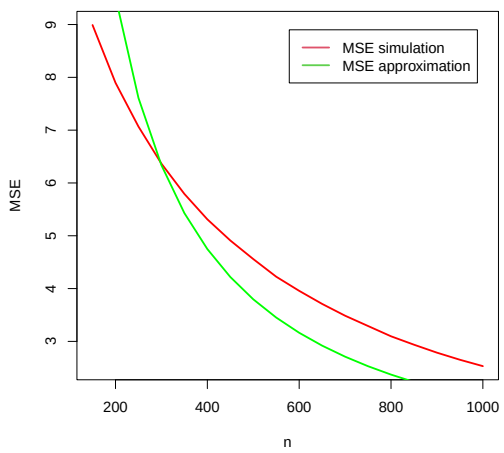
	Scatter matrix is $\Psi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$	Scatter matrix is $\Psi = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 12 \end{pmatrix}$
classic		
spatial signs		

Table 2: This is for a multivariate normal distribution. The red line is the simulated MSE and the green line is the MSE approximated by using Theorem 5.1 for the classical case and Theorem 5.2 for the spatial sign case.

It would be nice two know which estimator is more precise. This is why in this chapter the asymptotic MSEs that are found in Theorem 5.1 and Theorem 5.2 will be compared. This will be done for dimension 2, 3 and 4. If the dimension gets higher, then the number of eigenvalues that need to be chosen also grows. Hence it's gets more and more difficult to compare it, when the dimension grows. This is the reason why there will only be looked at dimension 2, 3 and 4.

One thing that is immediately clear if we compare MSE_c and MSE_{ss}, is that MSE_c and the kurtosis are linearly dependent with each other and MSE_{ss} is independent of the kurtosis. This means that at some point if the kurtosis is large enough, then $\text{MSE}_{ss} < \text{MSE}_c$. So principal component analysis based on spatial signs works equally well for a really large kurtosis as for a small kurtosis. For classical principal component analysis this is not the case.

5.1 Dimension 2

For dimension 2 it is possible to find an explicit expression for κ such that $\text{MSE}_c < \text{MSE}_{ss}$, because then the integrals (5) and (8) can be solved explicitly. All the results in this subsection are for dimension 2.

Proposition 5.1. *Let $X = A\epsilon + \mu$ be elliptical distributed. Define $\Psi_0 = \frac{AA^T}{tr(AA^T)}$. Let λ_1, λ_2 be the eigenvalues of Ψ_0. If $\lambda_1, \lambda_2 > 0$ and $\lambda_1 \neq \lambda_2$, then*

$$MSE_c < MSE_{ss} \iff \kappa < \frac{1}{2\sqrt{\lambda_1(1-\lambda_1)}}$$

and

$$MSE_c > MSE_{ss} \iff \kappa > \frac{1}{2\sqrt{\lambda_1(1-\lambda_1)}}$$

Proof. Let $\lambda_1, \lambda_2 > 0$ such that $\lambda_1 + \lambda_2 = 1$ and $\lambda_1 \neq \lambda_2$ be arbitrary given. By Theorem 4.1 and Theorem 4.2

$$\begin{aligned} \text{MSE}_c < \text{MSE}_{ss} &\iff \\ (\kappa+1)\frac{2}{n}\frac{\lambda_1\lambda_2}{(\lambda_1-\lambda_2)^2} < \frac{2}{n}\frac{\eta_{12}}{(\delta_1-\delta_2)^2} &\iff \\ (\kappa+1)\frac{\lambda_1\lambda_2}{(\lambda_1-\lambda_2)^2} < \frac{\eta_{12}}{(\delta_1-\delta_2)^2} & \end{aligned}$$

Then by proposition 4.1 and corollary 4.1 we get

$$\begin{aligned} (\kappa+1)\frac{\lambda_1\lambda_2}{(\lambda_1-\lambda_2)^2} < \frac{\eta_{12}}{(\delta_1-\delta_2)^2} &\iff \\ (\kappa+1)\frac{\lambda_1\lambda_2}{(\lambda_1-\lambda_2)^2} < \frac{\frac{-\lambda_1\lambda_2+\frac{1}{2}\sqrt{\lambda_1\lambda_2}(\lambda_1+\lambda_2)}{(\lambda_1-\lambda_2)^2}}{\left(\frac{\sqrt{\lambda_1}}{\sqrt{\lambda_1}+\sqrt{\lambda_2}} - \frac{\sqrt{\lambda_2}}{\sqrt{\lambda_1}+\sqrt{\lambda_2}}\right)^2} &\iff \\ (\kappa+1)\frac{\lambda_1\lambda_2}{(\lambda_1-\lambda_2)^2} < \frac{\sqrt{\lambda_1\lambda_2}(\sqrt{\lambda_1}+\sqrt{\lambda_2})^2}{2(\lambda_1-\lambda_2)^2} &\iff \\ 2\sqrt{\lambda_1\lambda_2}(\kappa+1) < (\sqrt{\lambda_1}+\sqrt{\lambda_2})^2 &\iff \\ 2\kappa\sqrt{\lambda_1\lambda_2} < \lambda_1+\lambda_2 = 1 &\iff \\ \kappa < \frac{1}{2\sqrt{\lambda_1\lambda_2}} = \frac{1}{2\sqrt{\lambda_1(1-\lambda_1)}} & \end{aligned}$$

The expression

$$\text{MSE}_c > \text{MSE}_{ss} \iff \kappa > \frac{1}{2\sqrt{\lambda_1(1-\lambda_1)}}$$

can be proved by replacing $<$ with $>$ everywhere.

□

Proposition 5.2. *Let $X = A\epsilon + \mu$ be elliptical distributed. Define $\Psi_0 = \frac{AA^T}{tr(AA^T)}$. Let λ_1, λ_2 be the eigenvalues of Ψ_0. If $\lambda_1, \lambda_2 > 0$, $\lambda_1 \neq \lambda_2$ and $\kappa < 1$, then $MSE_c < MSE_{ss}$.*

Proof. It is easy to check that the minimum of $\kappa = \frac{1}{2\sqrt{\lambda_1(1-\lambda_1)}}$ is $\kappa = 1$. From Proposition 5.1 it then follows that if $\kappa < 1$, then $\text{MSE}_c < \text{MSE}_{ss}$. □

The kurtosis of the multivariate normal distribution is always 0, hence for a multivariate normal distribution of dimension 2 it always holds that $\text{MSE}_c < \text{MSE}_{ss}$

Proposition 5.3. *Suppose $X = A\epsilon + \mu$ has a multivariate t-distribution. Define $\Psi_0 = \frac{AA^T}{tr(AA^T)}$. Let λ_1, λ_2 be the eigenvalues of Ψ_0. If $\lambda_1, \lambda_2 > 0$ and $\lambda_1 \neq \lambda_2$, then*

$$MSE_c < MSE_{ss} \iff 4\sqrt{\lambda_1(1-\lambda_1)} + 4 < \nu$$

and

$$MSE_c > MSE_{ss} \iff 4\sqrt{\lambda_1(1-\lambda_1)} + 4 > \nu$$

Proof. Let $\lambda_1, \lambda_2 > 0$ such that $\lambda_1 + \lambda_2 = 1$ and $\lambda_1 \neq \lambda_2$ be arbitrary given. For a multivariate t-distribution and $p = 2$ we have that

$$\kappa = \frac{\nu - 2}{\nu - 4} - 1$$

Now by Proposition 5.1 we get

$$\begin{aligned} \text{MSE}_c < \text{MSE}_{ss} &\iff \\ \frac{\nu-2}{\nu-4} - 1 < \frac{1}{2\sqrt{\lambda_1(1-\lambda_1)}} &\iff \\ \frac{2}{\nu-4} < \frac{1}{2\sqrt{\lambda_1(1-\lambda_1)}} &\iff \\ 4\sqrt{\lambda_1(1-\lambda_1)} + 4 < \nu & \end{aligned}$$

The expression

$$\text{MSE}_c > \text{MSE}_{ss} \iff 4\sqrt{\lambda_1(1-\lambda_1)} + 4 > \nu$$

can be proved by replacing $<$ with $>$ everywhere. □

5.2 Dimension 3

For dimension 3 it is a bit harder to compare MSE_c and MSE_{ss}, because there is one more eigenvalue that can be chosen freely and there is not an explicit expression for δ_i and η_{ij}, $i, j = 1, 2, 3, i \neq j$. However the integrals (5) and (8) could be solved numerically, hence it is possible to compare MSE_c and MSE_{ss} numerically.

In all the figures below for everything above the function it holds that $\text{MSE}_{ss} < \text{MSE}_c$ and below the function it holds that $\text{MSE}_{ss} > \text{MSE}_c$. In figure 9 a 3D plot is made of the κ for which $\text{MSE}_c = \text{MSE}_{ss}$. It seems like when λ_2 or λ_3 gets closer to zero or one κ seems to increase. When λ_2 and λ_3 get further away from zero and one κ seems to decrease.

To get a better idea of what happens, also some 2D plots are made. In figures 4, 5, 6, 7 and 8 2D plots are made for a fixed λ_1, respectively λ_1 is $\frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}$ and $\frac{5}{6}$. In figure 5 it is a parabola. In all these figures as λ_1 gets closer to zero or $1 - \lambda_1$ (then λ_3 gets closer to zero), then κ increases. Except in figure 8, there it first increases as it gets closer to zero or $1 - \lambda_1$ and at some point it decreases. This could also be an error in the numerical computation. The minimal points in the figures seem to be approximately in the middle of λ_2 and $1 - \lambda_1$. So when $\lambda_2 = \lambda_3$. Note that it is not possible to calculate the asymptotic MSEs for $\lambda_2 = \lambda_3$, because we then have to divide by zero. The largest point in the figures is approximately $\kappa = 1.4$ in figure 4 and 8. The smallest point in the figures is approximately $\kappa = 0.65$ in figure 5.

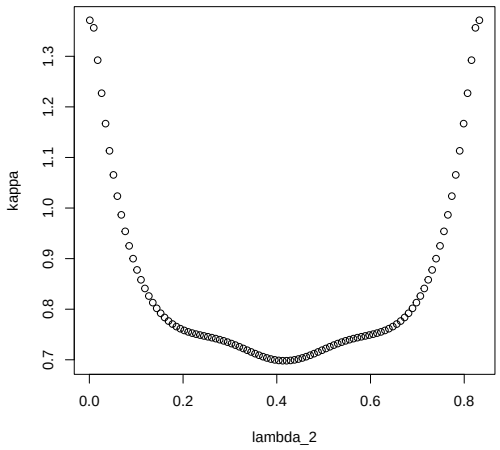


Figure 4: The dimension is 3. The first eigenvalue $\lambda_1 = \frac{1}{6}$ is fixed. On the x-axis is the second eigenvalue λ_2 and on the y-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

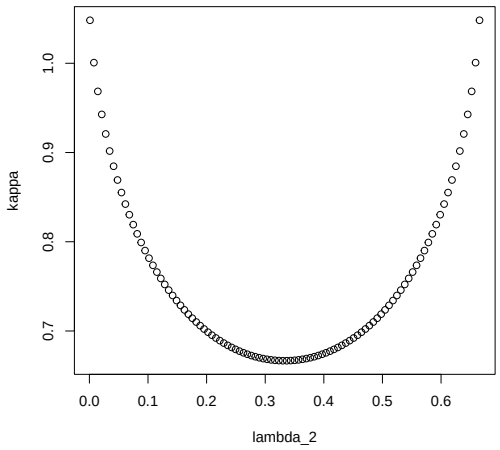


Figure 5: The dimension is 3. The first eigenvalue $\lambda_1 = \frac{1}{3}$ is fixed. On the x-axis is the second eigenvalue λ_2 and on the y-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

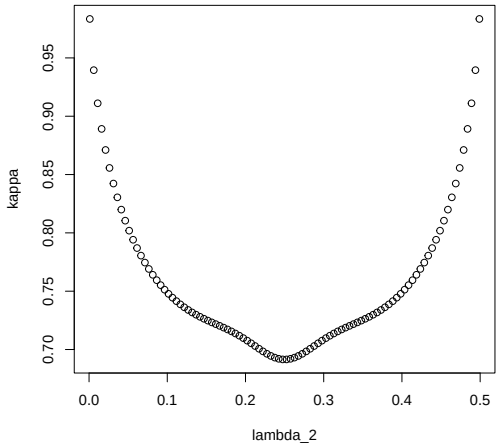


Figure 6: The dimension is 3. The first eigenvalue $\lambda_1 = \frac{1}{2}$ is fixed. On the x-axis is the second eigenvalue λ_2 and on the y-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

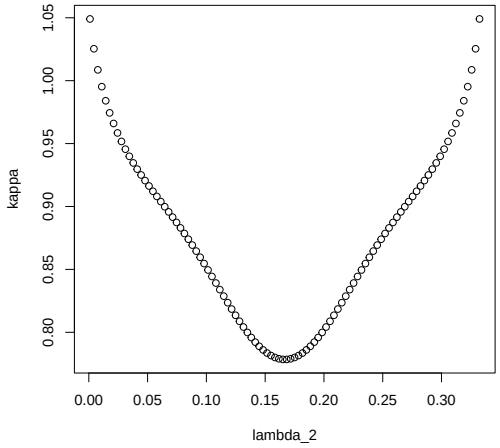


Figure 7: The dimension is 3. The first eigenvalue $\lambda_1 = \frac{2}{3}$ is fixed. On the x-axis is the second eigenvalue λ_2 and on the y-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

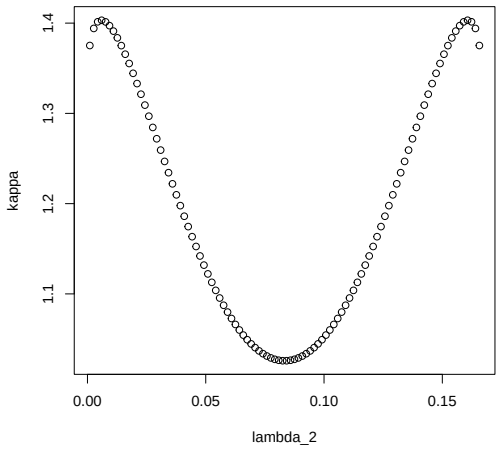


Figure 8: The dimension is 3. The first eigenvalue $\lambda_1 = \frac{5}{6}$ is fixed. On the x-axis is the second eigenvalue λ_2 and on the y-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

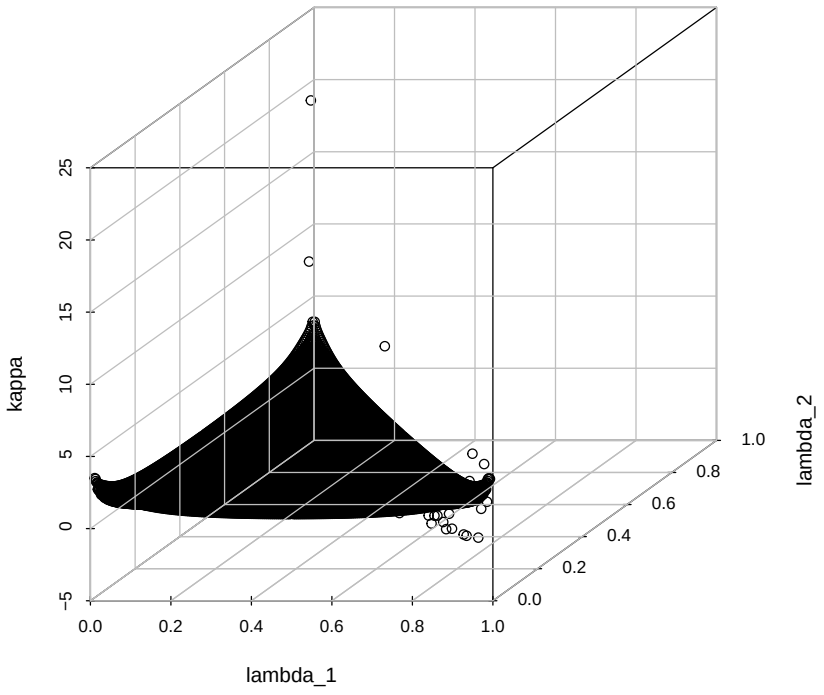


Figure 9: The dimension is 3. On the x-axis and the y-axis there is the eigenvalues, λ_1 and respectively λ_2. On the z-axis there is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$. Note here that the third eigenvalue can be computed from the fact that $\lambda_1 + \lambda_2 + \lambda_3 = 1$

5.3 Dimension 4

In all the figures below again for everything above the function it holds that $\text{MSE}_{ss} < \text{MSE}_c$ and below the function it holds that $\text{MSE}_{ss} > \text{MSE}_c$. In figures 10, 11, 12, 13 and 14 a 3D plot is made of the κ for which $\text{MSE}_c = \text{MSE}_{ss}$ and a fixed λ_1, respectively λ_1 is $\frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}$ and $\frac{5}{6}$. Again it seems like when λ_2 or λ_3 gets closer to zero or one, κ seems to increase. When λ_2 and λ_3 get further away from zero and one, κ seems to decrease. The global minimum seems to be at approximately $\kappa = 0.5$ in all five figures. So it seems that for most eigenvalues we have that if $\kappa < 0.5$, then $\text{MSE}_c < \text{MSE}_{ss}$.

In table 3 the κ for which $\text{MSE}_c = \text{MSE}_{ss}$ is computed for fixed λ_1 and λ_2. The λ_3 is on the x-axis and κ on the y-axis. For $\lambda_1 = \frac{1}{6}, \lambda_2 = \frac{2}{6}$ and $\lambda_1 = \frac{1}{6}, \lambda_2 = \frac{3}{6}$ it seems to behave the same as for dimension 2. For $\lambda_1 = \frac{1}{6}, \lambda_2 = \frac{4}{6}$ and $\lambda_2 = \frac{2}{6}, \lambda_2 = \frac{3}{6}$ there are two minimums with a maximum in the middle.

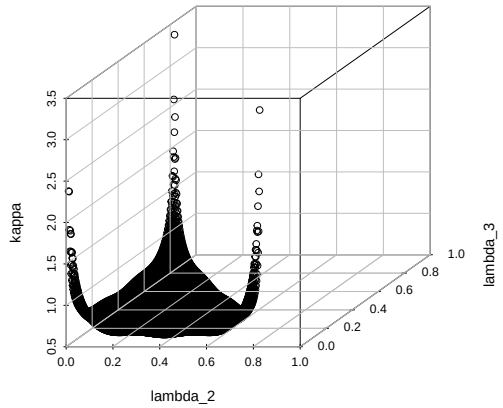


Figure 10: The dimension is 4. The first eigenvalue $\lambda_1 = \frac{1}{6}$ is fixed. On the x-axis is the second eigenvalue λ_2, on the y-axis is the third eigenvalue λ_3 and on the z-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

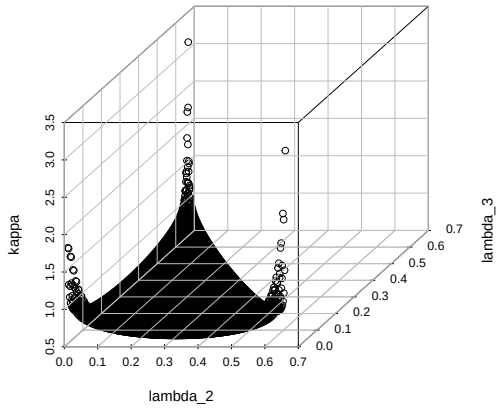


Figure 11: The dimension is 4. The first eigenvalue $\lambda_1 = \frac{1}{3}$ is fixed. On the x-axis is the second eigenvalue λ_2, on the y-axis is the third eigenvalue λ_3 and on the z-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

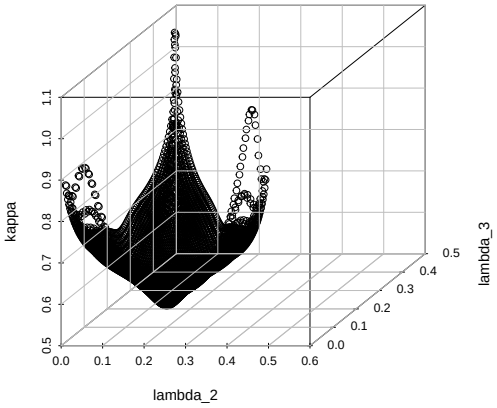


Figure 12: The dimension is 4. The first eigenvalue $\lambda_1 = \frac{1}{2}$ is fixed. On the x-axis is the second eigenvalue λ_2, on the y-axis is the third eigenvalue λ_3 and on the z-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

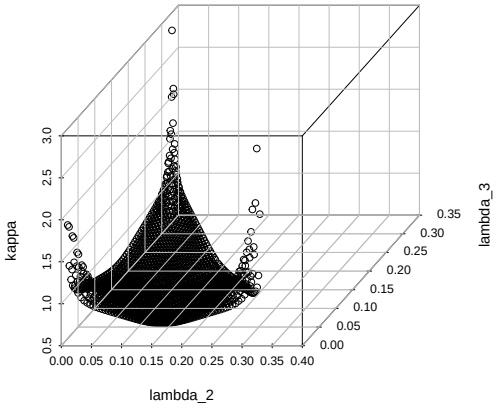


Figure 13: The dimension is 4. The first eigenvalue $\lambda_1 = \frac{2}{3}$ is fixed. On the x-axis is the second eigenvalue λ_2, on the y-axis is the third eigenvalue λ_3 and on the z-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

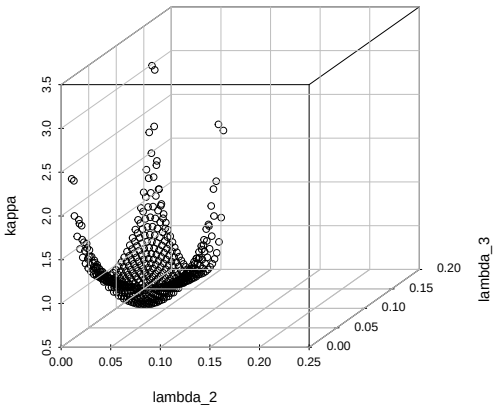


Figure 14: The dimension is 4. The first eigenvalue $\lambda_1 = \frac{5}{6}$ is fixed. On the x-axis is the second eigenvalue λ_2, on the y-axis is the third eigenvalue λ_3 and on the z-axis is the κ for which it holds that $\text{MSE}_c = \text{MSE}_{ss}$.

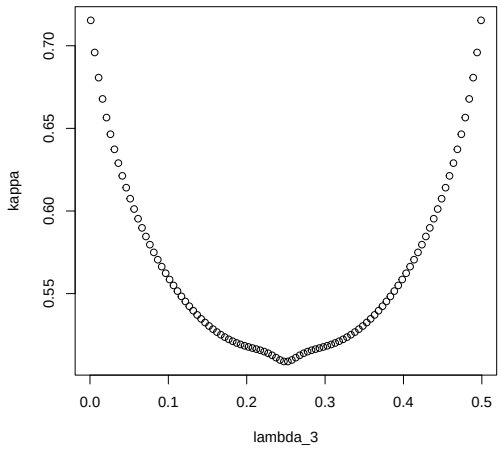
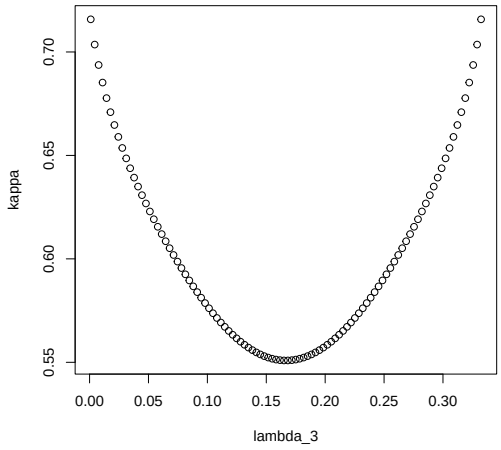
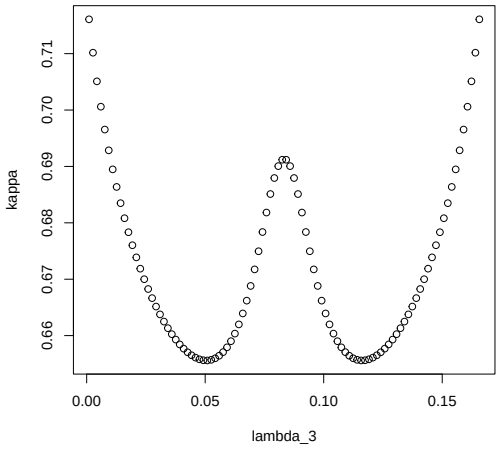
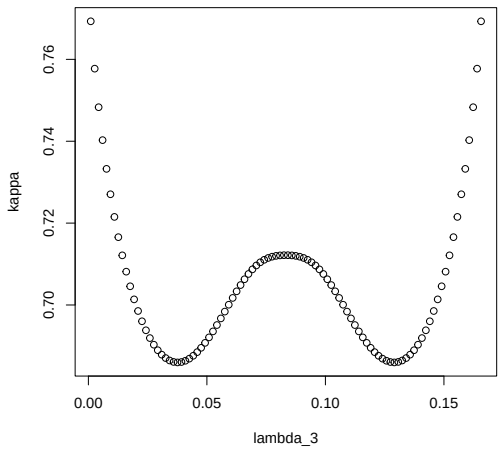
	$\lambda_1 = \frac{1}{6}$	$\lambda_1 = \frac{1}{3}$
$\lambda_2 = \frac{1}{3}$		
$\lambda_2 = \frac{1}{2}$		
$\lambda_2 = \frac{2}{3}$		

Table 3: The dimension is 4. The first and second eigenvalues are fixed. On the x-axis is the third eigenvalue λ_3 and on the y-axis is the κ for which it holds that $\mathsf{MSE}_c = \mathsf{MSE}_{ss}$.

6 Confidence ellipsoids

Theorem 4.1 and Theorem 4.2 could also be used for other things then finding the MSE. It could for example be used to find confidence ellipsoids. In [13] it is given that the formula of a confidence ellipsoid of a p-variate normal distribution is

$$(x-\mu)^T\Sigma^{-1}(x-\mu) \leq \chi_p^2(k) \tag{22}$$

Here μ is the mean vector, Σ the covariance matrix and $\chi_p^2(k)$ is the quantile function of a chi-squared distribution with degrees of freedom p and probability k. Here the degrees of freedom p is the dimension of the multivariate normal distribution. When using a confidence ellipsoid in practice, Σ needs to be estimated.

6.1 Example: Confidence ellipses of loading vectors

Everything explained about biplots and loading vectors is based on [2]. Figures 15 and 16 are biplots of the first and the second principal component for the marks data set, which can be found in the bnlearn package in R (`https://search.r-project.org/CRAN/refmans/bnlearn/html/marks.html`). This data is of 88 students and their mark for the following topics: mechanics, vectors, algebra, analysis and statistics. So the dimension is 5 and the sample size is 88. In figure 15 it is done for the classical principal components and in figure 16 it is done for the principal components based on spatial signs. These biplots are plots of the first and second principal components, where in the same plot the loadings for the first and second principal components are plotted. We plot these loadings as a vector for every variable. These vectors, which we will call the loading vectors, are computed in the following way for the classical case. Write $\hat{\Sigma}$ as an eigenvalue decomposition $\hat{\Sigma} = \hat{G}\hat{\Lambda}\hat{G}^T$, with $\hat{\lambda}_1 \geq ... \geq \hat{\lambda}_p \geq 0$. Then the loading vector is $w_i = (\hat{g}_{i1}, \hat{g}_{i2})^T$, for variable i. For the spatial sign case $\hat{\Sigma}_s$ is used instead of $\hat{\Sigma}$ to compute the loading vectors in figure 16. In these biplots the 0.95-confidence ellipses of the loading vectors are also included. These confidence ellipses are calculated by using formula (22). The covariance matrix Σ in (22) is estimated by using the covariances (2) and (3) for the classical case and the covariances (12) and (13) for the spatial sign case. Also including the 0.95-confidence ellipses in these biplots could for example be helpful in analyzing the data. In these plots this could be analyzed in the following way. The first principal component tells us if a student is good in general or not, because all the confidence ellipses are on the right side. The second principal component tells us that, a student is good at the topics for which the confidence ellipses are at the upper side of the plane or is good in the topics for which the confidence ellipses are at the lower side of the plane. So in figure 15, a student is better at mechanics and vectors or they are better at analysis and statistics. One difference we see between the two figures is that for the spatial sign case the confidence ellipses are larger then for the classical case. Which then also gives us another analysis for figure 16, because now the confidence ellipsoid of vectors and analysis is not completely at the upper or lower side of the plane. So now we would conclude that the second principal components measures if a student is good in mechanics or statistics.

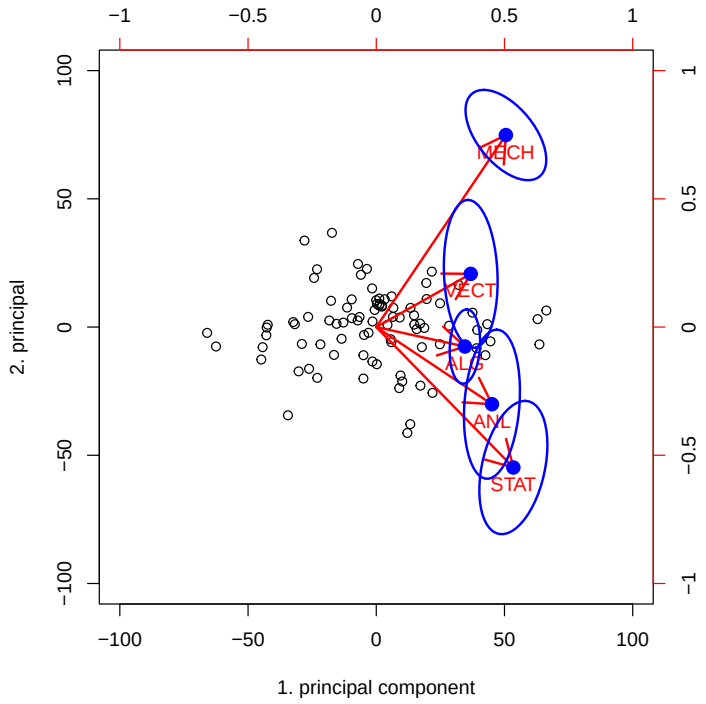


Figure 15: Biplot for the marks data set. The red arrows are the loading vectors of the variables. The blue ellipses are the 0.95-confidence ellipses of the loading vectors.

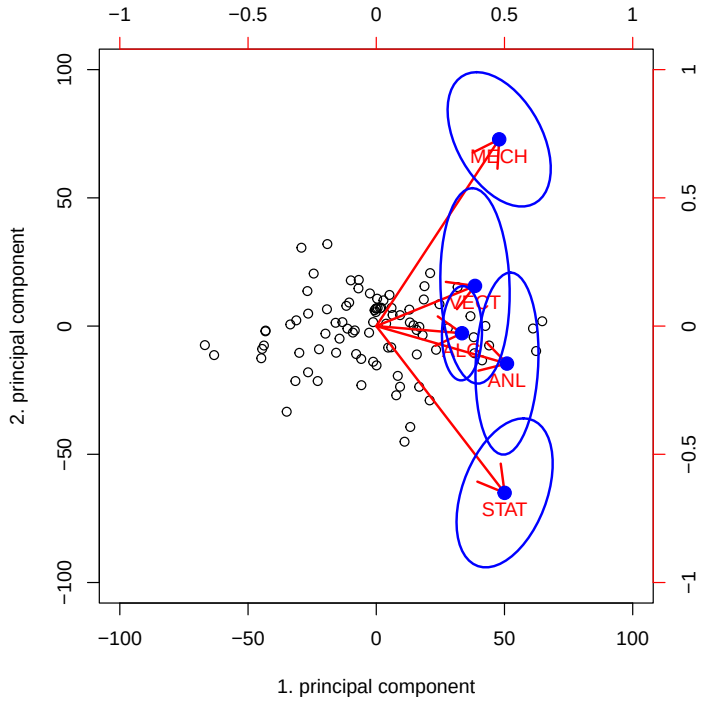


Figure 16: Biplot for the marks data set. Here the principal components are the principal components based on spatial signs. The red arrows are the loading vectors of the variables. The blue ellipses are the 0.95-confidence ellipses of the loading vectors.

6.2 Simulation for dimension 2

The 0.95-confidence ellipsoid for the eigenvectors could also be calculated. This could then for example be used to simulate how often the real eigenvector is in this 0.95-confidence ellipsoid. This could be used to see how well the convergence of the results of Theorem 4.1 and Theorem 4.2 work.

If formula (22) is used to calculate the 0.95-confidence ellipsoid. Then the covariance matrix is not invertible. The reason for this is that the norm of the eigenvectors is always one, hence the last coordinate of the eigenvector is determined, up to the sign, by the other coordinates. This problem can be solved by using polar coordinates. Now the delta-method is needed to then calculate the new asymptotic covariance matrix.

In [14] it is given that the following relation holds between Cartesian coordinates (x, y) and polar coordinates (r, ϕ)

$$\begin{aligned} x &= r\cos\phi \\ y &= r\sin\phi \\ r^2 &= x^2 + y^2 \\ \frac{y}{x} &= \tan\phi \end{aligned}$$

If we let the second coordinate of the eigenvector be positive and let the norm of the eigenvector be 1. Then the eigenvector is unique. Then the following function gives us the eigenvector in polar coordinates (r, ϕ).

$$f(v_1, v_2) = (r, \phi) = (1, \arccos(v_1))$$

Now we only need to know the last coordinate, because the radius is always 1. Define the function

$$h(v_1, v_2) = \arccos(v_1)$$

Then $\nabla h(v_1, v_2) = (-1/\sqrt{1 - v_1^2}, 0)^T$

Write $\Sigma = O\Lambda O^T$, $\hat{\Sigma} = GLG^T$, $\Sigma_s = O\Delta O^T$ and $\hat{\Sigma}_s = \hat{O}D\hat{O}^T$ as eigenvalue decompositions, with $\lambda_1 > \lambda_2 > 0$, $\delta_1 > \delta_2 > 0$, $l_1 > l_2 > 0$, $d_1 > d_2 > 0$ and $o_{i1} \geq 0, g_{i1} \geq 0, \hat{o}_{i1} \geq 0, i = 1, 2$. Define B_i as the asymptotic covariance matrix of $\sqrt{n}(g_i - o_i)$ and $\tilde{B}_i$ as the asymptotic covariance matrix of $\sqrt{n}(\hat{o}_i - o_i), i = 1, 2$. By Theorem 4.1 respectively Theorem 4.2

$$B_i = (1 + \kappa) \sum_{j=1, j\neq i}^{2} \frac{\lambda_j \lambda_i}{(\lambda_j - \lambda_i)^2} o_j o_j^T$$

$$\tilde{B}_i = \sum_{j=1, j\neq i}^{2} \frac{\eta_{ji}}{(\delta_j - \delta_i)^2} o_j o_j^T$$

Now by using Theorem 4.2.3 of [4] we get that asymptotic covariance of $\sqrt{n}(o_i - g_i), i = 1, 2$ in polar coordinates is

$$(\nabla h(o_i))^T B_i \nabla h(o_i)$$

This asymptotic covariance can be estimated by estimating o_i with g_i and λ_i with l_i. If the distribution is not known, then κ can be estimated with $\hat{\kappa}$, see equation (1) The asymptotic covariance of $\sqrt{n}(o_i - \hat{o}_i)$ in polar coordinates is

$$(\nabla h(o_i))^T \tilde{B}_i \nabla h(o_i)$$

This asymptotic covariance can be estimated by estimating o_i with $\hat{o}_i$ and δ_i and $\eta_{12} = \eta_{21}$ with using the formulas of Proposition 4.1 and Corollary 4.1 and estimate λ_i with l_i. Note here that Lemma 4.1 and lemma 4.2 also holds without the normalisation of Ψ_0 [5], hence Σ could be used in the lemma's instead of Ψ_0.

Simulating how often the real eigenvector is in the 0.95-confidence interval is done in the following way. Take a sample from an elliptical distribution, then calculate the confidence interval and check if the real eigenvector (in polar coordinates) is in it. This is then repeated 10000 times.

In the tabel below this is done for a multivariate normal distribution, with covariance matrix

$$\Sigma = \begin{pmatrix} 1/3 & 0 \\ 0 & 2/3 \end{pmatrix}.$$

It is done for the covariance matrix (classical case) and the spatial sign covariance matrix (spatial sign case).

	spatial signs		classic and assuming that $\kappa = 0$		classic and estimating κ	
Real eigenvector	$(0,1)^T$	$(1,0)^T$	$(0,1)^T$	$(1,0)^T$	$(0,1)^T$	$(1,0)^T$
n = 50	91.62%	91.01%	91.39%	91.29%	90.00%	89.00%
n = 100	93.13%	93.05%	92.90%	92.86%	92.44%	92.40%
n = 500	94.98%	94.98%	94.58%	94.58%	94.61%	94.61%
n = 1000	94.71%	94.71%	95.05%	95.05%	95.05%	95.05%

Table 4: Percentage that the real eigenvector is in the confidence interval of the eigenvector for p = 2 and a multivariate normal distribution, with covariance matrix $\Sigma = \begin{pmatrix} 1/3 & 0 \\ 0 & 2/3 \end{pmatrix}$.

It seems like it convergences in all cases to 95%, hence the convergence of the eigenvector seems to work well. It seems that for the classical case it converges faster towards 95% then for the spatial sign case.

In the tabel below it is done for a multivariate t-distribution with scatter matrix

$$\Psi = \begin{pmatrix} 1/3 & 0 \\ 0 & 2/3 \end{pmatrix}.$$

	spatial signs		classic and assuming that $\kappa = 2$		classic and estimating κ	
Real eigenvector	$(0,1)^T$	$(1,0)^T$	$(0,1)^T$	$(1,0)^T$	$(0,1)^T$	$(1,0)^T$
n = 50	87.33%	86.76%	94.18%	93.52%	84.26%	83.74%
n = 100	90.33%	90.29%	95.39%	95.28%	89.09%	89.00%
n = 500	94.01%	94.01%	96.42%	96.24%	93.73%	93.73%
n = 1000	94.51%	94.51%	95.94%	95.94%	94.11%	94.11%

Table 5: Percentage that the real eigenvector is in the confidence interval of the eigenvector for p = 2 and a multivariate t-distribution, where the degrees of freedom is 5 and the scatter matrix is $\Psi = \begin{pmatrix} 1/3 & 0 \\ 0 & 2/3 \end{pmatrix}$.

Here it also seems to converge towards 95%, exept if it is assumed that $\kappa = 2$, then it doesn't seem to converge. Here it seems like it converges faster towards 95% for the spatial sign case, then the classical case where κ is estimated.

6.3 Simulation for dimension 3

The same can be done for dimension 3, but now using spherical coordinates. In [14] it is given that the following relation holds between Cartesian coordinates (x, y, z) and spherical coordinates (R, θ, ϕ)

$$\begin{aligned} R &= x^2 + y^2 + z^2 \\ x &= R \sin\theta \cos\phi \\ y &= R \sin\theta \sin\phi \\ z &= R \cos\theta \end{aligned}$$

And note that $R\sin\theta = \sqrt{x^2+y^2}$. Again the eigenvectors have norm 1 and we take the second coordinate positive. Then the eigenvectors are unique. The following function then gives the eigenvector in spherical coordinates (R, θ, ϕ).

$$\tilde{f}(v_1, v_2, v_3) = (R, \theta, \phi) = \left(1, \arccos(v_3), \arccos\left(v_1/\sqrt{v_1^2+v_2^2}\right)\right)$$

Now only the last two coordinates are needed, because the radius is always 1. Define the following function

$$\tilde{h}(v_1, v_2, v_3) = \left(\arccos(v_3), \arccos\left(v_1/\sqrt{v_1^2+v_2^2}\right)\right)$$

Then

$$\nabla\tilde{h} = \begin{pmatrix} 0 & \frac{-v_2}{v_1^2+v_2^2} \\ 0 & \frac{v_1}{v_1^2+v_2^2} \\ \frac{-1}{\sqrt{1-v_3^2}} & 0 \end{pmatrix}$$

Write $\Sigma = O\Lambda O^T$, $\hat{\Sigma} = GLG^T$, $\Sigma_s = O\Delta O^T$ and $\hat{\Sigma}_s = \hat{O}D\hat{O}^T$ as eigenvalue decompositions, with $\lambda_1 > \lambda_2 > \lambda_3 > 0$, $\delta_1 > \delta_2 > \delta_3 > 0$, $l_1 > l_2 > l_3 > 0$, $d_1 > d_2 > d_3 > 0$ and $o_{i1} \geq 0, g_{i1} \geq 0, \hat{o}_{i1} \geq 0, i = 1, 2, 3$. Define E_i as the asymptotic covariance matrix of $\sqrt{n}(g_i - o_i)$ and $\tilde{E}_i$ as the asymptotic covariance matrix of $\sqrt{n}(\hat{o}_i - o_i), i = 1, 2, 3$. By Theorem 4.1 respectively Theorem 4.2

$$E_i = (1+\kappa)\sum_{j=1, j\neq i}^{3} \frac{\lambda_j\lambda_i}{(\lambda_j - \lambda_i)^2} o_j o_j^T$$

$$\tilde{E}_i = \sum_{j=1, j\neq i}^{3} \frac{\eta_{ji}}{(\delta_j - \delta_i)^2} o_j o_j^T$$

Now by using Theorem 4.2.3 of [4] we get that asymptotic covariance matrix of $\sqrt{n}(o_i - g_i), i = 1, 2, 3$ in spherical coordinates is

$$(\nabla\hat{h}(o_i))^T E_i \nabla\hat{h}(o_i)$$

This asymptotic covariance matrix can be estimated the same way as for dimension 2. The asymptotic covariance matrix of $\sqrt{n}(o_i - \hat{o}_i)$ in spherical coordinates is

$$(\nabla\hat{h}(o_i))^T \tilde{E}_i \nabla\hat{h}(o_i)$$

This asymptotic covariance can be estimated in the same way as for dimension 2, but now the integrals (5) and (8) are used to estimate δ_i and η_{ij}, instead of using Proposition 4.1 and Corollary 4.1. These integrals need to be solved numerically.

In the table below the simulation is done for a multivariate normal distribution, with covariance matrix

$$\Sigma = \begin{pmatrix} 1/6 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}.$$

Note here that the spherical coordinates for the vector $(0, 0, 1)^T$ is not defined, so we can not calculate the confidence interval/ellipse for this by using spherical coordinates.

	spatial signs		classic and assuming that $\kappa = 0$		classic and estimating κ	
Real eigenvector	$(0,1,0)^T$	$(1,0,0)^T$	$(0,1,0)^T$	$(1,0,0)^T$	$(0,1,0)^T$	$(1,0,0)^T$
n = 50	69.00%	76.38%	74.66%	80.97%	72.82%	79.14%
n = 100	79.00%	85.74%	83.84%	88.34%	83.04%	87.31%
n = 500	92.94%	94.25%	93.29%	94.17%	93.10%	93.96%
n = 1000	94.15%	94.77%	94.70%	95.05%	94.69%	94.92%

Table 6: Percentage that the real eigenvector is in the confidence ellipse of the eigenvector for p = 3 and a multivariate normal distribution, where the covariance matrix is $\Sigma = \begin{pmatrix} 1/6 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}$.

It seems like it convergences in all cases to 95%, hence the convergence of the eigenvector seems to work well. It seems that for the classical case it converges faster towards 95% then for the spatial sign case, which was also the case for dimension 2.

In the table below it is done for a multivariate t-distribution with scatter matrix

$$\Psi = \begin{pmatrix} 1/6 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}.$$

	spatial signs		classic and assuming that $\kappa = 2$		classic and estimating κ	
Real eigenvector	$(0,1,0)^T$	$(1,0,0)^T$	$(0,1,0)^T$	$(1,0,0)^T$	$(0,1,0)^T$	$(1,0,0)^T$
n = 50	64.91%	72.39%	74.36%	82.28%	60.10%	66.54%
n = 100	76.36%	82.91%	82.52%	89.35%	72.18%	78.88%
n = 500	90.97%	92.88%	93.67%	95.45%	89.23%	91.75%
n = 1000	92.90%	93.81%	95.21%	95.86%	92.36%	93.23%

Table 7: Percentage that the real eigenvector is in the confidence ellipse of the eigenvector for p = 3 and a multivariate t-distribution, where the degrees of freedom is 5 and the scatter matrix is $\Psi = \begin{pmatrix} 1/6 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}$.

Here it also seems to converge towards 95%, except if it is assumed that $\kappa = 2$, then it doesn't seem to converge towards 95%. Which was also the case for dimension 2. Here it seems like it converges faster towards 95% for the spatial sign case, then the classical case where κ is estimated.

It seems that for dimension 3 it takes a bit longer to converge towards 95% then for dimension 2. Which makes sense, because for dimension 3 there are more things that need to be estimated, hence more errors that are made.

7 Discussion

When a distribution has a large kurtosis it seems like it would be better to use principal components analysis based on spatial signs instead of using classical principal component analysis. Because as the kurtosis increases, then the MSE will also increase for the classical case, but the MSE for the spatial sign case is independent of the kurtosis. However when one eigenvector is really close to zero, we saw that the κ, for which the two MSEs are the same, increases. Hence when the kurtosis is really large and one of the eigenvalues is really close to zero, it is not always clear which MSE is smaller.

The MSE was simulated and then we saw that the asymptotic MSE we found indeed seems to converge towards the simulated MSE. There we saw that for dimensions 2 and 3 this seems to work well and seems to converge. However for dimension 12 the same was done, but then we saw that even at $n = 1000$ it doesn't seem to converge yet. So for a large dimension it may take really long to converge.

The asymptotic result that were found for the eigenvectors could be used to find a confidence ellipsoid for the eigenvectors. Using this showed us that the real eigenvector seems to converge to almost being in the 0.95-confidence ellipsoids, 95% of the time. For the multivariate normal distribution this seemed to converge faster by using the covariance matrix, then using the spatial sign covariance matrix. For the multivariate t-distribution, with degrees of freedom 5, this seemed to converge faster by using the spatial sign covariance matrix. This simulation was only done for dimensions 2 and 3. It seemed like it converges faster to 95% for dimension 2, then dimension 3. So it might be a good idea to also check this for higher dimension. This is however not so easy to do, because you first need to use another coordinate system and then go to one dimension lower by using the fact that the norm is 1. For dimensions 2 and 3 this could be done by using polar and spherical coordinates. For higher dimension we first need to find a coordinate system that works for this. We also saw that the vector $(0, 0, 1)$ isn't defined in spherical coordinates, hence we couldn't do the simulation for this eigenvector.

8 Conclusion

Firstly principal component analysis and principal component analysis based on spatial signs were explained. Then we looked at some asymptotic results of the principal components that were already known. Then similar asymptotic results of the eigenvectors and eigenvalues of the sample spatial sign covariance matrix were proven. The asymptotic result of the eigenvectors could then be used to find the asymptotic MSE. Then the MSEs for the two cases were compared for dimensions 2, 3 and 4. For dimension two we even found an explicit formula for this. One of the differences between the MSEs we have seen is that the MSE for the classical case is linearly dependent on the kurtosis and the MSE of the spatial sign case is independent of the kurtosis. As the kurtosis increases the MSE for the classical case also increases and the MSE for the spatial sign case doesn't change. So for a large kurtosis we would expect principal component analysis based on spatial signs to work better, then classical principal component analysis. Even for a really large kurtosis principal component analysis based on spatial signs works equally well as for a small kurtosis. Lastly we looked at confidence ellipsoids. Where we gave an example how this could be used to find the confidence ellipses of loading vectors. Which could then be used in analyzing a data set. We also used confidence ellipsoids to check how well the asymptotic result of the eigenvectors work for dimensions 2 and 3. The percentage of the real eigenvectors that is in the 95%-confidence interval or ellipse seems to converge towards 95%, hence it seems to work well for dimensions 2 and 3.

References

[1] Wolfgang Hardle. *Applied multivariate statistical analysis*. Springer, Berlin [etc.], 2003.

[2] I.T Jolliffe. *Principal component analysis*. Springer series in statistics. Springer, New York, 2nd ed. edition, 2002.

[3] Alexander Dürre, Daniel Vogel, and Roland Fried. Spatial sign correlation. *Journal of multivariate analysis*, 135:89–105, 2015.

[4] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley Series in Probability and Statistics. Wiley, 2003.

[5] Alexander Dürre, David E. Tyler, and Daniel Vogel. On the eigenvalues of the spatial sign covariance matrix in more than two dimensions. *Statistics & probability letters*, 111:80–85, 2016.

[6] Sara Taskinen, Inge Koch, and Hannu Oja. Robustifying principal component analysis with spatial sign vectors. *Statistics & probability letters*, 82(4):765–774, 2012.

[7] Christophe Croux, Esa Ollila, and Hannu Oja. Sign and rank covariance matrices: Statistical properties and application to principal components analysis. In *Statistical Data Analysis Based on the L1-Norm and Related Methods*, Statistics for Industry and Technology, pages 257–269. Birkhuser Basel, Basel, 2002.

[8] Hannu Oja. *Multivariate nonparametric methods with R an approach based on spatial signs and ranks*. Lecture notes in statistics, 199. Springer, New York [etc.], 2010.

[9] Vctor H. Lachos, Edgar J. Lpez Moreno, Kun Chen, and Celso Rmulo Barbosa Cabral. Finite mixture modeling of censored data using the multivariate student-t distribution. *Journal of multivariate analysis*, 159:151–167, 2017.

[10] Aldo M Garay, Luis M Castro, Jacek Leskow, and Victor H Lachos. Censored linear regression models for irregularly observed longitudinal data using the multivariate-t distribution. *Statistical methods in medical research*, 26(2):542–566, 2017.

[11] Fetsje Bijma. *Inleiding in de statistiek*. Epsilon uitgaven ; deel 76, tweede druk. edition, 2016.

[12] Hamid Reza Kamali and Parisa Shahnazari-Shahrezaei. Estimating the non-existent mean and variance of the f-distribution by simulation. *Journal of modern applied statistical methods*, 9(2):599–603, 2010.

[13] Mlnoru Slotani. Tolerance regions for a multivariate normal population. *Annals of the Institute of Statistical Mathematics*, 16(1):135–153, 1964.

[14] Robert A. Adams. *Calculus : a complete course*. Pearson, Toronto, eighth edition. edition, 2013.

A R Code

R-code for the plost in figure 1 and 2.

```
#contour plots for the multivariate normal distribution
#and the multivariate t-distribution.

library(mvtnorm)

#mulivariate normal distribution
pdf("contour_normal.pdf",width=6,height=6)
x <- seq(-3.5,3.5,length.out=200)
y <- x
z <- matrix(0,nrow=200,ncol=200)
mu <- c(0,0) #mean mu
sigma <- matrix(c(1,0,0,2),nrow=2) #covariance sigma
for (i in 1:200) {
  for (j in 1:200) {
    z[i,j] <- dmvnorm(c(x[i],y[j]),
                      mean=mu,sigma=sigma, log = FALSE) #density
  }
}
contour(x,y,z, levels=seq(from=0.01,to=0.11,by=0.01)) #contour plot
dev.off()

#multivariate t-distribution
pdf("contour_t.pdf",width=6,height=6)
x <- seq(-3.5,3.5,length.out=200)
y <- x
z <- matrix(0,nrow=200,ncol=200)
mu <- c(0,0) #mean mu
sigma <- matrix(c(1,0,0,2),nrow=2) #covariance sigma
for (i in 1:200) {
  for (j in 1:200) {
    z[i,j] <- dmvt(c(x[i],y[j]),
                      delta=mu,sigma=sigma, df = 5, log = FALSE) #density
  }
}
contour(x,y,z, levels=seq(from=0.01,to=0.11,by=0.01)) #contour plot
dev.off()

#both in one plot
pdf("contour_t_and_normal.pdf",width=6,height=6)
x1 <- seq(-3.5,3.5,length.out=200)
y1 <- x
z1 <- matrix(0,nrow=200,ncol=200)
mu <- c(0,0) #mean mu
sigma <- matrix(c(1,0,0,2),nrow=2) #covariance sigma
for (i in 1:200) {
  for (j in 1:200) {
    z1[i,j] <- dmvnorm(c(x1[i],y1[j]),
                      mean=mu,sigma=sigma, log = FALSE) #density
  }
}
x2 <- seq(-3.5,3.5,length.out=200)
y2 <- x
z2 <- matrix(0,nrow=200,ncol=200)
```

```
mu <- c(0,0) #mean mu
sigma <- matrix(c(1,0,0,2),nrow=2) #covariance sigma
for (i in 1:200) {
  for (j in 1:200) {
    z2[i,j] <- dmvt(c(x2[i],y2[j]),
                    delta=mu,sigma=sigma, df = 5, log = FALSE) #density
  }
}
contour(x1,y1,z1, levels=seq(from=0.01,to=0.11,by=0.01), col = 'red') #contour plot
contour(x2,y2,z2, add=TRUE, levels=seq(from=0.01,to=0.11,by=0.01), col = "blue") #contou
dev.off()
```

R-code for the plots in table 1 and 2

```
#Making plot with simulated and asymptotic MSE

library(mvtnorm)
library(sscor)
library(ICSNP)
d <- 2 #dimension
m <- 10000 #repeat m times
mu <- c(0,0) #mean
df <- 5 #degrees of freedom
#the n (number of samples) we want to plot
n_vector <- c(20:200)

#covariance matrix: (this could be changed to something else)
sigma <- matrix(c(1,0,0,2), ncol=d)

sum <- c(rep(0,m))
sum_ss <- c(rep(0,m))
MSE <- c(rep(0,200))
MSE_ss <- c(rep(0,200))

E_Y <- eigen(sigma)$vectors #eigenvectors of sigma


for (n in n_vector) {
  set.seed(1)
  sum <- c(rep(0,m))
  sum_ss <- c(rep(0,m))
  for (k in 1:m) {
    #taking samples from the multivariate normal distribution
    x <- rmvnorm(n, mean=mu, sigma=sigma)
    #taking samples from the multivariate t distribution
    #(the line below could be uncommented, if we want to use the mult t-distribution)
    #x <- rmvt(n=n, sigma = sigma, df = 5, delta = mu)
    mean <- colMeans(x) #colum means of x
    x_standarized <- x-rep(mean, each = nrow(x)) #x standarized w.r.t location

    emp_cov <-cov(x_standarized) #empirical/sample covariance matrix
    V0 <- emp_cov/sum(diag(emp_cov)) #emp_cov matrix scaled such that the trace is 1
    Y <- eigen(V0)$vectors #eigenvectors of V0 (estimated eigenvectors)

    #norm^2 of difference between theoretical eigenvectors and estimated eigenvectors
    for (i in 1:d) {
      sum[k] <- sum[k] + min(norm((Y[,i]-E_Y[,i])),norm((Y[,i]+E_Y[,i])))^2
```

```
    }

    #spatial sign of x standarized with spatial median
    x_ss <- spatial.sign(x, center = TRUE, shape = FALSE)

    #covariance matrix of x_ss (Spatial sign covariance matrix)
    SSCM <- cov(x_ss)
    SSCM <- SSCM/sum(diag(SSCM)) #scaling SSCM such that the trace is 1
    #eigenvectors of SSCM (estimated eigenvectors based on spatial signs)
    Y_ss <- eigen(SSCM)$vectors

    #norm^2 of difference between theoretical eigenvectors
    #and estimated eigenvectors of SSCM
    for (i in 1:d) {
      sum_ss[k] <- sum_ss[k] +
                   min(norm((Y_ss[,i]-E_Y[,i])),norm((Y_ss[,i]+E_Y[,i])))^2
    }

  }
  MSE[n] <- 1/m * sum(sum) #MSE
  MSE_ss[n] <- 1/m * sum(sum_ss) #MSE based on spatial signs
}



lambda <- eigen(sigma)$values #eigenvalues of sigma the covariance matrix
V <- 0
#computing the asymptotic MSE (from theorem 5.3) multiplied by n
for (j in 1:d) {
  for (k in 1:d) {
    if(k != j){
      V <- V + (lambda[j]*lambda[k])/(lambda[k]-lambda[j])^2
    }
  }
}

#plot of MSE simulation and approximation of the MSE by using theorem 5.1
x = MSE[n_vector]
pdf(file = "MSE.pdf")
plot(n_vector, x, col ='red', type ='l',
     xlab="n", ylab="MSE")
lines(n_vector, V/(n_vector-1), col = 'green')
legend("topright",    # Position
       inset = 0.05, # Distance from the margin as a fraction of the plot region
       legend = c("MSE simulation", "MSE approximation"),
       lty = c(1, 1),
       col = c(2, 3),
       lwd = 2)
dev.off()

#calculating eta
nij <- function(x,i,j) {z <- 1; for (k in 1:d) {
  z <- z*(1+lambda[k]*x)^0.5

} ;(lambda[i]*lambda[j]*0.25*x)/((1+lambda[i]*x)*(1+lambda[j]*x)*z)}
```

```
integrate(nij,0,Inf, i =1, j=2)

#calculating delta
deltai <- function(x,i) {
  z <- 1; for (k in 1:d) {
    z <- z*(1+lambda[k]*x)^0.5
  } ; (lambda[i]*0.5)/((1+lambda[i]*x)*z)
}

integrate(deltai,0,Inf, i =1)

#computing the asymptotic MSE (from theorem 5.2) multiplied by n
V_ss <-0
for (a in 1:d) {
  for (b in 1:d) {
    if(b != a){
      V_ss <- V_ss +(integrate(nij,0,Inf, i =a, j=b)$value/
                       (integrate(deltai,0,Inf, i =a)$value-
                          integrate(deltai,0,Inf, i =b)$value)^2)
    }
  }
}

#plot of MSE simulation and approximating MSE with the asymptotic MSE (of the theorem)
#for the spatial sign case
y = MSE_ss[n_vector]
pdf(file = "MSE_ss.pdf")
plot(n_vector, y, col ='red', type ='l',
     xlab="n", ylab="MSE", lwd=2.0, cex.axis = 1, cex.lab = 1)
lines(n_vector, V_ss/(n_vector-1), col = 'green', lwd=3.0)
legend("topright",   # Position
       inset = 0.05, # Distance from the margin as a fraction of the plot region
       legend = c("classical", "spatial_signs"),
       lty = c(1, 1),
       col = c(2, 3),
       lwd = 2,
       cex=1)
dev.off()
```

R-code for the plots in figures 4, 5, 6, 7 and 8.

```
#plot MSE_c = MSE_ss for dimension 3
#fixed lambda_1

kappa <- c(rep(0,100))
MSE_approx_c <- c(rep(0,100))
MSE_approx_ss<-c(rep(0,100))
#the value 1/6 can be changed to something else to get
#a plot with another fixed lambda_1
lambda1 <- 1/6 #fixed lambda_1 (first eigenvalue)
n <- seq(0.001, 1-lambda1-0.001,length=100)
for (i in 1:100) {
  lambda <- c(lambda1, n[i], 1-lambda1-n[i]) #lambda_2 the second eigenvalue
  lambda
  #computing asymptotic MSE (from theorem 5.1) multiplied by n
  for (j in 1:3) {
    for (k in 1:3) {
```

```
      if(k != j){
        MSE_approx_c [i] <- MSE_approx_c [i] + (lambda[j]*lambda[k])/
                            (lambda[k]-lambda[j])^2
      }
    }
  }
  #computing eta
  nij <- function(x,i,j) {z <- 1; for (k in 1:3) {
    z <- z*(1+lambda[k]*x)^0.5
  } ;(lambda[i]*lambda[j]*0.25*x)/((1+lambda[i]*x)*(1+lambda[j]*x)*z)}

  #computing delta
  deltai <- function(x,i) {
    z <- 1; for (k in 1:3) {
      z <- z*(1+lambda[k]*x)^0.5
    } ; (lambda[i]*0.5)/((1+lambda[i]*x)*z)
  }

  #computing asymptotic MSE (from theorem 5.2) multiplied by n
  #spatial sign case
  for (a in 1:3) {
    for (b in 1:3) {
      if(b != a){
        MSE_approx_ss[i] <- MSE_approx_ss[i] +
          (integrate(nij,0,Inf, i =a, j=b)$value/(integrate(deltai,0,Inf, i =a)$value
            -integrate(deltai,0,Inf,i=b)$value)^2)
      }
    }
  }
  #computing kappa such that MSE_c = MSE_ss
  kappa[i] <- MSE_approx_ss[i]/MSE_approx_c[i] -1
}

MSE_approx_c
MSE_approx_ss
kappa

#plot MSE_c = MSE_ss
pdf("1_6_p=3.pdf",width=6,height=6)
plot(n[1:100], kappa[1:100], xlab = "lambda_2", ylab = "kappa")
dev.off()
```

R-code for the plot in figure 9.

```
#plot MSE_c = MSE_ss for dimension 3
#3D plot

x<-0
y<-0
z<-0
kappa <- 0
MSE_approx_c <- 0
MSE_approx_ss<-0
n <- seq(0, 1, length=200)
for (i in 2:199) {
  for (l in 2:199) {
    lambda_3 <- n[i] + n[l] #lambda_3 (third eigenvalue)
```

```
    #n[i] is the first eigenvalue and n[l] is the second eigenvalue
    if(n[i]+n[l] <1 & i != l & n[i] != 1- n[i] - n[l] & n[l] != 1- n[i] -n[l]){
      kappa <- 0
      MSE_approx_c <- 0
      MSE_approx_ss<-0
      lambda <- c(n[i], n[l], 1-n[l]-n[i]) #lambda = (lambda_1,lambda_2,lambda_3)
      #computing asymptotic MSE (from theorem 5.1) multiplied by n
      for (j in 1:3) {
        for (k in 1:3) {
          if(k != j){
            MSE_approx_c <- MSE_approx_c + (lambda[j]*lambda[k])/
              (lambda[k]-lambda[j])^2
          }
        }
      }
      #computing eta
      nij <- function(x,i,j) {z <- 1; for (k in 1:3) {
        z <- z*(1+lambda[k]*x)^0.5
      } ;(lambda[i]*lambda[j]*0.25*x)/((1+lambda[i]*x)*(1+lambda[j]*x)*z)}

      #computing delta
      deltai <- function(x,i) {
        z <- 1; for (k in 1:3) {
          z <- z*(1+lambda[k]*x)^0.5
        } ; (lambda[i]*0.5)/((1+lambda[i]*x)*z)
      }

      #computing asymptotic MSE (from theorem 5.2) multiplied by n
      for (a in 1:3) {
        for (b in 1:3) {
          if(b != a){
            MSE_approx_ss <- MSE_approx_ss +
            (integrate(nij,0,Inf, i=a, j=b)$value/(integrate(deltai,0,Inf, i=a)$value
             -integrate(deltai,0,Inf,i=b)$value)^2)
          }
        }
      }
      if(is.infinite(MSE_approx_ss/MSE_approx_c -1) ==
         FALSE & is.nan(MSE_approx_ss/MSE_approx_c -1) == FALSE){
        x <- c(x, n[i]) #lambda_1 on x-axis
        y <- c(y, n[l]) #lambda_2 on y-axis
        z <- c(z, MSE_approx_ss/MSE_approx_c -1) #kappa on z-axis
      }

    }
  }
}
#plot MSE_c = MSE_ss
pdf(file = "p=3.pdf")
source('http://www.sthda.com/sthda/RDoc/functions/addgrids3d.r')
scatterplot3d(x[2:length(x)], y[2:length(y)], z[2:length(z)],
              xlab = "lambda_1", ylab = "lambda_2", zlab = "kappa")
addgrids3d(x[2:length(x)], y[2:length(y)], z[2:length(z)],
           grid = c("xy", "xz", "yz"))
dev.off()
```

R-code for the plots in table 3.

```
#plot MSE_c = MSE_ss for dimension 4
#fixed lambda_1 and lamda_2

kappa <- c(rep(0,100))
MSE_approx_c <- c(rep(0,100))
MSE_approx_ss<-c(rep(0,100))
#the values 1/6 and 2/6 can be changed to something else to get a plot with
#another fixed lambda_1 and lambda_2
lambda1 <- 1/6 #fixed lambda_1
lambda2 <- 2/6 #fixed lambda_2
n <- seq(0.001, 1-lambda1 - lambda2-0.001,length=100)
for (i in 1:100) {
  #lambda = (lambda_1, lambda_2, lambda_3, lambda_4)
  lambda <- c(lambda1, lambda2, n[i], 1-lambda1-lambda2-n[i])
  lambda
  #computing asymptotic MSE (from theorem 5.1) multiplied by n
  for (j in 1:4) {
    for (k in 1:4) {
      if(k != j){
        MSE_approx_c [i] <- MSE_approx_c [i] +
          (lambda[j]*lambda[k])/(lambda[k]-lambda[j])^2
      }
    }
  }
  #computing eta
  nij <- function(x,i,j) {z <- 1; for (k in 1:4) {
    z <- z*(1+lambda[k]*x)^0.5
  } ;(lambda[i]*lambda[j]*0.25*x)/((1+lambda[i]*x)*(1+lambda[j]*x)*z)}

  #computing delta
  deltai <- function(x,i) {
    z <- 1; for (k in 1:4) {
      z <- z*(1+lambda[k]*x)^0.5
    } ; (lambda[i]*0.5)/((1+lambda[i]*x)*z)
  }

  #computing asymptotic MSE (from theorem 5.2) multiplied by n
  for (a in 1:4) {
    for (b in 1:4) {
      if(b != a){
        MSE_approx_ss[i] <- MSE_approx_ss[i] +
          (integrate(nij,0,Inf, i =a, j=b)$value/
             (integrate(deltai,0,Inf, i =a)$value-
                integrate(deltai,0,Inf,i=b)$value)^2)
      }
    }
  }
  #computing kappa such that MSE_c = MSE_ss
  kappa[i] <- MSE_approx_ss[i]/MSE_approx_c[i] -1
}

MSE_approx_c
MSE_approx_ss
kappa

#plot MSE_c = MSE_ss
```

```
pdf("2_3_p=4.pdf",width=6,height=6)
plot(n[1:100], kappa[1:100], xlab = "lambda_3", ylab = "kappa")
dev.off()
```

R-code for the plots in figures 10, 11, 12, 13 and 14.

```
#3D plot MSE_c = MSE_ss for dimension 4
#fixed lambda_1

x<-0
y<-0
z<-0
kappa <- 0
MSE_approx_c <- 0
MSE_approx_ss<-0
#the values 1/6 can be changed to something else to get
#a plot with another fixed lambda_1
lambda_1 <- 1/6 #fixed lambda_1
n <- seq(0, 1, length=200)
for (i in 2:199) {
  for (l in 2:199) {
    lambda_3 <- n[i] + n[l]
    if(n[i]+n[l] + lambda_1 <1 & i != l & n[i] != 1- n[i] - n[l]
       - lambda_1 & n[l] != 1- n[i] -n[l] - lambda_1){
      kappa <- 0
      MSE_approx_c <- 0
      MSE_approx_ss<-0
      #lambda = (lambda_1, lambda_2, lambda_3, lambda_4)
      lambda <- c(lambda_1, n[i], n[l], 1-n[l]-n[i]-lambda_1)
      #computing asymptotic MSE (from theorem 5.1) multiplied by n
      for (j in 1:4) {
        for (k in 1:4) {
          if(k != j){
            MSE_approx_c <- MSE_approx_c +
              (lambda[j]*lambda[k])/(lambda[k]-lambda[j])^2
          }
        }
      }
      #computing eta
      nij <- function(x,i,j) {z <- 1; for (k in 1:4) {
        z <- z*(1+lambda[k]*x)^0.5
      } ;(lambda[i]*lambda[j]*0.25*x)/((1+lambda[i]*x)*(1+lambda[j]*x)*z)}

      #computing delta
      deltai <- function(x,i) {
        z <- 1; for (k in 1:4) {
          z <- z*(1+lambda[k]*x)^0.5
        } ; (lambda[i]*0.5)/((1+lambda[i]*x)*z)
      }

      #computing asymptotic MSE (from theorem 5.2) multiplied by n
      for (a in 1:4) {
        for (b in 1:4) {
          if(b != a){
            MSE_approx_ss <- MSE_approx_ss +
              (integrate(nij,0,Inf, i =a, j=b)$value/
                 (integrate(deltai,0,Inf, i =a)$value
```

```
                    -integrate(deltai,0,Inf,i=b)$value)^2)
            }
          }
        }
        if(is.infinite(MSE_approx_ss/MSE_approx_c -1) ==
           FALSE & is.nan(MSE_approx_ss/MSE_approx_c -1) == FALSE){
          x <- c(x, n[i]) #lambda_1 on x-axis
          y <- c(y, n[l]) #lambda_2 on y-axis
          z <- c(z, MSE_approx_ss/MSE_approx_c -1) #kappa on z-axis
        }

    }
  }
}
#plot of MSE_c = MSE_ss
pdf("1_p=4.pdf",width=6,height=6)
source('http://www.sthda.com/sthda/RDoc/functions/addgrids3d.r')
scatterplot3d(x[2:length(x)], y[2:length(y)], z[2:length(z)],
              xlab = "lambda_2", ylab = "lambda_3", zlab = "kappa")
addgrids3d(x[2:length(x)], y[2:length(y)], z[2:length(z)],
           grid = c("xy", "xz", "yz"))
dev.off()
```

R-code for figure 15.

```
#data example using confidence ellipsoids

library(bnlearn)
data(marks)
n <- length(marks[,1])

# calculation of eigenvalues
eigenm <- eigen(cov(marks))

mean <- colMeans(marks) #mean
kappa <- 0
marksmatrix <- as.matrix(marks)
#estimating kappa (you could also take an kappa and then
#uncomment the following 6 lines)
for (i in 1:88) {
  kappa <-kappa +
    (t(marksmatrix[i, ]- mean)%*%inv(cov(marks))%*%(marksmatrix[i, ]-mean))^2
}
kappa
kappa <- 1/(5*(5+2)*88)*kappa -1
kappa <- kappa[1,1]

# looks nicer if we make loadings of first principal component positive
eigenm$vector[,1] <- eigenm$vector[,1]*(-1)

# plotting principal components
pdf("biplotkappa.pdf",width=6,height=6)
par(mar=c(4,4,2.5,2.5))

plot(c(-100,100),c(-100,100),type="n",xlab="1. principal component",
     ylab="2. principal component")
Xval <- (as.matrix(marks)
```

```
        -matrix(apply(marks,2,mean),ncol=5,nrow=n,byrow=TRUE))%*%eigenm$vector[,1]
Yval <- (as.matrix(marks)
        -matrix(apply(marks,2,mean),ncol=5,nrow=n,byrow=TRUE))%*%eigenm$vector[,2]
points(Xval,Yval)

# now we add the direction of the principal components
#(multiply everything with 100 to make it look nicer)
for(i in 1:5) {
  arrows(x0=0,x1=eigenm$vector[i,1]*100,y0=0,
         y1=eigenm$vector[i,2]*100,lwd=2,col="red")
  text(eigenm$vector[i,1]*100,eigenm$vector[i,2]*100,
       labels=colnames(marks)[i],pos=1,col="red",cex=1)
}

# calculation of the asymptotic covariances
Cov1 <- Cov2 <- Cov12 <- matrix(0,ncol=5,nrow=5)

# covariance of loadings of first principal component
for (i in 2:5) Cov1 <- Cov1+(eigenm$values[i]*eigenm$values[1]/
                               (eigenm$values[i]-eigenm$values[1])^2
                             *eigenm$vector[,i]%*%t(eigenm$vector[,i]))

Cov1 <- Cov1*(1+kappa)
# covariance of loadings of second principal component
for (i in c(1,3,4,5)) Cov2 <- Cov2+(eigenm$values[i]*eigenm$values[2]/
                                      (eigenm$values[i]-eigenm$values[2])^2
                                    *eigenm$vector[,i]%*%t(eigenm$vector[,i]))

Cov2 <- Cov2*(1+kappa)
# covariance between loadings of first and second rincipal component
Cov12 <- -(eigenm$values[1]*eigenm$values[2]/
             (eigenm$values[1]-eigenm$values[2])^2
           *eigenm$vector[,1]%*%t(eigenm$vector[,2]))
Cov12 <- Cov12*(1+kappa)
# draw ellipses
library(car)
for (i in 1:5) {
  # one has to give the center and the covariance matrix
  #and the radius which is the squareroot of the quantile
  ellipse(center=c(eigenm$vector[i,1:2])*100,
          shape=matrix(c(Cov1[i,i],Cov12[i,i],Cov12[i,i],Cov2[i,i]),ncol=2)/
            n*100^2,radius=sqrt(qchisq(0.95,2)))
}
axis(side=3,col="red",at=seq(from=-100,to=100,by=50),labels=seq(from=-1,to=1,b=0.5))
axis(side=4,col="red",at=seq(from=-100,to=100,by=50),labels=seq(from=-1,to=1,b=0.5))


dev.off()
```

R-code for figure 16.

```
library(bnlearn)
library(ICSNP)
data(marks)
n <- length(marks[,1])

# calculation of eigenvalues (now of spatial sign covariance matrix)
```

```
eigenm <- eigen(cov(spatial.sign(marks, center = TRUE, shape = FALSE) ))

#also need eigenvalues of covariance matrix
lambda <- eigen(cov(marks))$values

# looks nicer if we make loadings of first principal component positive
eigenm$vector[,1] <- eigenm$vector[,1]*(-1)

# plotting principal components
pdf("biplot_ss_2.pdf",width=6,height=6)
par(mar=c(4,4,2.5,2.5))

#Here we now have use the data standarized with spatial median
sm <- spatial.median(marks)
plot(c(-100,100),c(-100,100),type="n",xlab="1. principal component",
     ylab="2. principal component")
Xval <- (as.matrix(marks)-rep(sm, each = nrow(marks)))%*%eigenm$vector[,1]
Yval <- (as.matrix(marks)-rep(sm, each = nrow(marks)))%*%eigenm$vector[,2]
points(Xval,Yval)

# now we add the direction of the principal components
#(multiply everything with 100 to make it look nicer)
for(i in 1:5) {
  arrows(x0=0,x1=eigenm$vector[i,1]*100,y0=0,y1=eigenm$vector[i,2]*100,lwd=2,col="red")
  text(eigenm$vector[i,1]*100,eigenm$vector[i,2]*100,
       labels=colnames(marks)[i],pos=1,col="red",cex=1)
}

# calculation of the asymptotic covariances
Cov1 <- Cov2 <- Cov12 <- matrix(0,ncol=5,nrow=5)

#calculating delta and eta
d = 5
#eta
nij <- function(x,i,j) {z <- 1; for (k in 1:d) {
  z <- z*(1+lambda[k]*x)^0.5

} ;(lambda[i]*lambda[j]*0.25*x)/((1+lambda[i]*x)*(1+lambda[j]*x)*z)}
eta1 <- eta2 <- c(0,0,0,0,0)
for (k in 2:5) {
  eta1[k] <- integrate(nij,0,Inf, i =1, j=k)$value
}
for (k in c(1,3,4,5)) {
  eta2[k] <- integrate(nij,0,Inf, i =2, j=k)$value
}
#delta
delta <- c(0,0,0,0,0)
deltai <- function(x,i) {
  z <- 1; for (k in 1:d) {
    z <- z*(1+lambda[k]*x)^0.5
  } ; (lambda[i]*0.5)/((1+lambda[i]*x)*z)
}
for (k in 1:5) {
  delta[k] <- integrate(deltai,0,Inf, i =k)$value
}
```

```
# covariance of loadings of first principal component
for (i in 2:5) Cov1 <- Cov1+eta1[i]/
  (delta[i]-delta[1])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

# covariance of loadings of second principal component
for (i in c(1,3,4,5)) Cov2 <- Cov2+eta2[i]/
  (delta[i]-delta[2])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

# covariance between loadings of first and second rincipal component
Cov12 <- -eta1[2]/(delta[1]-delta[2])^2*eigenm$vector[,1]%*%t(eigenm$vector[,2])

# draw ellipses
library(car)
for (i in 1:5) {
  # one has to give the center and the covariance matrix
  #and the radius which is the squareroot of the quantile
  ellipse(center=c(eigenm$vector[i,1:2])*100,
          shape=matrix(c(Cov1[i,i],Cov12[i,i],Cov12[i,i],Cov2[i,i]),ncol=2)/
            n*100^2,radius=sqrt(qchisq(0.95,2)))
}
axis(side=3,col="red",at=seq(from=-100,to=100,by=50),labels=seq(from=-1,to=1,b=0.5))
axis(side=4,col="red",at=seq(from=-100,to=100,by=50),labels=seq(from=-1,to=1,b=0.5))


dev.off()
```

R-code for the spatial signs of tables 4, 5, 6 and 7

```
#simulating how often the real eigenvector is in the 95% confidence interval
#spatial sign case

set.seed(3)


library(bnlearn)
library(ICSNP)
library(matlib)
#for dimension 2
d <- 2 #dimension
n <- 50 #n number of samples
tel1 <- 0
tel2 <- 0
mu <-c(0,0) #mu
sigma <-sigma <- matrix(diag(c(1/3,2/3)), ncol =d) #sigma (scatter matrix)

for (m in 1:10000) {


vector <- eigen(sigma)$vectors #eigenvectors of sigma

#taking samples from the multivariate normal distribution
x <- rmvnorm(n, mean=mu, sigma=sigma)
#taking samples from the multivariate t distribution
#if we want to do it for the multivariate t distribution the following line
#could be uncommented
#x <- rmvt(n=n, sigma = sigma, df = 5, delta = mu)
```

```
# calculation of eigenvalues (now of spatial sign covariance matrix)
eigenm <- eigen(cov(spatial.sign(x, center = TRUE, shape = FALSE) ))

#also need eigenvalues of covariance matrix
mean <- colMeans(x) #colum means of x
lambda <- eigen(cov(x-rep(mean, each = nrow(x))))$values #eigenvalues

# calculation of the asymptotic covariances
Cov1 <- Cov2 <- Cov12 <- matrix(0,ncol=2,nrow=2)

#calculating delta and eta
#eta
nij <- function(x,i,j) {z <- 1; for (k in 1:d) {
  z <- z*(1+lambda[k]*x)^0.5

} ;(lambda[i]*lambda[j]*0.25*x)/((1+lambda[i]*x)*(1+lambda[j]*x)*z)}
eta1 <- eta2 <- c(0,0,0,0,0)
for (k in 2:d) {
  eta1[k] <- integrate(nij,0,Inf, i =1, j=k)$value
}
for (k in c(1)) {
  eta2[k] <- integrate(nij,0,Inf, i =2, j=k)$value
}
#delta
delta <- c(0,0,0,0,0)
deltai <- function(x,i) {
  z <- 1; for (k in 1:d) {
    z <- z*(1+lambda[k]*x)^0.5
  } ; (lambda[i]*0.5)/((1+lambda[i]*x)*z)
}
for (k in 1:d) {
  delta[k] <- integrate(deltai,0,Inf, i =k)$value
}

# covariance of loadings of first principal component
for (i in 2:d) Cov1 <- Cov1+eta1[i]/
  (delta[i]-delta[1])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

# covariance of loadings of second principal component
for (i in c(1)) Cov2 <- Cov2+eta2[i]/
  (delta[i]-delta[2])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

# covariance between loadings of first and second rincipal component
Cov12 <- -eta1[2]/(delta[1]-delta[2])^2*eigenm$vector[,1]%*%t(eigenm$vector[,2])

#making the second coordinate of the eigenvectors positive
if (eigenm$vectors[2,1]<0){
  eigenm$vectors[,1]*-1
}
if (eigenm$vectors[2,2]<0){
  eigenm$vectors[,2]*-1
}

#calculating polar coordinated of eigenvectors
eigen_polar <- c(0,0)
for(i in c(1,2)){
```

```
  eigen_polar[i] <- acos(eigenm$vectors[1,i])
}

#making the second coordinate of the real eigenvectors positive
if (vector[2,1]<0){
  vector[,1]*-1
}
if (vector[2,2]<0){
  vector[,2]*-1
}

#calculating gradient of h
h1 <- c(-1/(sqrt(1-eigenm$vectors[1,1]^2)), 0)
h2 <- c(-1/(sqrt(1-eigenm$vectors[1,2]^2)), 0)

Cov1 <- Cov1/n
Cov2 <- Cov2/n

NewCov1 <- t(h1)%*%Cov1%*%h1
NewCov2 <- t(h2)%*%Cov2%*%h2

#calculating polar coordinates of real eigenvectors
vector_polar <- c(0,0)
for(i in c(1,2)){
  vector_polar[i] <- atan2(vector[2,i], vector[1,i])
}

#checking if the real eigenvector is in the confidence interval
if(eigen_polar[1]- 1.96* sqrt(NewCov1[1,1]) <= vector_polar[1] & vector_polar[1]
   <= eigen_polar[1]+ 1.96 * sqrt(NewCov1[1,1]) ){
  tel1 <- tel1 + 1
}
if(eigen_polar[2]- 1.96* sqrt(NewCov2[1,1]) <= vector_polar[2] & vector_polar[2]
   <= eigen_polar[2]+ 1.96* sqrt( NewCov2[1,1] )){
  tel2 <- tel2 + 1
}
}
#number of times the real eigenvectors are in the confidence interval
tel1
tel2

#for dimension 3
set.seed(3)
#p=3
library(bnlearn)
library(ICSNP)
library(matlib)
library(matrixcalc)
d <- 3 #dimension
n <- 50 #n number of samples
tel1 <- 0
tel2 <- 0
tel3 <-0
mu <-c(0,0,0) #mu
sigma <-sigma <- matrix(diag(c(1/6,2/6,3/6)), ncol =d) #sigma (scatter matrix)
m <- 0
```

```
while (m < 10001) {
  vector <- eigen(sigma)$vectors #real eigenvectors

  #taking samples from the multivariate normal distribution
  x <- rmvnorm(n, mean=mu, sigma=sigma)
  #if we want to do it for the multivariate t distribution
  #the folowing line could be uncommented
  #taking samples from the multivariate t distribution
  #x <- rmvt(n=n, sigma = sigma, df = 5, delta = mu)

  # calculation of eigenvalues (now of spatial sign covariance matrix)
  eigenm <- eigen(cov(spatial.sign(x, center = TRUE, shape = FALSE) ))

  #also need eigenvalues of covariance matrix
  mean <- colMeans(x) #colum means of x
  lambda <- eigen(cov(x-rep(mean, each = nrow(x))))$values

  # calculation of the asymptotic covariances
  Cov1 <- Cov2 <- Cov12 <- Cov3 <- matrix(0,ncol=d,nrow=d)

  #calculating delta and eta
  #eta
  nij <- function(x,i,j) {z <- 1; for (k in 1:d) {
    z <- z*(1+lambda[k]*x)^0.5

  } ;(lambda[i]*lambda[j]*0.25*x)/((1+lambda[i]*x)*(1+lambda[j]*x)*z)}
  eta1 <- eta2 <- eta3 <- c(0,0,0)
  for (k in 2:d) {
    eta1[k] <- integrate(nij,0,Inf, i =1, j=k)$value
  }
  for (k in c(1,3)) {
    eta2[k] <- integrate(nij,0,Inf, i =2, j=k)$value
  }
  for (k in c(1,2)) {
    eta3[k] <- integrate(nij,0,Inf, i =3, j=k)$value
  }
  #delta
  delta <- c(0,0,0)
  deltai <- function(x,i) {
    z <- 1; for (k in 1:d) {
      z <- z*(1+lambda[k]*x)^0.5
    } ; (lambda[i]*0.5)/((1+lambda[i]*x)*z)
  }
  for (k in 1:d) {
    delta[k] <- integrate(deltai,0,Inf, i =k)$value
  }

  # covariance of loadings of first principal component
  for (i in 2:d) Cov1 <- Cov1+eta1[i]/
    (delta[i]-delta[1])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

  # covariance of loadings of second principal component
  for (i in c(1,3)) Cov2 <- Cov2+eta2[i]/
    (delta[i]-delta[2])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

  for (i in c(1,2)) Cov3 <- Cov3+eta3[i]/
```

```
  (delta[i]-delta[3])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

#making the second coordinate of the eigenvectors positive
if (eigenm$vectors[2,1]<0){
  eigenm$vectors[,1]*-1
}
if (eigenm$vectors[2,2]<0){
  eigenm$vectors[,2]*-1
}
if (eigenm$vectors[2,3]<0){
  eigenm$vectors[,3]*-1
}

#calcualting polar coordinates of eigenvectors
eigen_polar <- matrix(0,ncol=2,nrow=d)
for(i in c(1,2,3)){
  eigen_polar[i,1] <- acos(eigenm$vectors[3,i])
  eigen_polar[i,2] <- acos(eigenm$vectors[1,i]/
                           sqrt(eigenm$vectors[1,i]^2 + eigenm$vectors[2,i]^2))
}

#making the second coordinate of the real eigenvectors positive
if (vector[2,1]<0){
  vector[,1]*-1
}
if (vector[2,2]<0){
  vector[,2]*-1
}
if (vector[2,3]<0){
  vector[,3]*-1
}

#calculating the gradient of h_tilde
h1 <- matrix(c(0,0,-1/(sqrt(1-eigenm$vectors[1,3]^2)),
      -eigenm$vectors[1,2]/sqrt(eigenm$vectors[1,1]^2 + eigenm$vectors[1,2]^2),
      eigenm$vectors[1,1]/sqrt(eigenm$vectors[1,1]^2 + eigenm$vectors[1,2]^2),0),
             ncol = 2, nrow = 3)
h2 <- matrix(c(0,0,-1/(sqrt(1-eigenm$vectors[2,3]^2)),
      -eigenm$vectors[2,2]/sqrt(eigenm$vectors[2,1]^2 + eigenm$vectors[2,2]^2),
      eigenm$vectors[2,1]/sqrt(eigenm$vectors[2,1]^2 + eigenm$vectors[2,2]^2),0),
             ncol = 2, nrow = 3)
h3 <- matrix(c(0,0,-1/(sqrt(1-eigenm$vectors[3,3]^2)),
      -eigenm$vectors[3,2]/sqrt(eigenm$vectors[3,1]^2 + eigenm$vectors[3,2]^2),
      eigenm$vectors[3,1]/sqrt(eigenm$vectors[3,1]^2 + eigenm$vectors[3,2]^2),0),
             ncol = 2, nrow = 3)


Cov1 <- Cov1/n
Cov2 <- Cov2/n
Cov3 <- Cov3/n

NewCov1 <- t(h1)%*%Cov1%*%h1
NewCov2 <- t(h2)%*%Cov2%*%h2
NewCov3 <- t(h3)%*%Cov3%*%h3

#calculating the polar coordinates of the real eigenvectors
```

```
  vector_polar <- matrix(0,ncol=2,nrow=d)
  for(i in c(1,2,3)){
    vector_polar[i, 1] <- acos(vector[3,i])
    vector_polar[i, 2] <- acos(vector[1,i]/sqrt(vector[1,i]^2 + vector[2,i]^2))
  }

  #check if the real eigenvector is in the confidence ellipsoid
  if(is.matrix(try(inv(NewCov3))) == TRUE & is.matrix(try(inv(NewCov2))) == TRUE){
  inequality <- ifelse(t(eigen_polar[2,] - vector_polar[2,]) %*%
                         inv(NewCov2)%*%(eigen_polar[2,] - vector_polar[2,])
                       <= qchisq(0.95,2), 1, 0)
  if (inequality[1,1] == 1){
    tel2 <- tel2 + 1
  }
  inequality <- ifelse(t(eigen_polar[3,] - vector_polar[3,]) %*%
                         inv(NewCov3)%*%(eigen_polar[3,] - vector_polar[3,])
                       <= qchisq(0.95,2), 1, 0)
  if (inequality[1,1] == 1){
    tel3 <- tel3 + 1
  }
  m <- m + 1
  }
}
#number of times the real eigenvector is in the confidence ellipsoid
tel2
tel3


tel1
```

R-code for the classical case of tables 4, 5, 6 and 7

```
#simulating how often the real eigenvector is in the 95% confidence interval
#classical case

#for dimension 2
#p=2
set.seed(3)
library(bnlearn)
library(ICSNP)
library(matlib)
d <- 2 #dimension 2
n <- 50 #number of samples
tel1 <- 0
tel2 <- 0
mu <-c(0,0) #mu
sigma <-sigma <- matrix(diag(c(1/3,2/3)), ncol =d) #sigma (scatter matrix)

for (m in 1:100000) {
  vector <- eigen(sigma)$vectors #eigenvectors

  #taking samples from the multivariate normal distribution
  x <- rmvnorm(n, mean=mu, sigma=sigma)
  #taking samples from the multivariate t distribution
  #if we want to do it for the multivariate t distribution the following line
  #could be uncommented
  #x <- rmvt(n=n, sigma = sigma, df = 5, delta = mu)
```

```
#eigenvalues  and vectors of covariance matrix
mean <- colMeans(x) #colum means of x
lambda <- eigen(cov(x-rep(mean, each = nrow(x))))$values
eigenm <- eigen(cov(x-rep(mean, each = nrow(x))))

#estimating kappa
kappa <- 0
for (i in 1:n) {
  kappa <-kappa +  (t(x[i, ]- mean)%*%
                      inv(cov(x-rep(mean, each = nrow(x))))%*%(x[i, ]-mean))^2
}
kappa
kappa <- 1/(8*n)*kappa -1
kappa <- kappa[1,1]
#the following line could be uncommented to take a fixed kappa (which can be changed)
#kappa <- 0

# calculation of the asymptotic covariances
Cov1 <- Cov2 <- Cov12 <- matrix(0,ncol=2,nrow=2)

# covariance of loadings of first principal component
for (i in 2:d) Cov1 <- Cov1+(1+kappa)*lambda[1]*lambda[2]/
  (lambda[1]-lambda[2])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

# covariance of loadings of second principal component
for (i in c(1)) Cov2 <- Cov2+(1+kappa)*lambda[1]*lambda[2]/
  (lambda[1]-lambda[2])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

#making the second coordinate positive
if (eigenm$vectors[2,1]<0){
  eigenm$vectors[,1]*-1
}
if (eigenm$vectors[2,2]<0){
  eigenm$vectors[,2]*-1
}

#calculating the polar coordinates of the eigenvectors
eigen_polar <- c(0,0)
for(i in c(1,2)){
  eigen_polar[i] <- acos(eigenm$vectors[1,i])
}

if (vector[2,1]<0){
  vector[,1]*-1
}
if (vector[2,2]<0){
  vector[,2]*-1
}

#calculating the gradient of h
h1 <- c(-1/(sqrt(1-eigenm$vectors[1,1]^2)), 0)
h2 <- c(-1/(sqrt(1-eigenm$vectors[1,2]^2)), 0)

Cov1 <- Cov1/n
```

```
  Cov2 <- Cov2/n

  NewCov1 <- t(h1)%*%Cov1%*%h1
  NewCov2 <- t(h2)%*%Cov2%*%h2

  #calculating polar coordinates of real eigenvectors
  vector_polar <- c(0,0)
  for(i in c(1,2)){
    vector_polar[i] <- atan2(vector[2,i], vector[1,i])
  }

  #check if the real eigenvector is in the confidence interval
  if(eigen_polar[1]- 1.96* sqrt(NewCov1[1,1]) <= vector_polar[1] & vector_polar[1]
     <= eigen_polar[1]+ 1.96 * sqrt(NewCov1[1,1]) ){
    tel1 <- tel1 + 1
  }
  if(eigen_polar[2]- 1.96* sqrt(NewCov2[1,1]) <= vector_polar[2] & vector_polar[2]
     <= eigen_polar[2]+ 1.96* sqrt( NewCov2[1,1] )){
    tel2 <- tel2 + 1
  }
}
#number of times the real eigenvector is in the confidence interval
tel1
tel2

#for dimension 3
#p=3
set.seed(3)
library(bnlearn)
library(ICSNP)
library(matlib)
d <- 3
n <- 50 #number of samples
tel1 <- 0
tel2 <- 0
tel3 <-0
mu <-c(0,0,0) #mu
sigma <-sigma <- matrix(diag(c(1/6,2/6,3/6)), ncol =d) #sigma (scatter matrix)
m <- 0
while (m < 10001) {
  vector <- eigen(sigma)$vectors #eigenvectors of sigma

  #taking samples from the multivariate normal distribution
  x <- rmvnorm(n, mean=mu, sigma=sigma)
  #taking samples from the multivariate t distribution
  #if we want to do it for the multivariate t distribution the following line
  #could be uncommented
  #x <- rmvt(n=n, sigma = sigma, df = 5, delta = mu)

  #eigenvalues and eigenvectors of covariance matrix
  mean <- colMeans(x) #colum means of x
  lambda <- eigen(cov(x-rep(mean, each = nrow(x))))$values
  eigenm <- eigen(cov(x-rep(mean, each = nrow(x))))

  kappa <- 0
  for (i in 1:n) {
```

```
  kappa <-kappa +  (t(x[i, ]- mean)%*%inv(cov(x-rep(mean, each = nrow(x))))
                    %*%(x[i, ]-mean))^2
}
kappa <- 1/(15*n)*kappa -1
kappa <- kappa[1,1]
#the following line could be uncommented to take a fixed kappa
#kappa <- 2

# calculation of the asymptotic covariances
Cov1 <- Cov2 <- Cov12 <- Cov3 <- matrix(0,ncol=d,nrow=d)

# covariance of loadings of first principal component
for (i in 2:d) Cov1 <- Cov1+(1+kappa)*lambda[1]*lambda[i]/
  (lambda[i]-lambda[1])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

# covariance of loadings of second principal component
for (i in c(1,3)) Cov2 <- Cov2+(1+kappa)*lambda[i]*lambda[2]/
  (lambda[i]-lambda[2])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])

for (i in c(1,2)) Cov3 <- Cov3+(1+kappa)*lambda[i]*lambda[3]/
  (lambda[i]-lambda[3])^2*eigenm$vector[,i]%*%t(eigenm$vector[,i])
# covariance between loadings of first and second rincipal component
#Cov12 <- -eta1[2]/(delta[1]-delta[2])^2*eigenm$vector[,1]%*%t(eigenm$vector[,2])

#making the second coordinate of the eigenvectors positive
if (eigenm$vectors[2,1]<0){
  eigenm$vectors[,1]*-1
}
if (eigenm$vectors[2,2]<0){
  eigenm$vectors[,2]*-1
}
if (eigenm$vectors[2,3]<0){
  eigenm$vectors[,3]*-1
}

#calculating polar coordinates of the eigenvectors
eigen_polar <- matrix(0,ncol=2,nrow=d)
for(i in c(1,2,3)){
  eigen_polar[i,1] <- acos(eigenm$vectors[3,i])
  eigen_polar[i,2] <- acos(eigenm$vectors[1,i]/
                           sqrt(eigenm$vectors[1,i]^2 + eigenm$vectors[2,i]^2))
}

#making the second coordinate of the real eigenvectors positive
if (vector[2,1]<0){
  vector[,1]*-1
}
if (vector[2,2]<0){
  vector[,2]*-1
}
if (vector[2,3]<0){
  vector[,3]*-1
}

#calculating the gradient of h_tilde
h1 <- matrix(c(0,0,-1/(sqrt(1-eigenm$vectors[1,3]^2)),
```

```
                  -eigenm$vectors[1,2]/sqrt(eigenm$vectors[1,1]^2
                                            + eigenm$vectors[1,2]^2),
                   eigenm$vectors[1,1]/sqrt(eigenm$vectors[1,1]^2
                                           + eigenm$vectors[1,2]^2),0),
                 ncol = 2, nrow = 3)
  h2 <- matrix(c(0,0,-1/(sqrt(1-eigenm$vectors[2,3]^2)),
                  -eigenm$vectors[2,2]/sqrt(eigenm$vectors[2,1]^2
                                            + eigenm$vectors[2,2]^2),
                   eigenm$vectors[2,1]/sqrt(eigenm$vectors[2,1]^2
                                           + eigenm$vectors[2,2]^2),0),
                 ncol = 2, nrow = 3)
  h3 <- matrix(c(0,0,-1/(sqrt(1-eigenm$vectors[3,3]^2)),
                  -eigenm$vectors[3,2]/sqrt(eigenm$vectors[3,1]^2
                                            + eigenm$vectors[3,2]^2),
                   eigenm$vectors[3,1]/sqrt(eigenm$vectors[3,1]^2
                                           + eigenm$vectors[3,2]^2),0),
                 ncol = 2, nrow = 3)


  Cov1 <- Cov1/n
  Cov2 <- Cov2/n
  Cov3 <- Cov3/n

  NewCov1 <- t(h1)%*%Cov1%*%h1
  NewCov2 <- t(h2)%*%Cov2%*%h2
  NewCov3 <- t(h3)%*%Cov3%*%h3

  #calculating polar coordinates of the real eigenvectors
  vector_polar <- matrix(0,ncol=2,nrow=d)
  for(i in c(1,2,3)){
    vector_polar[i, 1] <- acos(vector[3,i])
    vector_polar[i, 2] <- acos(vector[1,i]/sqrt(vector[1,i]^2 + vector[2,i]^2))
  }

  #checking if the real eigenvector is in the confidence ellipse
  if(is.matrix(try(inv(NewCov3))) == TRUE & is.matrix(try(inv(NewCov2))) == TRUE){
  inequality <- ifelse(t(eigen_polar[2,] - vector_polar[2,])  %*%
                         inv(NewCov2)%*%(eigen_polar[2,] - vector_polar[2,])
                       <= qchisq(0.95,2), 1, 0)
  if (inequality[1,1] == 1){
    tel2 <- tel2 + 1
  }
  inequality <- ifelse(t(eigen_polar[3,] - vector_polar[3,])  %*%
                         inv(NewCov3)%*%(eigen_polar[3,] - vector_polar[3,])
                       <= qchisq(0.95,2), 1, 0)
  if (inequality[1,1] == 1){
    tel3 <- tel3 + 1
  }
  m <- m+1
  }
}
#number of times the real eigenvector is in the confidence ellipse
tel2
tel3
```