Schedae Informaticae Vol. 28 (2019): 25–47
doi: 10.4467/20838476SI.19.002.14379

ICA based on Split Generalized Gaussian

Przemysław Spurek[1], Przemysław Rola[2], Jacek Tabor[1],
Aleksander Czechowski[3], Andrzej Bedychaj[1]
[1]Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Cracow, Poland, e-mail: *przemyslaw.spurek@uj.edu.pl, jacek.tabor@uj.edu.pl, andrzej.bedychaj@gmail.pl*
[2]Department of Mathematics of the Cracow University of Economics, Rakowicka 27, 31-510 Cracow, Poland, e-mail: *przemyslaw.rola@outlook.com*
[3] Delft University of Technology, Mekelweg 5, 2628 CD Delft, The Netherlands, e-mail: *a.t.czechowski@tudelft.nl*

Abstract. Independent Component Analysis (ICA) is a method for searching the linear transformation that minimizes the statistical dependence between its components. Most popular ICA methods use kurtosis as a metric of independence (non-Gaussianity) to maximize, such as FastICA and JADE. However, their assumption of fourth-order moment (kurtosis) may not always be satisfied in practice. One of the possible solution is to use third-order moment (skewness) instead of kurtosis, which was applied in ICA_{SG} and EcoICA. In this paper we present a competitive approach to ICA based on the Split Generalized Gaussian distribution (SGGD), which is well adapted to heavy-tailed as well as asymmetric data. Consequently, we obtain a method which works better than the classical approaches, in both cases: heavy tails and non-symmetric data.

1. Introduction

Independent component analysis (ICA) has become a standard data analysis technique applied to an array of problems in signal processing and machine learning. The ICA techniques have an application in the magnetic resonance [1], MRI [2, 3], EEG analysis [4–6], fault detection [7], financial time series separation [8], seismic recordings [9] and – the most importantly – image analysis [10–19].

(a) Original images 42049 and 220075. (b) Sum and subtraction of images.

(c) ICA$_{SGG}$. (d) ICA_{SG}. (e) NGPP.

(f) ICA$_{SGG}$. (g) ICA_{SG}. (h) NGPP.

Figure 1.: Comparison of image separation by our method (ICA$_{SGG}$), with ICA_{SG} and NGPP.

Classical ICA methods were introduced in the 1980s, since then a multitude of algorithms have been proposed for solving this problem. Most of the algorithms base either on decomposition of various matrices, maximum likelihood optimization or projection pursuit.

The matrix decomposition class of algorithms is represented by classic method like JADE [20].

Second type of the ICA methods tackle the problem by the maximum likelihood estimation. In such a case we search for the coordinate system optimally fitted to data as well as the marginal densities such that the data density factors in the base are the product of marginal densities. In [21], authors model skewness using the Split Gaussian distribution, which is well adapted to asymmetric data.

The most well-known example belonging to the last type – projection pursuit – is FastICA [22, 23], method that extracts the independent components either sequentially or simultaneously by maximizing kurtosis, an established measure of the non-Gaussianity.

An assumption of the kurtotic sources may not always be satisfied in practice. Typically data sets are bounded, and therefore the credible estimation of tails is not that easy. Another problem with these methods, is that they usually assume that the underlying density is symmetric, which is rarely the case. For weak-kurtosic but skewed sources, such methods could fail [24, 25].

Skewness (the third central moment) is another metric using in ICA. Any symmetric data, in particular the Gaussian one, has skewness equal to zero. One of the most popular ICA methods dedicated for skew data is PearsonICA [26, 27].

Unfortunately, all the above approaches work well only for asymmetric and weak-kurtosic source. Our goal is to find a method which is able to work in both situations. One of the possible solution is to use a mixture of skewness and kurtosis. In [28, 29] authors use the projection index which is a combination of third and fourth cumulants. The proposed method gives good results but it is a problem with modeling the proportion between skewness and kurtosis.

In our work we introduce a new approach to ICA in which we approximate the data density by product of Split Generalized Gaussian (SGG) distribution, which allows us to simultaneously model skewness and heavy-tails in data. Thanks to Theorem 3..1 we reduce the minimization of the maximum likelihood function from five to three parameters. Moreover, in Theorem 3..3 we give an explicit formula for gradient of the cost function, which allows the use of classical gradient descent method. Consequently we obtain ICA method which gives essentially better results then classical approaches with similar computational complexity.

We verified ICA_{SGG} in the case of density estimation of images and found the optimal parameters of Logistic, Split Gaussian, Split Generalized Gaussian distributions. In Fig. 2 we compared the values of the MLE function. In most of the cases Split Generalized Gaussian distribution fits the data with better precision.

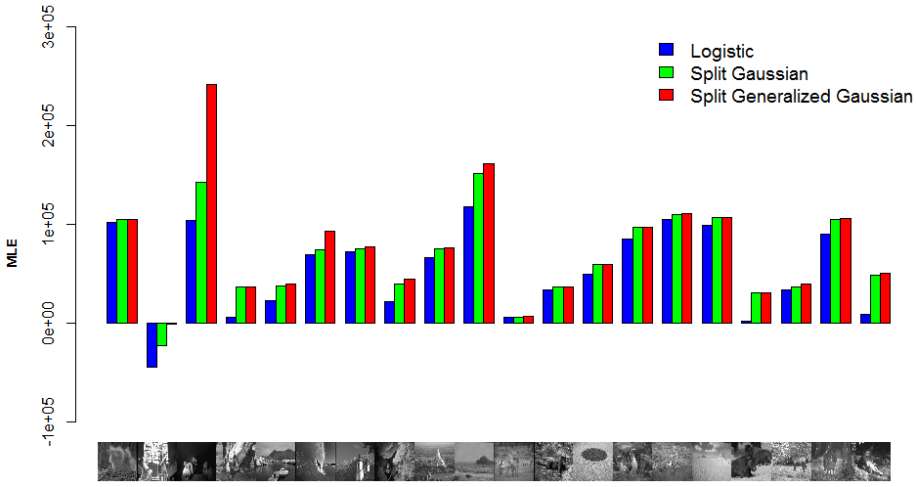


Figure 2.: The MLE estimation for image histograms with respect to Logistic, Split Gaussian and Split Generalized Gaussian distributions.

The results of ICA_{SG} [25] (described in our previous article), NGPP [29] (which

use a combination of third and fourth cumulants) and our method ICA_{SGG} are compared in Fig. 1 for the case of image separation (for more detail comparison we refer to Section 4.). In the experiment we mixed two images (see Fig. 1(a)) by adding and subtracting them (see Fig. 1(b)). Our approach gives essentially better results. In the case of other ICA methods we can see artifacts in the background, which means that the method does not separate signal properly.

This paper is arranged as follows. In the next section, the theoretical background of our approach to ICA is presented. We introduce a cost function which uses the SGG distribution and show that it is enough to minimize it respectively to only three parameters: vector m $\in \mathbb{R}^d$, $d \times d$ matrix W and scalar $c \in \mathbb{R}$. We also calculate the gradient of the cost function, which is necessary for the efficient use in the minimization procedure. In the third section we describe methodology behind maximum likelihood estimation applied to SGG. We will show that the problem can be simplified to the form of the likelihood function. The last section describes the numerical experiments. The effects of our algorithm are illustrated on the simulated as well as the real datasets.

2. Theoretical background

Let us first state formally the ICA problem. Given a random variable X, one want to find an unmixing matrix, i.e. an invertible matrix W such that $W^T X$ has independent components. It can be done under the assumption that based component of unmixed signals are non-Gaussian and are statistically independent from each other.

One of the possible methods for estimating the ICA solutions is maximum likelihood (ML) approach. The method is based on using the well-known fact that the density F of the mixture vector $X = AS$ can be described as

$$F(x) = \det(W) \cdot f_1(w_1^T x) \cdot \ldots \cdot f_d(w_d^T x) \text{ for } x \in \mathbb{R}^d \tag{2..1}$$

where $W = A^{-1}$ (w_i denotes the i-th column of W) and f_i denote the densities of the independent components. Hence, if we want to find such a basis that components become independent, we need to search for a matrix W and one-dimensional densities, from a fixed family $f_i \in \mathcal{F}$, such that the above approximation is optimal form the maximum likelihood point of view.

In maximum likelihood approach, we have to choose a density family $\mathcal{F}$. It may seem that the most natural choice is Gaussian densities. However, this is not the case as Gaussian densities are affine invariant, and therefore do not "prefer" any fixed choice of coordinates[1]. In other words we have to choose a family of densities which is distant from Gaussian ones.

[1] In fact one can observe that the choice of gaussian densities leads to PCA, if we restrict to the case of orthonormal bases

2.1. ICA based on Split Generalized Gaussian distribution

In this section we present our density model. Natural directions for extending the normal distribution are to introduce skewness or heavy-tails, and several proposals have indeed emerged, both in the univariate and multivariate case, see [30–36]. One of the most popular approaches for skewness is the Split Normal (SN) distribution [33] and for heavy tails is the General Gaussian (GG) distribution [34–36].

In our paper we use a generalization of the above models, which we call the Split Generalized Gaussian (SGG) distribution. We start from the one-dimensional case. After that, we present a possible generalization of this definition to the multidimensional setting, which corresponds to the formula (2..1). Contrary to the Split Gaussian distribution, we skip the assumption of the orthogonality of coordinates (often called principal components), and obtain an ICA model.

2.2. The one-dimensional case

Main limitations of the normal distribution are its symmetry and the fixed shape of its tails. As it was mentioned, most ICA methods are based on the maximization of non-Gaussianity. One of the most common and simplest parameters able to describe deviation from normality is skewness defined as the third-order central moment of a stochastic variable. It was found [37] that the information it can provide is equivalent to that yielded by the combination of two empirical parameters, the "left and right variances".

In order to modify the Gaussian pdf to describe deviation from symmetry, the left and right variances were proved to be easier to use than skewness [30–32]. Replacing the variance with two different left and right variances in Gaussian pdf, gave the asymmetric Split Gaussian model:

$$SN(x; m, \sigma^2, \tau^2) = \begin{cases} c \cdot \exp[-\frac{1}{2\sigma^2}(x-m)^2], & \text{where } x \leq m \\ c \cdot \exp[-\frac{1}{2\tau^2\sigma^2}(x-m)^2], & \text{where } x > m \end{cases}$$

where $c = \sqrt{\frac{2}{\pi}}\sigma^{-1}(1+\tau)^{-1}$.

As we see, the split normal distribution arises from merging two opposite halves of two probability density functions of normal distributions in their common mode. In general, the use of the Split Gaussian distribution (even in 1D) allows to fit data with better precision (from the likelihood function point of view).

Another measure of non-Gaussianity in terms of shape is represented by the kurtosis. The parameter is equal to three in the Gaussian case; the sharpness of the pdf shape is higher (lower) than the corresponding Gaussian function when the parameter is larger (smaller) than three. A good model for generalized symmetric pdfs has a variable sharpness. One of the most widely used symmetric pdf models with a variable sharpness is the Generalized Gaussian [34, 38]

$$GG(x; m, \alpha, c) = \frac{c}{2\alpha\Gamma(1/c)} \exp\left[-\frac{|x-m|^c}{\alpha^c}\right],$$

for $m \in \mathbb{R}$ and $\alpha, c \in \mathbb{R}_+$ where Γ is the standard Gamma function. The parameter c, which is theoretical ($c > 0$), influences the model sharpness, but cannot be estimated directly from data samples.

The main limitation affecting the generalized Gaussian model is the symmetry. As the left and right variances were replaced by the variance in the Gaussian pdf in order to build the asymmetric Gaussian model, these two parameters are introduced into the kurtosis-based generalized Gaussian pdf in a similar way, by transforming it into the following asymmetric – Split Generalized Gaussian model:

$$SGG(x; m, \sigma_l, \sigma_r, c) = \begin{cases} \frac{c}{(\alpha_l+\alpha_r)\Gamma(1/c)} \exp[-\frac{|x-m|^c}{\alpha_l^c}] & \text{where } x < m \\ \frac{c}{(\alpha_l+\alpha_r)\Gamma(1/c)} \exp[-\frac{|x-m|^c}{\alpha_r^c}] & \text{where } x \geq m \end{cases}$$

for $m \in \mathbb{R}$ and $\sigma_l, \sigma_r, c \in \mathbb{R}_+$. The relation between α_l, α_r and standard deviations σ_l, σ_r is

$$\alpha_i = \sigma_i \sqrt{\frac{\Gamma(1/c)}{\Gamma(3/c)}}, \text{ for } i = l, r.$$

2.3. Multidimensional Split Gaussian distribution

A natural generalization of the univariate Generalized Gaussian distribution to the multivariate settings was presented in [36]. Roughly speaking, authors assume that a vector $\mathrm{x} \in \mathbb{R}^d$ follows the multivariate Generalized Gaussian distribution, if its principal components are orthogonal and follow the one-dimensional Generalized Gaussian distribution.

In this article we introduce a possible generalization of the Split Generalized Gaussian distribution, but without the assumption of the orthogonality. The construction of the model is similar to the multivariate Split Gaussian distribution presented in [25] for ICA_{SG} method. Thanks to the use of the Split Generalized Gaussian distribution we can model skewness and kurtosis at the same time.

Definition 2..1. *A density of the multivariate Split Generalised Gaussian distribution is given by*

$$SGG_d(\mathrm{x}; \mathrm{m}, W, \sigma_l, \sigma_r, c) = |\det(W)| \prod_{j=1}^{d} SGG(\omega_j^T(\mathrm{x} - \mathrm{m}); 0, \sigma_{lj}, \sigma_{rj}, c),$$

where ω_j is the j-th column of non-singular matrix W, $\mathrm{m} = (m_1, \ldots, m_d)^T$, $\sigma_l = (\sigma_{l1}, \ldots, \sigma_{ld})$, $\sigma_r = (\sigma_{r1}, \ldots, \sigma_{rd})$ and c is a constant.

Our model is a natural generalization of the multivariate Generalized Gaussian distribution proposed in [32] and the multivariate Split Gaussian distribution described in [25].

The above model is flexible, and allows to fit data with greater precision. In the next section we discuss how to estimate optimal parameters in our model.

3. Methodology

In this section we will use the maximum likelihood estimation applied to the SGG distribution, introduced beforehand. Our goal is to maximize the likelihood function with respect to five parameters. Apart from the classical Gaussian, in case of the Split Generalized Gaussian distribution the MLE method comes down the optimization problem, as stated below.

It is worth to mention, that the problem can be simplified due to the form of the likelihood function. The core of this function can be represented by the component l, and the MLE is equivalent to the minimization of this l . To obtain this, the gradient method is going to be applied.

3.1. Optimization problem

We start with the problem reduction by considering the MLE only with regards to two parameters σ_l and σ_r. This approach allows as to get the explicit formulas for the estimators of former parameters. Consequently, it boils down the problem to the minimization of the quite simple function of three parameters m, W and c.

Theorem 3..1. *Let* $\mathrm{x}_1, \ldots, \mathrm{x}_n$ *be given. Then the likelihood maximized w.r.t.* σ_l *and* σ_r *is*

$$\hat{L}(X; \mathrm{m}, W, c) = \left(\frac{\kappa n}{ce}\right)^{\frac{dn}{c}} \left(|\det(W)|^{-\frac{c}{c+1}} \prod_{j=1}^{d} g_j(\mathrm{m}, W)\right)^{-\frac{n(c+1)}{c}} \tag{3..1}$$

where $\kappa = \left[\frac{1}{c}\Gamma(\frac{1}{c})\right]^{-c}$ *and*

$$\begin{aligned} g_j(\mathrm{m}, W, c) &= s_{1j}^{\frac{1}{c+1}} + s_{2j}^{\frac{1}{c+1}}, \\ s_{1j} &= \sum_{i \in I_j} |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^c, I_j = \{i = 1, \ldots, n \colon \omega_j^T(\mathrm{x}_i - \mathrm{m}) \leq 0\}, \\ s_{2j} &= \sum_{i \in I_j'} |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^c, I_j' = \{i = 1, \ldots, n \colon \omega_j^T(\mathrm{x}_i - \mathrm{m}) > 0\}, \end{aligned}$$

and the maximum likelihood estimators of α_{lj}, α_{rj} *are*

$$\hat{\alpha}_{lj} = \hat{\sigma}_{lj}\sqrt{\frac{\Gamma(\frac{1}{c})}{\Gamma(\frac{3}{c})}} \qquad \text{and} \qquad \hat{\alpha}_{rj} = \hat{\sigma}_{rj}\sqrt{\frac{\Gamma(\frac{1}{c})}{\Gamma(\frac{3}{c})}}$$

where the estimators of σ_l and σ_r are given by

$$\hat{\sigma}_{lj}^c(\mathrm{m}, W) = \tfrac{c}{n}\beta^{\frac{c}{2}} s_{1j}^{\frac{c}{c+1}} g_j(\mathrm{m}, W), \qquad \hat{\tau}_j(\mathrm{m}, W) = \left(\frac{s_{2j}}{s_{1j}}\right)^{\frac{1}{c+1}}$$

and

$$\hat{\sigma}_{rj}^c(\mathrm{m}, W) = \hat{\sigma}_{lj}^c(\mathrm{m}, W) \cdot \hat{\tau}_j^c(\mathrm{m}, W) = \tfrac{c}{n}\beta^{\frac{c}{2}} s_{2j}^{\frac{c}{c+1}} g_j(\mathrm{m}, W),$$

where $\beta = \frac{\Gamma(\frac{3}{c})}{\Gamma(\frac{1}{c})}$.

Proof. See Appendix 7.. □

The MLE is now reduced to the maximization of the function (3..1) of three parameters. To solve this the gradient method was applied.

3.2. Gradient

The gradient method seems to be the one of the most common optimization methods. This is also the method we are going to apply for our problem. To do this we calculate the gradient of the log-likelihood function.

In the first step we introduce the auxiliary function

$$l(X; \mathrm{m}, W, c) = |\det(W)|^{-\frac{c}{c+1}} \prod_{j=1}^{d} g_j(\mathrm{m}, W), \tag{3..2}$$

where ω_j stands for the j-th column of matrix W. Let us notice that

$$\ln \hat{L}(X; \mathrm{m}, W, c) = \frac{dn}{c} \ln\left(\frac{\kappa n}{ce}\right) - \frac{n(c+1)}{c} \ln l(X; \mathrm{m}, W, c) \tag{3..3}$$

We calculate a gradient of l and then we demonstrate the final result.

Theorem 3..2. *Let $X \subset \mathbb{R}^d$, $\mathrm{m} = (\mathrm{m}_1, \ldots, \mathrm{m}_d)^T \in \mathbb{R}^d$, $W = (\omega_{ij})_{1 \le i,j \le d}$ non-singular be given. Then*

$$\nabla_{\mathrm{m}} \ln l(X; \mathrm{m}, W, c) = \left(\frac{\partial \ln l(X; \mathrm{m}, W, c)}{\partial \mathrm{m}_1}, \ldots, \frac{\partial \ln l(X; \mathrm{m}, W, c)}{\partial \mathrm{m}_d}\right)^T,$$

where

$$\frac{\partial \ln l(X; \mathrm{m}, W)}{\partial \mathrm{m}_k} = \tfrac{c}{c+1} \sum_{j=1}^{d} \frac{1}{s_{1j}^{\frac{1}{c+1}} + s_{2j}^{\frac{1}{c+1}}} \cdot$$
$$\cdot \left(s_{1j}^{-\frac{c}{c+1}} \sum_{i \in I_j} |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^{c-1} \omega_{jk} - s_{2j}^{-\frac{c}{c+1}} \sum_{i \in I_j'} |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^{c-1} \omega_{jk} \right).$$

Moreover, $\nabla_W \ln l(X;\mathrm{m},W,c) = \left[\frac{\partial \ln l(X;\mathrm{m},W,c)}{\partial \omega_{pk}}\right]_{1\le p,k\le d}$, *where*

$$\frac{\partial \ln l(X;\mathrm{m},W,c)}{\partial \omega_{pk}} = -\frac{c}{c+1}(\omega^{-1})^T_{pk} + \frac{1}{s_{1p}^{\frac{1}{c+1}} + s_{2p}^{\frac{1}{c+1}}} \cdot$$
$$\cdot\Bigg(-\frac{c}{c+1} s_{1p}^{-\frac{c}{c+1}} \sum_{i\in I_p} |\omega_p^T(\mathrm{x}_i - \mathrm{m})|^{c-1}(\mathrm{x}_{ik} - \mathrm{m}_k)$$
$$+\frac{c}{c+1} s_{2p}^{-\frac{c}{c+1}} \sum_{i\in I_p'} |\omega_p^T(\mathrm{x}_i - \mathrm{m})|^{c-1}(\mathrm{x}_{ik} - \mathrm{m}_k)\Bigg).$$

and

$$s_{1j} = \sum_{i\in I_j} |\omega_j^T(-\mathrm{x}_i + \mathrm{m})|^c, \qquad I_j = \{i = 1,\dots,n\colon \omega_j^T(\mathrm{x}_i - \mathrm{m}) \le 0\},$$
$$s_{2j} = \sum_{i\in I_j'} |\omega_j^T(-\mathrm{x}_i + \mathrm{m})|^c, \qquad I_j' = \{i = 1,\dots,n\colon \omega_j^T(\mathrm{x}_i - \mathrm{m}) > 0\}.$$

Finally

$$\frac{\partial \ln l(X;\mathrm{m},W,c)}{\partial c} = -\frac{1}{(c+1)^2}\ln|\det(W)| +$$
$$\sum_{j=1}^{d} \frac{1}{s_{1j}^{\frac{1}{c+1}} + s_{2j}^{\frac{1}{c+1}}} \left(\frac{1}{c+1} s_{1j}^{-\frac{c}{c+1}} \frac{\partial s_{1j}}{\partial c} - \frac{s_{1j}^{\frac{1}{c+1}}}{(c+1)^2}\ln s_{1j} + \frac{1}{c+1} s_{2j}^{-\frac{c}{c+1}} \frac{\partial s_{2j}}{\partial c} - \frac{s_{2j}^{\frac{1}{c+1}}}{(c+1)^2}\ln s_{2j} \right)$$

where

$$\frac{\partial s_{1j}}{\partial c} = \sum_{i\in I_j} |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^c \ln|\omega_j^T(\mathrm{x}_i - \mathrm{m})|,$$
$$\frac{\partial s_{2j}}{\partial c} = \sum_{i\in I_j'} |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^c \ln|\omega_j^T(\mathrm{x}_i - \mathrm{m})|.$$

Proof. See Appendix 8.. □

Now we are ready to calculate the gradient of the log-likelihood function.

Theorem 3..3. *Let* $X \subset \mathbb{R}^d$, $c \in \mathbb{R}$, $\mathrm{m} = (\mathrm{m}_1,\dots,\mathrm{m}_d)^T \in \mathbb{R}^d$, $W = (\omega_{ij})_{1\le i,j\le d}$ *non-singular be given. Then*

$$\nabla_{\mathrm{m}} \ln \hat{L}(X;\mathrm{m},W,c) = \left(\frac{\partial \ln \hat{L}(X;\mathrm{m},W,c)}{\partial \mathrm{m}_1}, \dots, \frac{\partial \ln \hat{L}(X;\mathrm{m},W,c)}{\partial \mathrm{m}_d} \right)^T, \tag{3..4}$$

where

$$\frac{\partial \ln \hat{L}(X;\mathrm{m},W,c)}{\partial \mathrm{m}_k} = -\frac{n(c+1)}{c} \frac{\partial \ln l(X;\mathrm{m},W,c)}{\partial \mathrm{m}_k}. \tag{3..5}$$

Moreover, $\nabla_W \ln \hat{L}(X;\mathrm{m},W,c) = \left[\frac{\partial \ln \hat{L}(X;\mathrm{m},W,c)}{\partial \omega_{pk}}\right]_{1\le p,k\le d}$, *where*

$$\frac{\partial \ln \hat{L}(X;\mathrm{m},W,c)}{\partial \omega_{pk}} = -\frac{n(c+1)}{c} \frac{\partial \ln l(X;\mathrm{m},W,c)}{\partial \omega_{pk}} \tag{3..6}$$

Finally $\frac{\partial \ln \hat{L}(X;\mathrm{m},W,c)}{\partial c} =$

$$\frac{dn}{c^2}\left[\ln\left(\frac{ce}{n}\right) - 1 + c + \psi\left(\frac{1}{c}\right) \right] + \frac{n}{c^2}\ln l(X;\mathrm{m},W,c) - \frac{n(c+1)}{c} \frac{\partial \ln l(X;\mathrm{m},W,c)}{\partial c} \tag{3..7}$$

and ψ is the so-called digamma function, i.e. $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$.

Proof. 3..3 Recall that

$$\ln \hat{L} = \frac{dn}{c} \ln \left(\frac{\kappa n}{ce} \right) - \frac{n(c+1)}{c} \ln l$$

Then

$$\frac{\partial \ln \hat{L}}{\partial c} = \frac{\partial A}{\partial c} + \frac{n}{c^2} \ln l - \frac{n(c+1)}{c} \frac{\partial \ln l}{\partial c}$$

where $A = \frac{dn}{c} \ln \left(\frac{\kappa n}{ce} \right)$. Let us calculate $\frac{\partial A}{\partial c}$. Notice that

$$A = \frac{dn}{c} \ln \left[\frac{n}{ce} \left(\frac{1}{c} \Gamma(\frac{1}{c}) \right)^{-c} \right] = \frac{dn}{c} \ln \left(\frac{n}{ce} \right) - dn \ln \left(\frac{1}{c} \Gamma(\frac{1}{c}) \right) =$$
$$\frac{dn}{c} \ln \left(\frac{n}{e} \right) - \frac{dn}{c} \ln c - dn \ln \Gamma(\frac{1}{c}) + dn \ln c.$$

Then

$$\frac{\partial A}{\partial c} = -\frac{dn}{c^2} \ln \left(\frac{n}{e} \right) + \frac{dn}{c^2} \ln c - \frac{dn}{c^2} - \frac{dn}{\Gamma(\frac{1}{c})} \Gamma'(\frac{1}{c}) \cdot (-\frac{1}{c^2}) + \frac{dn}{c} =$$
$$\frac{dn}{c^2} \left[\ln c - 1 + c - \ln(\frac{n}{e}) + \frac{\Gamma'(\frac{1}{c})}{\Gamma(\frac{1}{c})} \right].$$

□

Thanks to the above Theorem we can use gradient descent, a first-order optimization algorithm. To find a local maximum of the cost function using gradient descent, one takes steps proportional to the gradient of the function at the current point.

4. Experiments and analysis

To evaluate our method we will compare it with the classical one. For this purpose we use Tucker's congruence coefficient [39] (uncentered correlation). Its values range between -1 and $+1$. This index can assess the similarity of extracted factors across different samples. Generally, a congruence coefficient of 0.9 indicates a high degree of factor similarity, while a coefficient of 0.95 or higher indicates that the factors are virtually identical.

We can also verify the quality of recomputing mixing matrix. The Amari-Cichocki-Yang (ACY) [40] error is an asymmetric measure of dissimilarity between two non-singular square matrices. The ACY error is invariant to permutation and rescaling of the columns (equals 0 if and only if matrices are identical up to column permutations and rescaling).

We evaluate our method in the context of images, sound and EEG data. For this purpose we use R packages `ica` [41], `PearsonICA` [42], `ProDenICA` [43], `tsBSS` [44], `fICA` [45], `ICtest` [46]. The most common method used in practice is FastICA [22,23] where negentropy is applied. To estimate it here we can use three different functions, i.e. logcosh, exp and kurtosis. Apart from this we compare our method with algorithm using Information-Maximization (Infomax) approach [47]. Similarly to FastICA we

(a) Original image. (b) Original image. (c) Sum of images. (d) Subtraction of images.

(e) ICA_{SGG}. (f) ICA_{SGG}. (g) ICA_{SG}. (h) ICA_{SG}.

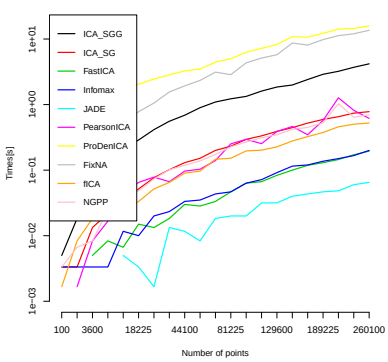
(i) FastICA with (logcosh). (j) FastICA with (logcosh). (k) NGPP. (l) NGPP.

(m) JADE. (n) JADE. (o) fICA. (p) fICA.

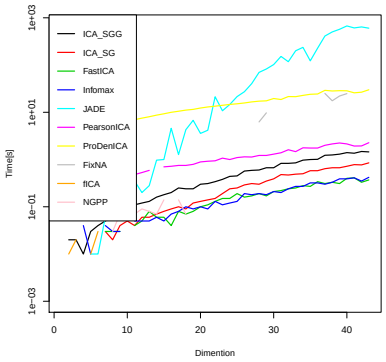
Figure 3.: Results of image separation with the use of various ICA algorithms.

take into account three possible non-linear functions: hyperbolic tangent, logistic and extended Infomax. Finally, we consider algorithm which applies Joint Approximate Diagonalization of Eigenmatrices (JADE) proposed by Cardoso and Souloumiac's [20, 23].

PearsonICA [26, 27] appears to be the one of the most popular ICA methods dedicated for skew data. It minimizes mutual information using a Pearson [48] system-based parametric model. Another model we consider is ProDenICA [49, 50], which is based not on a single nonlinear function, but on an entire function space of candidate nonlinearities. In particular, the method works with the functions in a reproducing



(a) Dependence of the number of data set instances.



(b) Dependence of the dimension of data.

Figure 4.: Comparison of computational efficiency between ICA_{SGG} and classical ICA methods (Time axis is given in the logarithmic scale).

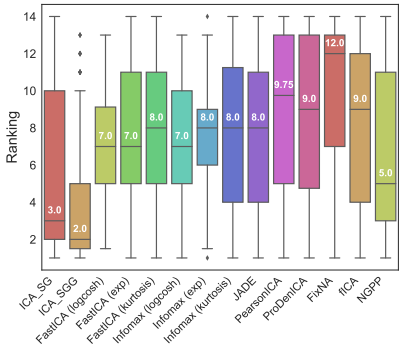
kernel Hilbert space, and make use of the "kernel trick" to search over this space efficiently. Finally, we compare our method with NGPP [29]. It uses the projection index, i.e. loosely speaking a combination of third and fourth cumulants.

4.1. Computational efficiency

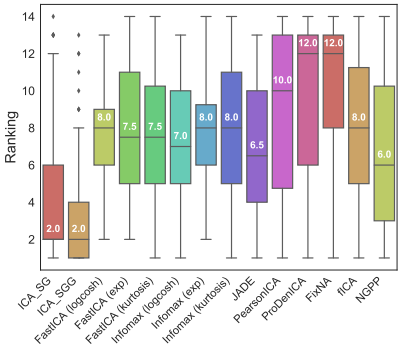
First, we verify the computational times of ICA_{SGG} and alternative ICA algorithms. We examine the influence on the number of data set instances and dimension of data.

We consider the classical image separation problem, where two images are mixed together. We use ten mixed examples and present mean evaluation times. To vary the size of data, images are scaled to different sizes, and running times of the algorithms are reported in each case. One can observe in Figure 4(a) that ICA_{SGG} is a little bit slower than NGPP but gives comparable results.

To examine the influence of data dimension on the evaluation time we also take into account the classical image separation problem, but we change the number of components from 2 to 40. ICA_{SGG} has similar complexity as state of the art method, see Figure 4(b). FastICA, Infomax and JADE are the most effective, but do not solve the problem of image separation sufficiently well, see Fig. 5. On the other hand, the ProDenICA and NGPP which give comparable results to ICA_{SGG}, have comparable computational times.



(a) Boxplots of ranks of ICA methods obtained by using the Tucker's congruence coefficient measure in the separation of images.



(b) Boxplots of ranks of ICA methods obtained by using Amari-Cichocki-Yang measures in the separation of images.

Figure 5.: Boxplots of ranks of ICA methods obtained by using the Tucker's congruence coefficient and Amari-Cichocki-Yang measures in the separation of images.

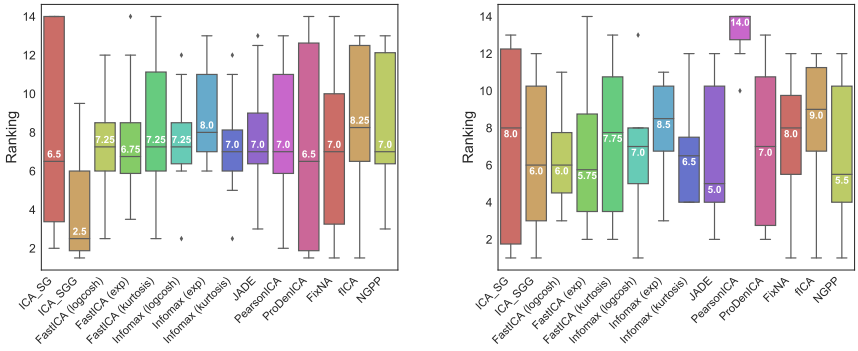
4.2. Separation of images

One of the most popular application of ICA is the separation of images. In our experiments we use three hundred images from: the USC-SIPI Image Database (of size 256×256 pixels and 512×512 pixels), and from Berkeley Segmentation Dataset of size 482×321. We make random pairs of above images and use them as a source signal, combined by the mixing matrix. From the practical perspective, we simply obtain two new images by adding and dividing source pictures. Our goal is to reconstruct original images by using only the knowledge about the mixed ones. As a summary from the experiment, in Fig. 5 we present a boxplots of ranks obtained by the methods.

In the case of the Tucker's congruence coefficient measure and Amari-Cichocki-Yang error in most of the situations we observe better results. The ICA_{SGG} method essentially better recovers original signals and as we can see in Fig. 3, ICA_{SGG} almost perfectly recovers source signal.

4.3. Cocktail-party problem

In this subsection we consider cocktail-party problem to compare our method with the classical ones. Imagine that you are in a room where two people are speaking simultaneously. You have two microphones, which you hold in different locations. The microphones give you two recorded time signals, which we could interpret as mixed signal x. Each of these recorded signals is a weighted sum of the speech signals emitted by the two speakers, which we denote by s. The cocktail-party problem is to



(a) Boxplots of ranks of ICA methods obtained by using the Tucker's congruence coefficient measure in the case of Cocktail-party problem.

(b) Boxplots of ranks of ICA methods obtained by using Amari-Cichocki-Yang measures in the case of Cocktail-party problem.

Figure 6.: Boxplots of ranks of ICA methods obtained by using the Tucker's congruence coefficient and Amari-Cichocki-Yang measures in the case of Cocktail-party problem.

estimate the two original speech signals.

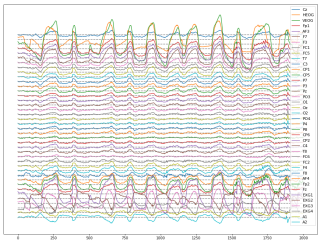
In our experiments we use signal obtained by mixing synthetic sources[2]. As a summary from the experiment, in Fig. 6 we present a boxplots of ranks obtained by the methods. In the case of cocktail-party problem our method recovers sources signal better or comparable to the classical methods.

4.4. EEG

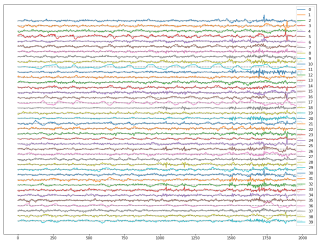
At the end of this section we present how our method works in the case of EEG signals. In this context, ICA is applied to many different problems like eye movements, blinks, muscle, heart and line noise e.t.c. In this experiment we focus only on eye movements and blink artifacts. The goal here is to demonstrate that our method is capable of finding artifacts in the real EEG data. However, we emphasize that it does not provide a complete solution to any of these practical problems. Such a solution usually entails a significant amount of domain-specific knowledge and engineering. Nevertheless, from these preliminary results with EEG data, we believe that the method presented in this paper provides a reasonable solution for signal separation, which is simple and effective enough to be easily customized for a broad range of practical problems.

For the EEG analysis, the rows of the input matrix x are the EEG signals recorded at different electrodes, the rows of the output data matrix $s = W\mathrm{x}$ are time courses of activation of the ICA components, and the columns of the inverse matrix W give

[2] We use signals from `http://research.ics.aalto.fi/ica/cocktail/cocktail_en.cgi`.

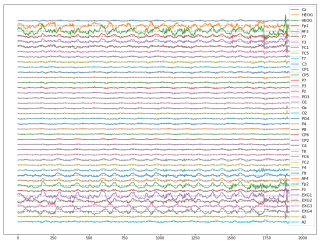


(a) Original signal from EEG.



(b) Sources signals obtained by ICA_{SGG}.

(c) Three components 9, 20, 36.



(d) Original EEG signal with removed three components 9, 20, 36.

Figure 7.: Results of ICA_{SGG}in the case of EEG data.

the projection strengths of the respective components onto scalp sensors.

One EEG data set used in the analysis was collected from 40 scalp electrodes (see Fig. 7(a)). The second and the third ones are located very near to the eye and can be understood as a base (we can use them for removing eye blinking artifacts). In Fig. 7(b) we present signals obtained by ICA_{SGG}. The scale of this figure is large but we can find the data which have spikes exactly in the same place as the two base signals (see Fig. 7(c)). After removing selected signal and going back to the original situation we obtain signal (see Fig. 7(d)) without eye blinking artifacts (compare Fig. 7(a) with Fig. 7(d)).

5. Conclusion

In this paper we introduce a new approach to ICA in which we approximate the data density by product of Split Generalized Gaussian distribution, which allows us to model at the same time skewness and heavy-tails in data. Consequently, we

obtain ICA method which gives essentially better results than classical approaches with slightly worse computational complexity.

We verify our approach on images, sound and EEG data. In the case of source signal reconstructing our approach better recover original signals. The main reason for that is the real data being usually skewed with heavy tails.

Acknowledgment

Research of P. Spurek was supported by the National Center of Science (Poland) grant no. 2015/19/D/ST6/01472. Research of J. Tabor was supported by the National Center of Science (Poland) grant no. UMO-2014/13/B/ST6/01792.

6. References

[1] Beckmann, C.F., Smith, S.M., *Probabilistic independent component analysis for functional magnetic resonance imaging.* Medical Imaging, IEEE Transactions on, 2004, **23**(2), pp. 137–152.

[2] Beckmann, C.F., Smith, S.M., *Tensorial extensions of independent component analysis for multisubject fmri analysis.* Neuroimage, 2005, **25**(1), pp. 294–311.

[3] Rodriguez, P.A., Calhoun, V.D., Adalı, T., *De-noising, phase ambiguity correction and visualization techniques for complex-valued ica of group fmri data.* Pattern recognition, 2012, **45**(6), pp. 2050–2063.

[4] Brunner, C., Naeem, M., Leeb, R., Graimann, B., Pfurtscheller, G., *Spatial filtering and selection of optimized components in four class motor imagery eeg data using independent components analysis.* Pattern Recognition Letters, 2007, **28**(8), pp. 957–964.

[5] Delorme, A., Sejnowski, T., Makeig, S., *Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis.* Neuroimage, 2007, **34**(4), pp. 1443–1449.

[6] Zhang, H., Yang, H., Guan, C., *Bayesian learning for spatial filtering in an eeg-based brain–computer interface.* IEEE transactions on neural networks and learning systems, 2013, **24**(7), pp. 1049–1060.

[7] Choi, S.W., Martin, E.B., Morris, A.J., Lee, I.B., *Fault detection based on a maximum-likelihood principal component analysis (pca) mixture.* Industrial & engineering chemistry research, 2005, **44**(7), pp. 2316–2327.

[8] Kiviluoto, K., Oja, E., Independent component analysis for parallel financial time series. In: *ICONIP*. vol. 2., 1998, pp. 895–898.

[9] Haghighi, A.M., Haghighi, I.M., et al., An ica approach to purify components of spatial components of seismic recordings. In: *SPE Annual Technical Conference and Exhibition*, Society of Petroleum Engineers, 2008.

[10] Yang, J., Gao, X., Zhang, D., Yang, J.y., *Kernel ica: An alternative formulation and its application to face recognition.* Pattern Recognition, 2005, **38**(10), pp. 1784–1787.

[11] Dagher, I., Nachar, R., *Face recognition using ipca-ica algorithm.* IEEE transactions on pattern analysis and machine intelligence, 2006, **28**(6), pp. 996–1000.

[12] Chuang, C.F., Shih, F.Y., *Recognizing facial action units using independent component analysis and support vector machine.* Pattern recognition, 2006, **39**(9), pp. 1795–1798.

[13] Tsai, D.M., Lin, P.C., Lu, C.J., *An independent component analysis-based filter design for defect detection in low-contrast surface images.* Pattern Recognition, 2006, **39**(9), pp. 1679–1694.

[14] Jenssen, R., Eltoft, T., *Independent component analysis for texture segmentation.* Pattern Recognition, 2003, **36**(10), pp. 2301–2315.

[15] Bressan, M., Guillamet, D., Vitria, J., *Using an ica representation of local color histograms for object recognition.* Pattern Recognition, 2003, **36**(3), pp. 691–701.

[16] Kim, K.I., Franz, M.O., Scholkopf, B., *Iterative kernel principal component analysis for image modeling.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, **27**(9), pp. 1351–1366.

[17] Luo, B., Wilson, R.C., Hancock, E.R., *Spectral embedding of graphs.* Pattern recognition, 2003, **36**(10), pp. 2213–2230.

[18] Luo, B., Wilson, R.C., Hancock, E.R., The independent and principal component of graph spectra. In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on.* vol. 2., IEEE, 2002, pp. 164–167.

[19] Lai, Z., Xu, Y., Chen, Q., Yang, J., Zhang, D., *Multilinear sparse principal component analysis.* IEEE transactions on neural networks and learning systems, 2014, **25**(10), pp. 1942–1950.

[20] Cardoso, J.F., Souloumiac, A., Blind beamforming for non-gaussian signals. In: *Radar and Signal Processing, IEE Proceedings F.* vol. 140., IET, 1993, pp. 362–370.

[21] Pham, D.T., Garat, P., *Blind separation of mixture of independent sources through a quasi-maximum likelihood approach.* Signal Processing, IEEE Transactions on, 1997, **45**(7), pp. 1712–1725.

[22] Hyvärinen, A., *Fast and robust fixed-point algorithms for independent component analysis.* Neural Networks, IEEE Transactions on, 1999, **10**(3), pp. 626–634.

[23] Helwig, N.E., Hong, S., *A critique of tensor probabilistic independent component analysis: implications and recommendations for multi-subject fmri data analysis.* Journal of neuroscience methods, 2013, **213**(2), pp. 263–273.

[24] Song, L., Lu, H., Ecoica: Skewness-based ica via eigenvectors of cumulant operator. In: *Asian Conference on Machine Learning*, 2016, pp. 445–460.

[25] Spurek, P., Tabor, J., Rola, P., Ociepka, M., *Ica based on asymmetry.* Pattern Recognition, 2017, **67**, pp. 230–244.

[26] Karvanen, J., Eriksson, J., Koivunen, V., Pearson system based method for blind separation. In: *Proceedings of Second International Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000), Helsinki, Finland*, 2000, pp. 585–590.

[27] Karvanen, J., Koivunen, V., *Blind separation methods based on pearson system and its extensions.* Signal Processing, 2002, **82**(4), pp. 663–673.

[28] Blaschke, T., Wiskott, L., *Cubica: Independent component analysis by simultaneous third-and fourth-order cumulant diagonalization.* IEEE Transactions on Signal Processing, 2004, **52**(5), pp. 1250–1256.

[29] Virta, J., Nordhausen, K., Oja, H., *Projection pursuit for non-gaussian independent components.* arXiv preprint arXiv:1612.05445, 2016.

[30] Azzalini, A., *A class of distributions which includes the normal ones.* Scandinavian journal of statistics, 1985, pp. 171–178.

[31] Azzalini, A., Dalla Valle, A., *The multivariate skew-normal distribution.* Biometrika, 1996, **83**(4), pp. 715–726.

[32] Villani, M., Larsson, R., *The multivariate split normal distribution and asymmetric principal components analysis.* Communications in Statistics—Theory and Methods, 2006, **35**(6), pp. 1123–1140.

[33] Gibbons, J., Mylroie, S., *Estimation of impurity profiles in ion-implanted amorphous targets using joined half-gaussian distributions.* Applied Physics Letters, 1973, **22**(11), pp. 568–569.

[34] Nandi, A.K., Mämpel, D., *An extension of the generalized gaussian distribution to include asymmetry.* Journal of the Franklin Institute, 1995, **332**(1), pp. 67–75.

[35] Tesei, A., Regazzoni, C.S., *Hos-based generalized noise pdf models for signal detection optimization.* Signal Processing, 1998, **65**(2), pp. 267–281.

[36] Pascal, F., Bombrun, L., Tourneret, J.Y., Berthoumieu, Y., *Parameter estimation for multivariate generalized gaussian distributions.* IEEE Transactions on Signal Processing, 2013, **61**(23), pp. 5960–5971.

[37] Tacconi, G., Tesei, A., Regazzoni, C., A new hos-based model for signal detection in non-gaussian noise: an application to underwater acoustic communications. In: *OCEANS'95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings.* vol. 1., IEEE, 1995, pp. 620–625.

[38] Miller, J., Thomas, J., *Detectors for discrete-time signals in non-gaussian noise.* IEEE Transactions on Information Theory, 1972, **18**(2), pp. 241–250.

[39] Lorenzo-Seva, U., Ten Berge, J.M., *Tucker's congruence coefficient as a meaningful index of factor similarity.* Methodology, 2006, **2**(2), pp. 57–64.

[40] Amari, S.i., Cichocki, A., Yang, H.H., A new learning algorithm for blind signal separation. In: *Advances in neural information processing systems*, 1996, pp. 757–763.

[41] Helwig, N.E., *ica: Independent Component Analysis.* 2015 R package version 1.0-1.

[42] Karvanen, J., *PearsonICA.* 2008 R package version 1.2-3.

[43] Hastie, T., Tibshirani, R., *ProDenICA: Product Density Estimation for ICA using tilted Gaussian density estimates.* 2010 R package version 1.0.

[44] Matilainen, M., Miettinen, J., Nordhausen, K., Oja, H., Taskinen, S., *tsBSS: Tools for Blind Source Separation for Time Series.* 2016 R package version 0.2.

[45] Miettinen, J., Nordhausen, K., Oja, H., Taskinen, S., *fICA: Classical, Reloaded and Adaptive FastICA Algorithms.* 2015 R package version 1.0-3.

[46] Nordhausen, K., Oja, H., Tyler, D.E., Virta, J., *ICtest: Estimating and Testing the Number of Interesting Components in Linear Dimension Reduction.* 2016 R package version 0.2.

[47] Bell, A.J., Sejnowski, T.J., *An information-maximization approach to blind separation and blind deconvolution.* Neural computation, 1995, **7**(6), pp. 1129–1159.

[48] Stuart, A., Kendall, M.G., et al., *The advanced theory of statistics.* Charles Griffin, 1968.

[49] Bach, F.R., Jordan, M.I., *Kernel independent component analysis.* Journal of machine learning research, 2002, **3**(Jul), pp. 1–48.

[50] Hastie, T., Tibshirani, R., Friedman, J., *The elements of statistical learning 2nd edition*, 2009.

7. Appendix A

Proof of Theorem 3..1. Let $X = \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$. We write

$$\mathrm{z}_i = W(\mathrm{x}_i - \mathrm{m}), \quad \mathrm{z}_{ij} = \omega_j^T(\mathrm{x}_i - \mathrm{m}),$$

for observation i, where $i = 1, \ldots, n$ and coordinates $j = 1, \ldots, d$.

Let us consider the likelihood function, i.e.

$$
\begin{aligned}
&L(X;\mathrm{m},W,\sigma_l,\sigma_r,c)=\prod_{i=1}^{n}SGG_d(\mathrm{x}_i;\mathrm{m},W,\sigma_l,\sigma_r,c)\\
&=\prod_{i=1}^{n}|\det(W)|\prod_{j=1}^{d}SGG(\omega_j^T(\mathrm{x}_i-\mathrm{m});0,\sigma_{lj},\sigma_{rj},c)\\
&=\Big(c_1|\det(W)|\Big)^n\Big(\prod_{j=1}^{d}(\sigma_{lj}+\sigma_{rj})\Big)^{-n}\cdot\\
&\cdot\prod_{i=1}^{n}\prod_{j=1}^{d}\exp\Big[-\beta^{\frac{c}{2}}\Big(\frac{|z_{ij}|}{\sigma_{lj}}\mathbb{1}_{\{z_{ij}\leq 0\}}+\frac{|z_{ij}|}{\sigma_{rj}}\mathbb{1}_{\{z_{ij}>0\}}\Big)^c\Big],
\end{aligned}
$$

where $c_1=\Big(\frac{c}{\Gamma(\frac{1}{c})}\sqrt{\beta}\Big)^d$ and $\beta=\frac{\Gamma(\frac{3}{c})}{\Gamma(\frac{1}{c})}$. Now we take the log-likelihood function, i.e.

$$
\begin{aligned}
&\ln(L(X;\mathrm{m},W,\sigma_l,\sigma_r,c))\\
&=\ln\Big(\Big(c_1|\det(W)|\Big)^n\Big(\prod_{j=1}^{d}(\sigma_{lj}+\sigma_{rj})\Big)^{-n}\Big)+\\
&\sum_{i=1}^{n}\sum_{j=1}^{d}\Big[-\beta^{\frac{c}{2}}\Big(\frac{|z_{ij}|}{\sigma_{lj}}\mathbb{1}_{\{z_{ij}\leq 0\}}+\frac{|z_{ij}|}{\sigma_{rj}}\mathbb{1}_{\{z_{ij}>0\}}\Big)^c\Big]\\
&=\ln\Big(\Big(c_1|\det(W)|\Big)^n\Big(\prod_{j=1}^{d}(\sigma_{lj}+\sigma_{rj})\Big)^{-n}\Big)-\\
&\beta^{\frac{c}{2}}\sum_{j=1}^{d}\Big(\sigma_{lj}^{-c}\sum_{i\in I_j}|z_{ij}|^c+\sigma_{rj}^{-c}\sum_{i\in I_j'}|z_{ij}|^c\Big)\\
&=\ln\Big(\Big(c_1|\det(W)|\Big)^n\Big(\prod_{j=1}^{d}(\sigma_{lj}+\sigma_{rj})\Big)^{-n}\Big)-\\
&\beta^{\frac{c}{2}}\sum_{j=1}^{d}\Big(\sigma_{lj}^{-c}s_{1j}+\sigma_{rj}^{-c}s_{2j}\Big).
\end{aligned}
$$

We fix m, W, c and maximize the log-likelihood function over σ_l and σ_r. In such a case we have to solve the following system of equations

$$
\begin{aligned}
\tfrac{\partial\ln(L(X;\mathrm{m},W,\sigma_l,\sigma_r,c))}{\partial\sigma_{lj}}&=-\tfrac{n}{\sigma_{lj}+\sigma_{rj}}+c\beta^{\frac{c}{2}}\sigma_{lj}^{-c-1}s_{1j}=0,\\
\tfrac{\partial\ln(L(X;\mathrm{m},W,\sigma_l,\sigma_r,c))}{\partial\sigma_{rj}}&=-\tfrac{n}{\sigma_{lj}+\sigma_{rj}}+c\beta^{\frac{c}{2}}\sigma_{rj}^{-c-1}s_{2j}=0,
\end{aligned}
$$

for $j=1,\ldots,d$. By simple calculations and substituting $\sigma_{rj}=\sigma_{lj}\left(\frac{s_{2j}}{s_{1j}}\right)^{\frac{1}{c+1}}=\sigma_{lj}\tau$ we obtain the expressions for the estimators

$$
\hat{\sigma}_{lj}^c(\mathrm{m},W)=\tfrac{c}{n}\beta^{\frac{c}{2}}s_{1j}^{\frac{c}{c+1}}g_j(\mathrm{m},W,c),\hat{\tau}_j(\mathrm{m},W)=\left(\frac{s_{2j}}{s_{1j}}\right)^{\frac{1}{c+1}}
$$

and

$$
\hat{\sigma}_{rj}^c(\mathrm{m},W)=\hat{\sigma}_{lj}^c(\mathrm{m},W)\cdot\hat{\tau}_j^c(\mathrm{m},W)=\tfrac{c}{n}\beta^{\frac{c}{2}}s_{2j}^{\frac{c}{c+1}}g_j(\mathrm{m},W,c).
$$

Substituting it into the log-likelihood function, we get

$$\begin{aligned}
&\hat{L}(X;\mathrm{m},W,c) = \\
&\Big(c_1|\det(W)|\Big)^n\Big(\prod_{j=1}^{d}(\tfrac{c}{n})^{\frac{1}{c}}\sqrt{\beta}g_j(\mathrm{m},W)^{\frac{c+1}{c}}\Big)^{-n}\cdot e^{-\frac{nd}{c}} \\
&= \left(\frac{nc^{c-1}}{e\Gamma(\frac{1}{c})^c}\right)^{\frac{dn}{c}}\Big(\frac{1}{|\det(W)|^{\frac{c}{c+1}}}\prod_{j=1}^{d}g_j(\mathrm{m},W)\Big)^{-\frac{n(c+1)}{c}} \\
&= \left(\frac{\kappa n}{ce}\right)^{\frac{dn}{c}}\Big(|\det(W)|^{-\frac{c}{c+1}}\prod_{j=1}^{d}g_j(\mathrm{m},W)\Big)^{-\frac{n(c+1)}{c}}
\end{aligned}$$

where $\kappa = \Big(\frac{c}{\Gamma(\frac{1}{c})}\Big)^c$. □

8. Appendix B

Before we prove Theorem 3..2, we recall the following lemma.

Lemma 8..1. *Let $A = (a_{ij})_{1\le i,j\le d}$ be a differentiable map from real numbers to $d\times d$ matrices then*

$$\frac{\partial\det(A)}{\partial a_{ij}} = \mathrm{adj}^T(A)_{ij}, \tag{8..1}$$

where $\mathrm{adj}(A)$ *stands for the adjugate of A, i.e. the transpose of the cofactor matrix.*

Proof. By the Laplace expansion $\det A = \sum_{j=1}^{d}(-1)^{i+j}a_{ij}M_{ij}$ where M_{ij} is the minor of the entry in the i-th row and j-th column. Hence

$$\frac{\partial\det A}{\partial a_{ij}} = (-1)^{i+j}M_{ij} = \mathrm{adj}^T(A)_{ij}.$$

□

Now we are ready to calculate the gradient of the function l.

Proof of Theorem 3..2. Let us start with the partial derivative of $\ln l$ with respect to m. We have

$$\begin{aligned}
&\tfrac{\partial\ln l(X;\mathrm{m},W,c)}{\partial\mathrm{m}_k} = \\
&\sum_{j=1}^{d}\tfrac{\partial\ln(g_j(\mathrm{m},W))}{\partial\mathrm{m}_k} = \sum_{j=1}^{d}\frac{1}{s_{1j}^{\frac{1}{c+1}}+s_{2j}^{\frac{1}{c+1}}}\frac{\partial\left(s_{1j}^{\frac{1}{c+1}}+s_{2j}^{\frac{1}{c+1}}\right)}{\partial\mathrm{m}_k} = \\
&\sum_{j=1}^{d}\frac{1}{s_{1j}^{\frac{1}{c+1}}+s_{2j}^{\frac{1}{c+1}}}\left(\frac{1}{(c+1)s_{1j}^{\frac{c}{c+1}}}\frac{\partial s_{1j}}{\partial\mathrm{m}_k}+\frac{1}{(c+1)s_{2j}^{\frac{c}{c+1}}}\frac{\partial s_{2j}}{\partial\mathrm{m}_k}\right).
\end{aligned}$$

Now, we need $\frac{\partial s_{1j}}{\partial \mathrm{m}_k}$ and $\frac{\partial s_{2j}}{\partial \mathrm{m}_k}$, therefore

$$\begin{aligned}
&\frac{\partial s_{1j}}{\partial \mathrm{m}_k} = \\
&\sum_{i\in I_j} \frac{\partial |\omega_j^T(\mathrm{x}_i-\mathrm{m})|^c}{\partial \mathrm{m}_k} = \sum_{i\in I_j} -c|\omega_j^T(\mathrm{x}_i-\mathrm{m})|^{c-1}\frac{\partial(\omega_j^T(\mathrm{x}_i-\mathrm{m}))}{\partial \mathrm{m}_k} = \\
&\sum_{i\in I_j} c|\omega_j^T(\mathrm{x}_i-\mathrm{m})|^{c-1}\omega_{jk} = \\
&\sum_{i\in I_j} c(-1)^{c-1}\left(\omega_j^T(\mathrm{x}_i-\mathrm{m})\right)^{c-1}\omega_{jk}.
\end{aligned}$$

Analogously we get

$$\begin{aligned}
&\frac{\partial s_{2j}}{\partial \mathrm{m}_k} = \sum_{i\in I_j'} -c|\omega_j^T(\mathrm{x}_i-\mathrm{m})|^{c-1}\omega_{jk} = \\
&\sum_{i\in I_j'} -c\left(\omega_j^T(\mathrm{x}_i-\mathrm{m})\right)^{c-1}\omega_{jk}.
\end{aligned}$$

Hence

$$\begin{aligned}
&\frac{\partial \ln l}{\partial \mathrm{m}_k} = \sum_{j=1}^{d} \frac{1}{s_{1j}^{\frac{1}{c+1}}+s_{2j}^{\frac{1}{c+1}}} \\
&\left(\frac{1}{(c+1)s_{1j}^{\frac{c}{c+1}}}\sum_{i\in I_j} c|\omega_j^T(\mathrm{x}_i-\mathrm{m})|^{c-1}\omega_{jk} - \right. \\
&\left.\frac{1}{(c+1)s_{2j}^{\frac{c}{c+1}}}\sum_{i\in I_j'} c|\omega_j^T(\mathrm{x}_i-\mathrm{m})|^{c-1}\omega_{jk}\right).
\end{aligned}$$

Now we calculate the partial derivative of $\ln l(X;\mathrm{m},W,c)$ with respect to the matrix W. We have

$$\frac{\partial \ln l(X;\mathrm{m},W,c)}{\partial \omega_{pk}} = \frac{\partial \ln |\det(W)|^{-\frac{c}{c+1}}}{\partial \omega_{pk}} + \sum_{j=1}^{d}\frac{\partial \ln(g_j(\mathrm{m},W))}{\partial \omega_{pk}}.$$

To calculate the derivative of the determinant we use Jacobi's formula (see Lemma 8..1). Hence

$$\begin{aligned}
&\frac{\partial \ln(\det(W)^{-\frac{c}{c+1}})}{\partial \omega_{pk}} = \\
&\det(W)^{\frac{c}{c+1}}\Big(-\frac{c}{c+1}\Big)\det(W)^{-\frac{2c+1}{c+1}}\frac{\partial \det(W)}{\partial \omega_{pk}} \\
&= -\frac{c}{c+1}\det(W)^{-1}\mathrm{adj}^T(W)_{pk} \\
&= -\frac{c}{c+1}\frac{1}{\det(W)}\left[\det(W)(W^{-1})_{pk}^T\right] = -\frac{c}{c+1}(\omega^{-1})_{pk}^T,
\end{aligned}$$

where $(\omega^{-1})_{pk}^T$ is the element in the p-th row and k-th column of the matrix $(W^{-1})^T$. Now we calculate

$$\begin{aligned}
&\frac{\partial \ln(g_j(\mathrm{m},W))}{\partial \omega_{pk}} \\
&= \frac{1}{s_{1j}^{\frac{1}{c+1}}+s_{2j}^{\frac{1}{c+1}}}\frac{\partial\left(s_{1j}^{\frac{1}{c+1}}+s_{2j}^{\frac{1}{c+1}}\right)}{\partial \omega_{pk}} = \\
&\frac{1}{s_{1j}^{\frac{1}{c+1}}+s_{2j}^{\frac{1}{c+1}}}\left(\frac{1}{(c+1)s_{1j}^{\frac{c}{c+1}}}\frac{\partial s_{1j}}{\partial \omega_{pk}} + \frac{1}{(c+1)s_{2j}^{\frac{c}{c+1}}}\frac{\partial s_{2j}}{\partial \omega_{pk}}\right),
\end{aligned}$$

where

$$\frac{\partial s_{1j}}{\partial \omega_{pk}} = \sum_{i \in I_j} \frac{\partial |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^c}{\partial \omega_{pk}} =$$
$$\sum_{i \in I_j} -c|\omega_j^T(\mathrm{x}_i - \mathrm{m})|^{c-1} \frac{\partial \omega_j^T(\mathrm{x}_i - \mathrm{m})}{\partial \omega_{pk}} =$$
$$\begin{cases} 0, & \text{if } j \neq p \\ \sum\limits_{i \in I_p} -c(-1)^{c-1} \left(\omega_p^T(\mathrm{x}_i - \mathrm{m})\right)^{c-1} (\mathrm{x}_{ik} - \mathrm{m}_k), & \text{if } j = p \end{cases}$$

and x_{ik} is the k-th element of the vector x_i. Analogously we get

$$\frac{\partial s_{2j}}{\partial \omega_{pk}} = \begin{cases} 0, & \text{if } j \neq p \\ \sum\limits_{i \in I'_p} c|\omega_p^T(\mathrm{x}_i - \mathrm{m})|^{c-1}(\mathrm{x}_{ik} - \mathrm{m}_k), & \text{if } j = p. \end{cases}$$

Hence we obtain Now we calculate the derivative with respect to c.

$$\frac{\partial \ln l(X;\mathrm{m},W)}{\partial c} = -\frac{\partial}{\partial c}\left(\tfrac{c}{c+1} \ln |\det(W)|\right) + \sum_{j=1}^{d} \frac{\partial \ln(g_j(\mathrm{m},W))}{\partial c} =$$
$$-\tfrac{1}{(c+1)^2} \ln |\det(W)| + \sum_{j=1}^{d} \frac{1}{s_{1j}^{\frac{1}{c+1}} + s_{2j}^{\frac{1}{c+1}}} \frac{\partial}{\partial c}\left(s_{1j}^{\frac{1}{c+1}} + s_{2j}^{\frac{1}{c+1}}\right) =$$
$$-\tfrac{1}{(c+1)^2} \ln |\det(W)| +$$
$$\sum_{j=1}^{d} \frac{1}{s_{1j}^{\frac{1}{c+1}} + s_{2j}^{\frac{1}{c+1}}} \left(s_{1j}^{\frac{1}{c+1}} \frac{\partial}{\partial c}\left(\tfrac{1}{c+1} \ln s_{1j}\right) + s_{2j}^{\frac{1}{c+1}} \frac{\partial}{\partial c}\left(\tfrac{1}{c+1} \ln s_{2j}\right) \right)$$
$$= -\tfrac{1}{(c+1)^2} \ln |\det(W)| +$$
$$\sum_{j=1}^{d} \frac{1}{s_{1j}^{\frac{1}{c+1}} + s_{2j}^{\frac{1}{c+1}}} \Bigg(-\frac{s_{1j}^{\frac{1}{c+1}}}{(c+1)^2} \ln s_{1j} +$$
$$\tfrac{1}{c+1} s_{1j}^{-\frac{c}{c+1}} \frac{\partial s_{1j}}{\partial c} - \frac{s_{2j}^{\frac{1}{c+1}}}{(c+1)^2} \ln s_{2j} + \tfrac{1}{c+1} s_{2j}^{-\frac{c}{c+1}} \frac{\partial s_{2j}}{\partial c} \Bigg).$$

where

$$\frac{\partial s_{1j}}{\partial c} = \sum_{i \in I_j} |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^c \ln |\omega_j^T(\mathrm{x}_i - \mathrm{m})|,$$
$$\frac{\partial s_{2j}}{\partial c} = \sum_{i \in I'_j} |\omega_j^T(\mathrm{x}_i - \mathrm{m})|^c \ln |\omega_j^T(\mathrm{x}_i - \mathrm{m})|.$$

□