ICA based on asymmetry

P. Spurek, J. Tabor

*Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6,
30-348 Cracow, Poland*

P. Rola

*Department of Mathematics of the Cracow University of Economics, Rakowicka 27,
31-510 Cracow, Poland*

M. Ociepka

*Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6,
30-348 Cracow, Poland*

Abstract

Independent Component Analysis (ICA) - one of the basic tools in data analysis - aims to find a coordinate system in which the components of the data are independent. Most of existing methods are based on the minimization of the function of fourth-order moment (kurtosis). Skewness (third-order moment) has received much less attention.

In this paper we present a competitive approach to ICA based on the Split Gaussian distribution, which is well adapted to asymmetric data. Consequently, we obtain a method which works better than the classical approaches, especially in the case when the underlying density is not symmetric, which is a typical situation in the color distribution in images.

Keywords: ICA, Split Normal distribution, skewness.

1. Introduction

Independent component analysis (ICA) is one of the most popular methods of data analysis and preprocessing. Historically, Herault and Jutten [1]

Email addresses: przemyslaw.spurek@ii.uj.edu.pl, jacek.tabor@ii.uj.edu.pl (P. Spurek, J. Tabor), przemyslaw.rola@outlook.com (P. Rola)

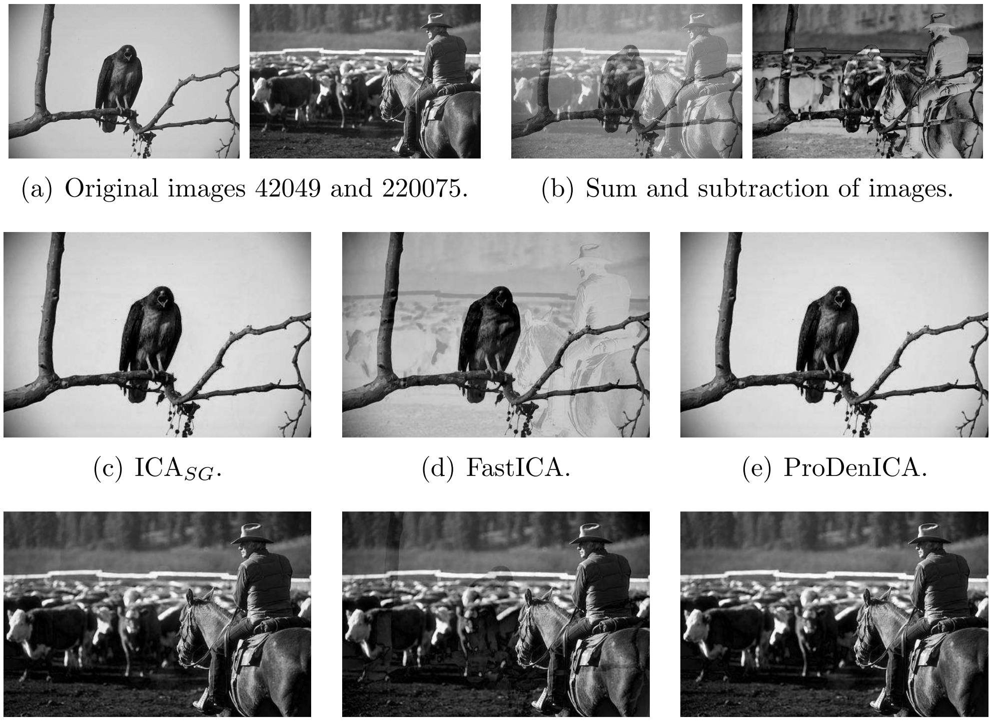
seem to be the first (around 1983) to have addressed the problem of ICA to separate mixtures of independent signals.

In signal processing ICA is a computational method for separating a multivariate signal into additive subcomponents and has been applied in magnetic resonance [2], MRI [3, 4], EEG analysis [5, 6, 7], fault detection [8], financial time series [9] and seismic recordings [10]. Moreover, it is hard to overestimate the role of ICA in pattern recognition and image analysis; its applications include face recognition [11, 12], facial action recognition [13], image filtering [14], texture segmentation [15], object recognition [16, 17], image modeling [18], embedding graphs in pattern-spaces [19, 20], multi-label learning [21] and feature extraction [22]. The calculation of ICA is discussed in several papers [23, 24, 25, 26, 27, 28, 29], where the problem is given various names, in particular it is also called "source separation problem".

ICA is similar in many aspects to principal component analysis (PCA). In PCA we look for an orthonormal change of basis so that the components are not linearly dependent (uncorrelated). ICA can be described as a search for the optimal basis (coordinate system) in which the components are independent. Let us now, for the readers convenience, describe how the ICA works. Data is represented by the random vector x and the components as the random vector s. The aim is to transform the observed data x into maximally independent components s with respect to some measure of independence. Typically we use a linear static transformation W, called the *transformation matrix*, combined with the formula $s = W\mathrm{x}$.

Most ICA methods are based on the maximization of non-Gaussianity. This follows from the fact that one of the theoretical foundations of ICA is given by the dual view at the Central Limit Theorem [30], which states that the distribution of the sum (average or linear combination) of N independent random variables approaches Gaussian as $N \to \infty$. Obviously if all source variables are Gaussian, ICA will not work.

The classical measure of non-Gaussianity is kurtosis (the forth central moment), which can be both positive or negative. Random variables that have a negative kurtosis are called subgaussian, and those with the positive one are called supergaussian. Supergaussian random variables have typically a "spiky" pdf with heavy tails, i.e. the pdf is relatively large at zero and at large values of the variable, while being small for intermediate values (ex. the Laplace distribution). Typically non-Gaussianity is measured by the absolute value of kurtosis (the square of kurtosis can also be used). Thus many methods of finding independent components are based on fitting a



(a) Original images 42049 and 220075.

(b) Sum and subtraction of images.

(c) ICA$_{SG}$.

(d) FastICA.

(e) ProDenICA.

(f) ICA$_{SG}$.

(g) FastICA.

(h) ProDenICA.

Figure 1: Comparison of images separation by our method (ICA$_{SG}$), with FastICA and ProDenICA.

density with similar kurtosis as the data, and consequently are very sensitive to the existence of outliers. Moreover, typically data sets are bounded, and therefore the credible estimation of tails is not easy. Another problem with these methods, is that they usually assume that the underlying density is symmetric, which is rarely the case.

In our work we introduce and explore a new approach ICA$_{SG}$, based on the asymmetry of the data, which can be measured by the third central moment (skewness). Any symmetric data, in particular gaussian, has skewness equal to zero. Negative values of skewness indicate the data skewed to the left and the positive ones indicate the data skewed to the right[1]. Consequently, skewness is a natural measure of non-Gaussianity. In our approach, instead of approximating the data by product of densities with heavy tails, we approximate it by a product of asymmetric densities (so called Split Gaussians).

Contrary to classical approaches which consider third or fourth central moment, our algorithm is based on second moments. This is a consequence of the fact that Split Gaussian distributions arise from merging two opposite halves of normal distributions in their common mode (for more information see Section 4). Therefore we use only second order moments to describe skewness in dataset, and therefore we obtain an effective ICA method which is resistant to outliers.

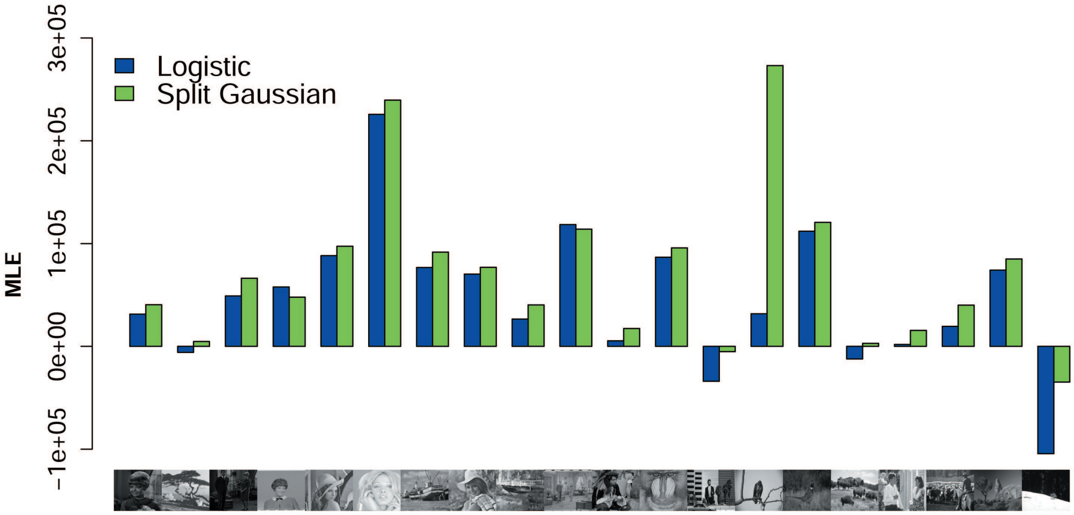
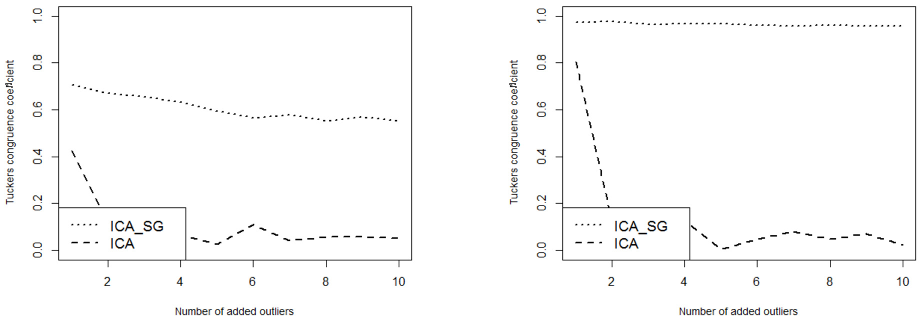


Figure 2: MLE estimation for image histograms with respect to Logistic and Split Gaussian distributions.

The results of classical ICA and ICA$_{SG}$ in the case of image separation (for

[1] By skewed to the left, we mean that the left tail is long relative to the right tail. Similarly, skewed to the right means that the long tail is on the right-hand side.

more detail comparison we refer to Section 6) is presented in Fig. 1. 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 than the classical FastICA approach, compare Fig. 1(c) to Fig. 1(d) and Fig. 1(f) to Fig. 1(g). In the case of classical ICA we can see artifacts in background, which means that the method does not separate signal properly. On the other hand, ProDenICA and ICA$_{SG}$ almost perfectly recovered images, compare Fig. 1(c) to Fig. 1(e) and Fig. 1(f) to Fig. 1(h).



(a) Resistance on outliers in the case of data with heavy tails.

(b) Resistance on outliers in the case of skew data.

Figure 3: Comparison between our approach and classical ICA in the case of resistance on outliers.

In general, ICA$_{SG}$ in most cases gives better results than other ICA methods, see Section 6, while its numerical complexity lies below the methods which obtain comparable results, that is ProDenICA and PearsonICA. This is caused in particular by the fact that asymmetry is more common than heavy tails in real data sets – we performed the symmetry test by using R package `lawstat` [31] with 5 percent confidence ratio, and it occurred that all image datasets we used in our paper have asymmetric densities. We also verified it in the case of density estimation of our images. We found optimal parameters of Logistic and Split Gaussian distributions and compared the values of MLE function in Fig. 2. As we see, in most cases Split Gaussian distribution fits the data better than the Logistic one.

Summarizing the results obtained in the paper, our method works better than classical approaches for asymmetric data, and is more resistant to outliers (see Example 1.1).

Example 1.1. *We consider the data with heavy tails (a sample from the Logistic distribution) and skewed ones (a sample from the Split Normal distribution). We added to the data outliers uniformly generated from rectangle* $[\min(X_1) - \mathrm{sd}(X_1), \max(X_1) + \mathrm{sd}(X_1)] \times [\min(X_2) - \mathrm{sd}(X_2), \max(X_2) + \mathrm{sd}(X_2)]$*, where* $\mathrm{sd}(X_i)$ *is a standard deviation of the* i*-th coordinate of* X*. In Fig. 3 we present how the absolute value of the Tucker's congruence coefficient (the similarity measure of extracted factors, see Section 6) is changing when we add the outliers.*

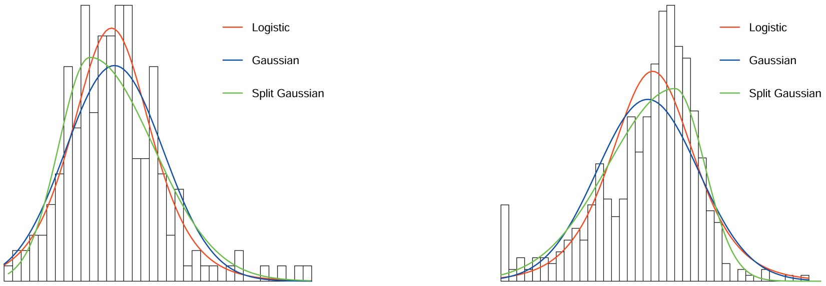
As we see, ICA_{SG} *is more stable and deals better with outliers in the data, which follows from the fact that classical ICA typically depends on the moments of order four, while our approach uses moments of order two.*

This paper is arranged as follows. In the second section, we discuss related works. In the third, the theoretical background of our approach to ICA is presented. We introduce a cost function which uses the General Split Gaussian distribution and show that it is enough to minimize it respectively to only two parameters: vector $\mathrm{m} \in \mathbb{R}^d$ and $d \times d$ matrix W. We also calculate the gradient of the cost function, which is necessary for the efficient use in the minimization procedure. The last section describes numerical experiments. The effects of our algorithm are illustrated on simulated and real datasets.

2. Related works

Various ICA methods were discussed in [23, 24, 25, 26, 27, 28]. Herault and Jutten seem to be the first who introduced the ICA around 1983. They proposed an iterative real-time algorithm based on a neuro-mimetic architecture, which nevertheless, can show the lack of convergence in a number of cases [32]. It is worth mentioning that in their framework, higher-order statistics were not introduced explicitly. Giannakis et al. [33] addressed the issue of identifiability of ICA in 1987 using third-order cumulants. However, the resulting algorithm required an exhaustive search.

Lacoume and Ruiz [34] sketched a mathematical approach to the problem using higher-order statistics, which can be interpreted as a measure of fitting independent components. Cardoso [35, 36] focused on the algebraic properties of the fourth-order cumulants (kurtosis) what is still a popular approach [37]. Unfortunately kurtosis has some drawbacks in practice, when its value has to be estimated from a measured sample. The main problem is



(a) Logistic, Split Normal and Classical Gaussian distribution fitted to data with heavy tails.

(b) Logistic, Split Normal and Classical Gaussian distribution fitted to skew data.

Figure 4: Logistic, Split Normal and Classical Gaussian distribution fitted to data with heavy tails and skew one.

that kurtosis can be very sensitive to the outliers. Its value may depend on only a few observations in the tails of the distribution. In high-dimensional problems, where separation process contains PCA (for dimension reduction), whitening (for scale normalization), and standard ICA this effect is called a small sample size problem [38, 39]. This is caused by the fact that for the high-dimensional data sets ICA algorithms tend to extract the independent features simply by the projections that isolate single or very few samples (outliers). To address the difficulty random pursuit and locality pursuit methods were applied [39].

Another commonly used solution is to use skewness [40, 41, 42, 43] instead of kurtosis. Unfortunately, skewness has received much less attention than kurtosis, and consequently methods based on skewness are usually not well theoretically justified.

One of the most popular ICA method dedicated to the skew data is PearsonICA [44, 45], which minimizes mutual information using a Pearson [46] system-based parametric model. The model covers a wide class of source distributions including skewed distributions. The Pearson system is defined by the differential equation

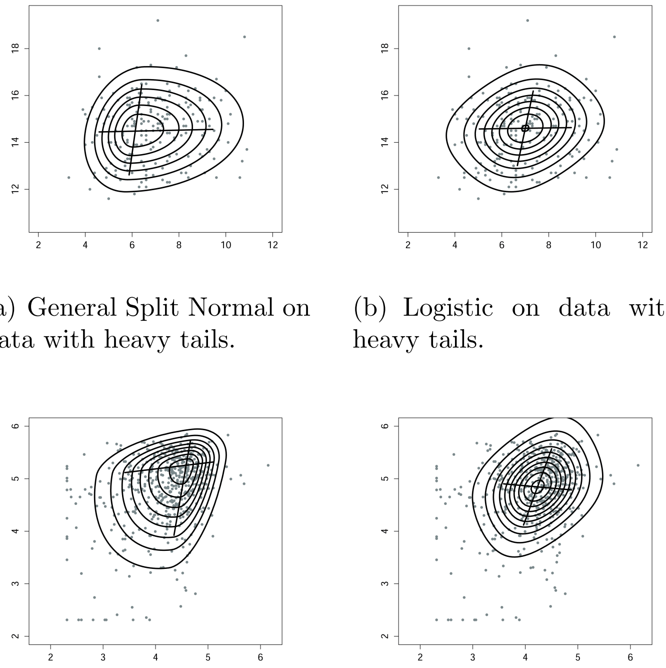
$$f'(x) = \frac{(a_1 x - a_0) f(x)}{b_0 + b_1 x + b_2 x^2},$$

where a_0, a_1, b_0, b_1 and b_2 are the parameters of the distribution. The parameters of the Pearson system can be estimated using the method of moments. Therefore such algorithms have strong limitations connected with the optimization procedure. The main problems are number of parameters which have to be fitted and numerical efficiency of the minimization procedure.

An important measure of fitting independent components is given by negentropy [47]. FastICA [48], one of the most popular implementations of ICA, uses this approach. Negentropy is based on the information-theoretic quantity of (differential) entropy. This concept leads to the mutual information which is the natural information-theoretic measure of the independence of random variables. Consequently, one can use it as the criterion for finding the ICA transformation [28, 49]. It can be shown that minimization of the mutual information is roughly equivalent to maximization of negentropy and it is easier to estimate since we do not need additional parameters. ProDenICA [50, 51] 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 kernel Hilbert space, and make use of the "kernel trick" to search over this space efficiently. The use of a function space makes it possible to adapt to a variety of sources and thus makes ProDenICA algorithms more robust to varying source distributions.

A somewhat similar approach to ICA is based on the maximum likelihood estimation [27]. It is closely connected to the infomax principle since the likelihood is proportional to the negative of mutual information. In recent publications, the maximum likelihood estimation is one of the mot popular [24, 52, 53, 54, 55, 56, 57] approaches to ICA. Maximum likelihood approach needs the source pdf. In the classical ICA it is common to use the super-Gaussian logistic density or other heavy tails distributions.

In this paper we present ICA$_{SG}$, a method which joins the positive aspects of classical ICA$_{SG}$ approaches with recent ones like ProDenICA or Pearson ICA. First of all we use a General Split Gaussian distribution, which uses second order moments to describe skewness in dataset, and therefore is relatively robust to noise or outliers. The GSG distribution can be fitted by minimizing a simple function, which depends on only two parameters $\mathrm{m} \in \mathbb{R}^d$, $W \in \mathcal{M}(\mathbb{R}^d)$, see Theorem 5.1. Moreover we calculate its gradient,



(a) General Split Normal on data with heavy tails.

(b) Logistic on data with heavy tails.

(c) General Split Normal on skew data.

(d) Logistic on skew data.

Figure 5: Logistic and General Split Normal distributions fitted to data with heavy tails and skew ones.

and therefore we can use numerically efficient gradient type algorithms, see Theorem 5.2.

3. Theoretical justification

Let us describe the idea[2] behind ICA [30]. Suppose that we have a random vector X in $\mathbb{R}^d$ which is generated by the model with the density F. Then it is well-known that components of X are independent iff there exist one-dimensional densities $f_1, \ldots, f_d \in \mathcal{D}_{\mathbb{R}}$, where by $\mathcal{D}_{\mathbb{R}}$ we denote the set of

[2] In fact it is one of the possible approaches, as there are many explanations which lead to similar formula.

densities on $\mathbb{R}$, such that

$$F(\mathrm{x}) = f_1(x_1) \cdot \ldots \cdot f_d(x_d), \text{ for } \mathrm{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d.$$

Now suppose that the components of X are not independent, but that we know (or suspect) that there is a basis A (we put $W = A^{-1}$) such that in that base the components of X become independent. This may be formulated in the form

$$F(\mathrm{x}) = \det(W) \cdot f_1(\omega_1^T(\mathrm{x} - \mathrm{m})) \cdot \ldots \cdot f_d(\omega_d^T(\mathrm{x} - \mathrm{m})) \text{ for } x \in \mathbb{R}^d, \tag{3.1}$$

where $\omega_i^T(\mathrm{x} - \mathrm{m})$ is the i-th coefficient of $\mathrm{x} - \mathrm{m}$ (the basis is centered in m) in the basis A (ω_i denotes the i-th column of W). Observe, that for a fixed family of one-dimensional densities $\mathcal{F} \subset \mathcal{D}_{\mathbb{R}}$, the set of all densities given by (3.1) for $f_i \in \mathcal{F}$, forms an affine invariant set of densities.

Thus, if we want to find such a basis that components become independent, we need to search for a matrix W and one-dimensional densities such that the approximation

$$F(\mathrm{x}) \approx \det(W) \cdot f_1(\omega_1^T(\mathrm{x} - \mathrm{m})) \cdot \ldots \cdot f_d(\omega_d^T(\mathrm{x} - \mathrm{m})), \text{ for } \mathrm{x} \in \mathbb{R}^d,$$

is optimal. However, before proceeding to practical implementations, we need to precise:

1. how to measure the above approximation,
2. how to deal with data X, since we do not have the density,
3. how to work with the family of all possible densities.

The answer to the first point is simple and is given by the Kullback-Leibler divergence, which is defined to be the integral:

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x,$$

where p and q denote the densities of P and Q. This can be written as

$$D_{\mathrm{KL}}(P\|Q) = h(P) - MLE(P, Q),$$

where h is the classical Shannon entropy. Thus to minimize the Kullback-Leibler divergence, we can equivalently maximize the MLE. This is helpful, since for a discrete data X we have nice estimator of the LE (likelihood estimation):

$$LE(X,Q) = \frac{1}{|X|}\sum_{\mathrm{x}\in X} \ln(q(x)).$$

Thus we arrive at the following problem.

Problem [reduced]. *Let X be a data set. Find an unmixing matrix W, center* m*, and densities $f_1,\dots,f_d \in \mathcal{D}_{\mathbb{R}}$ so that the value*

$$\begin{aligned}&LE(X,f_1,\dots,f_d,\mathrm{m},W) =\\ &\tfrac{1}{|X|}\sum_{\mathrm{x}\in X} \ln(f_1(\omega_1^T(\mathrm{x}-\mathrm{m}))\dots f_d(\omega_d^T(\mathrm{x}-\mathrm{m}))) + \ln(\det(W)) =\\ &\tfrac{1}{|X|}\sum_{i=1}^{d}\sum_{\mathrm{x}\in X} \ln(f_i(\omega_i^T(\mathrm{x}-\mathrm{m}))) + \ln(\det(W))\end{aligned}$$

is maximized.

However, there is still a problem with the last point, as the search over the space of all densities $\mathcal{D}_{\mathbb{R}}$ is not feasible. Thus, we naturally have to reduce our search to a subclass of all densities $\mathcal{F}$ (which should be parametrized by a finite amount of parameters).

Problem [final]. *Let $X \subset \mathbb{R}^d$ be a data set and $\mathcal{F} \subset \mathcal{D}_{\mathbb{R}}$ be a set of densities. Find an unmixing matrix W, center* m*, and densities $f_1,\dots,f_d \in \mathcal{F}$ so that the value*

$$\frac{1}{|X|}\sum_{i=1}^{d}\sum_{\mathrm{x}\in X} \ln(f_i(\omega_i^T(\mathrm{x}-\mathrm{m}))) + \ln(\det(W))$$

is maximized.

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[3]. In other words we have to choose a family of densities which is distant from Gaussian ones.

In the classical ICA approach it is common to use the super-Gaussian logistic distribution:

$$f(x;\mu,s) = \frac{e^{\frac{x-\mu}{s}}}{s\left(1+e^{\frac{x-\mu}{s}}\right)^2} = \frac{1}{4s}\,\mathrm{sech}^2\!\left(\frac{x-\mu}{2s}\right).$$

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

The main difference between the gaussian and super-gaussian is the existence of the heavy tails. This can be also viewed as the difference in the fourth moments.

However, such a choice leads to some negative consequences, namely the model is very sensitive to outliers. Moreover, if the data is not-symmetric, the approximation could not give the expected results, as the model consists only of symmetric densities.

The idea behind this paper was to choose the model of densities which wouldn't have the two above disadvantages. So, instead of choosing the family which differs from the Gaussians by the size of tail (fourth moment), we chose a family which would allow estimation of asymmetric densities – Split Gaussian distribution [58].

Example 3.1. *In Fig. 4 and Fig. 5 we present a comparison between the Logistic and the Split Normal distribution in 1d and 2d respectively. In experiments we use the classical skew dataset Lymphoma [59, 60] and the classical heavy tails dataset Australian athletes [61]. In the case of heavy tails both methods work nice, since real dataset represent heavy tails which are not symmetric and the skew model is able to detect it. On the other hand, in the case of skew data Split Normal gives essentially better results.*

4. Split Gaussian distribution

In this section we present our density model. A natural direction for extending the normal distribution is the introduction of some skewness, and several proposals have indeed emerged, both in the univariate and multivariate case, see [62, 63, 64]. One of the most popular approaches is the Split Normal (SN) distribution, or the Split Gaussian (SG) distribution [58]. In our paper we use a generalization of this model, which we call the General Split Normal (GSN) distribution.

We start from the one-dimensional case. After that we present a possible generalization of this definition to the multidimensional setting, which corresponds with the formula (3.1). Contrary to the Split Gaussian distribution, we skip the assumption of the orthogonality of coordinates (often called principal components), and obtain an ICA model.

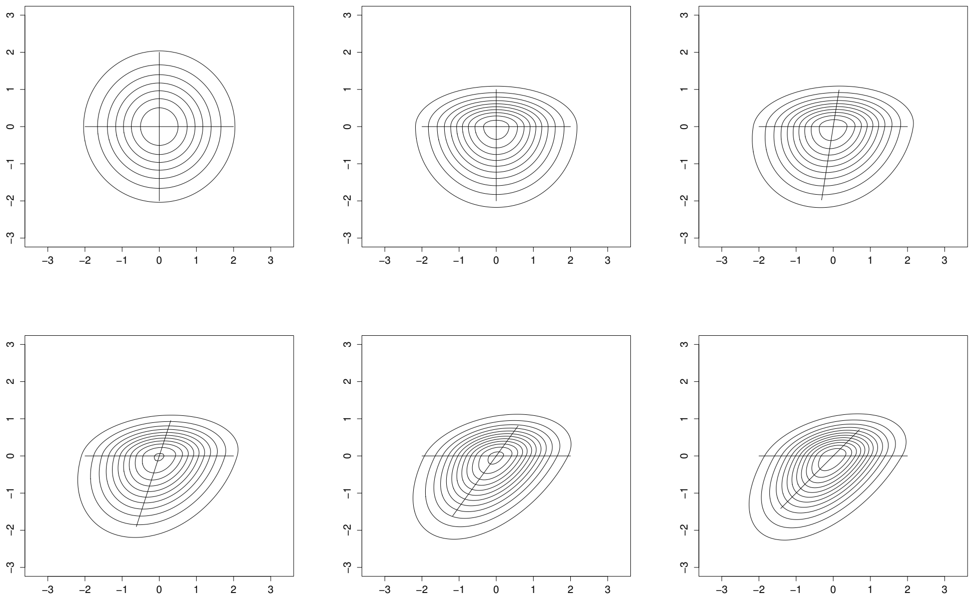


Figure 6: Level sets of the General Split Normal distribution with different parameters.

4.1. One-dimensional case

The density of the one-dimensional Split Gaussian distribution is given by the formula

$$SN(x; m, \sigma^2, \tau^2) = \left\{ \begin{array}{ll} c \cdot \exp[-\frac{1}{2\sigma^2}(x-m)^2], & \text{for } x \leq m, \\ c \cdot \exp[-\frac{1}{2\tau^2\sigma^2}(x-m)^2], & \text{for } x > m, \end{array} \right.$$

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). In 1982 John [65] showed that the likelihood function can be expressed in an intensive form, in which the scale parameters σ and τ are a function of the location parameter m (see Theorem 3.1 proved by [64]). Thanks to this theorem we can maximize the likelihood function numerically with respect to a single parameter m only. The rest of parameters are explicitly given by simple formulas.

4.2. Multidimensional Split Gaussian distribution

A natural generalization of the univariate split normal distribution to the multivariate settings was presented by [64]. Roughly speaking, authors assume that a vector $\mathrm{x} \in \mathbb{R}^d$ follows the multivariate Split Normal distribution, if its principal components are orthogonal and follow the one-dimensional Split Normal distribution.

Definition 4.1 (Definition 2.2. [64])**.** *A density of the multivariate Split Normal distribution is given by*

$$SN_d(\mathrm{x}; \mathrm{m}, \Sigma, \tau) = \prod_{j=1}^{d} SN(\omega_j^T(\mathrm{x} - \mathrm{m}); 0, \sigma_j^2, \tau_j^2),$$

where ω_j is the eigenvector corresponding to the j-th largest eigenvalue in the spectral decomposition of $\Sigma = W\mathcal{A}W^T$ and $\mathrm{m} = [m_1, \ldots, m_d]^T$, $\mathcal{A} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ and $\tau = [\tau_1^2, \ldots, \tau_d^2]$.

One can easily observe that the principal components $\omega_j^T \mathrm{x}$ are independent.

For this generalization a similar theorem, like in the one-dimensional case, is valid. We can extract the maximum likelihood estimation by maximizing the function with respect to two parameters $\mathrm{m} \in \mathbb{R}^d$ and $W \in \mathcal{M}_d(\mathbb{R})$ where columns of W are orthonormal vectors ($\mathcal{M}_d(\mathbb{R})$ denotes the set of d-dimensional square matrices).

We may use this theorem for numerical maximization of the likelihood function w.r.t. m and W. Unfortunately, the optimization process on Stiefel manifold (the set of orthogonal matrices) studied by [66] is numerically ineffective and requires additional tools. This problem can be omitted by using Eulerian angles described by [67]. In the two-dimensional case, W is explicitly parametrized as

$$W = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}, \quad -\frac{\pi}{2} < \theta \leq \frac{\pi}{2}.$$

In such a case we can straightforwardly apply standard numerical optimization algorithm.

Both of these solutions can be applied. Nevertheless, unnatural assumption of the orthogonality of principal components causes two negative effects: the optimization process is time consuming and the model with the restriction that the coordinates are orthogonal can not accommodate data as good as the general one. Therefore, in this article we use more flexible model – the General Split Normal [68] distribution:

Definition 4.2. *A density of the multivariate General Split Normal distribution is given by*

$$GSN_d(\mathrm{x}; \mathrm{m}, W, \sigma^2, \tau^2) = \det(W) \prod_{j=1}^{d} SN(\omega_j^T(\mathrm{x} - \mathrm{m}); 0, \sigma_j^2, \tau_j^2),$$

where ω_j *is the* j*-th column of non-singular matrix* W*,* $\mathrm{m} = (m_1, \ldots, m_d)^T$*,* $\sigma = (\sigma_1, \ldots, \sigma_d)$ *and* $\tau = (\tau_1, \ldots, \tau_d)$*.*

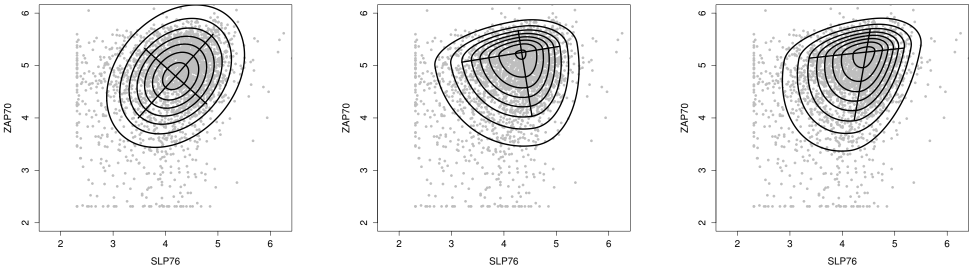
Our model is a natural generalization of the multivariate Split Normal distribution proposed in [64] (see Definition 4.1) and is given in the form formulated by (3.1) for the set of Split Gaussian densities. Clearly every Split Normal distribution is a General Split Normal distribution.

The above generalization is flexible and allows to fit data with greater precision, see Fig. 7. The level sets of the GSN distribution with different parameters are presented in Fig. 6. We skip the constraints of orthogonality of the principal components. Consequently, we can apply the standard optimization procedure directly. In the next section we discuss how to fit data in our model.

5. Maximum likelihood estimation

In the previous section we introduced the GSN distribution. Now we show how to use the likelihood estimation in our setting. As it was mentioned, we have to maximize the likelihood function with respect to four parameters. In the case of the General Split Normal distribution (contrary to the classical Gaussian one) we do not have explicit formulas and consequently we heave to solve the optimization problem.

In the first subsection, we reduce our problem to the simpler one by introducing the function l. Minimization of l is equivalent to maximization of the likelihood function. In the second subsection we present how to minimize our function by using the gradient method.



(a) Level sets of classical Gaussian distribution.
(b) Level sets of Split Gaussian distribution.
(c) Level sets of General Split Gaussian distribution.

Figure 7: Comparison between fitting Gaussian, Split Gaussian and General Split distribution on [59, 60]. Observe that, contrary to Split Gaussian, General Split Gaussian does not have orthogonal basis.

5.1. Optimization problem

The density of the GSN distribution depends on four parameters $\mathrm{m} \in \mathbb{R}^d$, $W \in \mathcal{M}(\mathbb{R}^d)$, $\sigma \in \mathbb{R}^d$, $\tau \in \mathbb{R}^d$. We can find them by minimizing the simpler function, which depends on only $m \in \mathbb{R}^d$ and $W \in \mathcal{M}(\mathbb{R}^d)$. Other parameters are given by explicit formulas.

Theorem 5.1. *Let* $\mathrm{x}_1, \ldots, \mathrm{x}_n$ *be given. Then the likelihood maximized w.r.t.* σ *and* τ *is*

$$\hat{L}(X; \mathrm{m}, W) = \left(\frac{2n}{\pi e}\right)^{dn/2} \left(\frac{1}{|\det(W)|^{\frac{2}{3}}} \prod_{j=1}^{d} g_j(\mathrm{m}, W)\right)^{-3n/2}, \tag{5.1}$$

where

$$\begin{gathered} g_j(\mathrm{m}, W) = s_{1j}^{1/3} + s_{2j}^{1/3}, \\ s_{1j} = \sum_{i \in I_j} [\omega_j^T(\mathrm{x}_i - \mathrm{m})]^2, I_j = \{i = 1, \ldots, n \colon \omega_j^T(\mathrm{x}_i - \mathrm{m}) \leq 0\}, \\ s_{2j} = \sum_{i \in I_j^c} [\omega_j^T(\mathrm{x}_i - \mathrm{m})]^2, I_j^c = \{i = 1, \ldots, n \colon \omega_j^T(\mathrm{x}_i - \mathrm{m}) > 0\}, \end{gathered}$$

and the maximum likelihood estimators of σ_j^2 *and* τ_j *are*

$$\hat{\sigma}_j^2(\mathrm{m}, W) = \tfrac{1}{n} s_{1j}^{2/3} g_j(\mathrm{m}, W), \quad \hat{\tau}_j(\mathrm{m}, W) = \left(\frac{s_{2j}}{s_{1j}}\right)^{1/3}.$$

Proof. See Appendix 8. □

Thanks to the above theorem, instead of looking for the maximum of the likelihood function, it is enough to obtain the maximum of the simpler function (5.1) which depends on two parameters $\mathrm{m} \in \mathbb{R}^d$ and $W \in \mathcal{M}(\mathbb{R}^d)$

$$l(X; \mathrm{m}, W) = \frac{1}{|\det(W)|^{\frac{2}{3}}} \prod_{j=1}^{d} g_j(\mathrm{m}, W), \tag{5.2}$$

where ω_j stands for the j-th column of matrix W. Consequently, maximization of (5.1) is equivalent to minimization of (5.2), see the following corollary.

Corollary 5.1. *Let $X \subset \mathbb{R}^d$, $\mathrm{m} \in \mathbb{R}^d$, $W \in \mathcal{M}(\mathbb{R}^d)$ be given, then*

$$\operatorname*{argmax}_{\mathrm{m},W} \hat{L}(X; \mathrm{m}, W) = \operatorname*{argmin}_{\mathrm{m},W} l(X; \mathrm{m}, W).$$

5.2. Gradient

One of the possible methods of optimization is the gradient method. Since the minimum of l is equal to the minimum of $\ln(l)$, in this subsection we calculate the gradient of $\ln(l)$. Before we prove suitable Theorem 5.2, we recall the following lemma.

Lemma 5.1. *Let $A = (a_{ij})_{1 \leq i,j \leq d}$ be a differentiable map from real numbers to $d \times d$ matrices then*

$$\frac{\partial \det(A)}{\partial a_{ij}} = \mathrm{adj}^T(A)_{ij}, \tag{5.3}$$

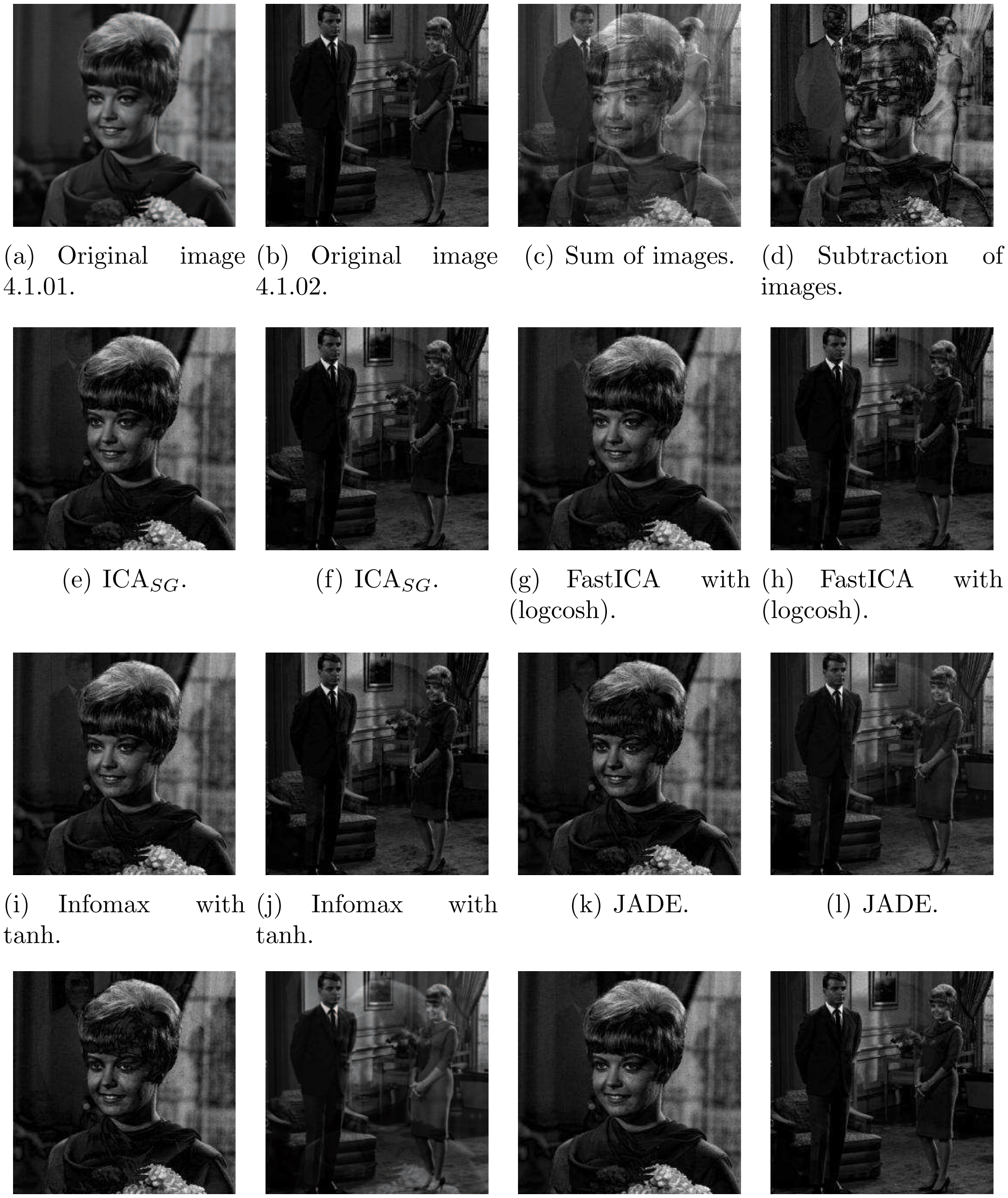
where $\mathrm{adj}(A)$ stands for the adjugate of A, i.e. the transpose of the cofactor matrix.

Proof. By the Laplace expansion $\det A = \sum\limits_{j=1}^{d} (-1)^{i+j} a_{ij} M_{ij}$ 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 gradient of our cost function.



(a) Original image 4.1.01.

(b) Original image 4.1.02.

(c) Sum of images.

(d) Subtraction of images.

(e) ICA$_{SG}$.

(f) ICA$_{SG}$.

(g) FastICA with (logcosh).

(h) FastICA with (logcosh).

(i) Infomax with tanh.

(j) Infomax with tanh.

(k) JADE.

(l) JADE.

(m) PearsonICA.

(n) PearsonICA.

(o) ProDenICA.

(p) ProDenICA.

Figure 8: Results of image separation with the uses of various ICA algorithms.

Theorem 5.2. *Let $X \subset \mathbb{R}^d$, $\mathrm{m} = (\mathrm{m}_1, \ldots, \mathrm{m}_d)^T \in \mathbb{R}^d$, $W = (\omega_{ij})_{1 \leq i,j \leq d}$ nonsingular be given. Then $\nabla_{\mathrm{m}} \ln l(X; \mathrm{m}, W) = \left(\frac{\partial \ln l(X;\mathrm{m},W)}{\partial \mathrm{m}_1}, \ldots, \frac{\partial \ln l(X;\mathrm{m},W)}{\partial \mathrm{m}_d} \right)^T$, where*

$$\tfrac{\partial \ln l(X;\mathrm{m},W)}{\partial \mathrm{m}_k} = \sum_{j=1}^{d} \tfrac{-1}{s_{1j}^{\frac{1}{3}} + s_{2j}^{\frac{1}{3}}} \left(\tfrac{1}{3 s_{1j}^{\frac{2}{3}}} \sum_{i \in I_j} 2\omega_j^T (\mathrm{x}_i - \mathrm{m}) \omega_{jk} + \tfrac{1}{3 s_{2j}^{\frac{2}{3}}} \sum_{i \in I_j^c} 2\omega_j^T (\mathrm{x}_i - \mathrm{m}) \omega_{jk} \right).$$

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

$$\begin{aligned} \tfrac{\partial \ln l(X;\mathrm{m},W)}{\partial \omega_{pk}} &= -\tfrac{2}{3}(\omega^{-1})_{pk}^T + \tfrac{1}{s_{1p}^{\frac{1}{3}} + s_{2p}^{\frac{1}{3}}} \left(\tfrac{1}{3} s_{1p}^{-\frac{2}{3}} \sum_{i \in I_p} 2\omega_p^T (\mathrm{x}_i - \mathrm{m})(\mathrm{x}_{ik} - \mathrm{m}_k) + \right. \\ &\left. + \tfrac{1}{3} s_{2p}^{-\frac{2}{3}} \sum_{i \in I_p^c} 2\omega_p^T (\mathrm{x}_i - \mathrm{m})(\mathrm{x}_{ik} - \mathrm{m}_k) \right), \end{aligned}$$

and

$$\begin{aligned} s_{1j} &= \sum_{i \in I_j} [\omega_j^T (\mathrm{x}_i - \mathrm{m})]^2, I_j = \{ i = 1, \ldots, n \colon \omega_j^T (\mathrm{x}_i - \mathrm{m}) \leq 0 \}, \\ s_{2j} &= \sum_{i \in I_j^c} [\omega_j^T (\mathrm{x}_i - \mathrm{m})]^2, I_j^c = \{ i = 1, \ldots, n \colon \omega_j^T (\mathrm{x}_i - \mathrm{m}) > 0 \}. \end{aligned}$$

Proof. See Appendix 9. □

Thanks to the above theorem we can use gradient descent, a first-order optimization algorithm. To find a local minimum of the cost function $\ln(l)$ using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function, see Algorithm 1.

At the end of this section we present comparison of computational efficiency between ICA$_{SG}$ and various ICA methods, see Fig. 9. In our experiment we consider the classical image separation problem, where we mixed two images by adding and subtracting them. We use ten pairs of images. Each pair was scaled to different sizes. In Fig. 9 we present mean value of computation time. FastICA, Infomax and JADE are the most effective but do not solve the problem of image separation sufficiently well, see Tab. 1. On the other hand, the ProDenICA which gives comparable result to ICA$_{SG}$, is much slower.

```
**Algorithm 1** :
  Input
    data set $X$
  Initial conditions
    initialization of mean vector $\mathrm{m} = \mathrm{mean}(X)$
    initialization of matrix $W = \mathrm{cov}(X)$
  Gradient algorithm
    obtain new values of m and $V$ by applying gradient method for function
  $\log(l)$ (see formula 5.1):

    $$(\mathrm{m}, W) = \operatorname*{argmin}_{\bar{\mathrm{m}}, \bar{W}} \log(l(X; \bar{\mathrm{m}}, \bar{W})),$$

    where

    $$\nabla_{\mathrm{m}} \ln l(X; \mathrm{m}, W)$$
    $$\nabla_{W} \ln l(X; \mathrm{m}, W)$$

    are given by Theorem 5.2
    calculate $\sigma \in \mathbb{R}^d$ and $\tau \in \mathbb{R}^d$ by using Theorem 5.1
  Return value
    return optimal ICA basis $(\mathrm{m}, W)$.
```

6. Experiments and analysis

To compare our method to classical ones we use Tucker's congruence coefficient [69] (uncentered correlation) defined by

$$Cr(s, \bar{s}) = \frac{\sum_{i=1}^{d} s_i \bar{s}_i}{\sqrt{\sum_{i=1}^{d} s_i^2} \sqrt{\sum_{i=1}^{d} \bar{s}_i^2}}.$$

Its values range between -1 and $+1$. It can be used to study 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. In the case of ICA methods multiplying by the scalar any of the sources do not change results. Therefore the sign of congruence coefficient is not important and we can compare absolute value of Tucker's congruence.

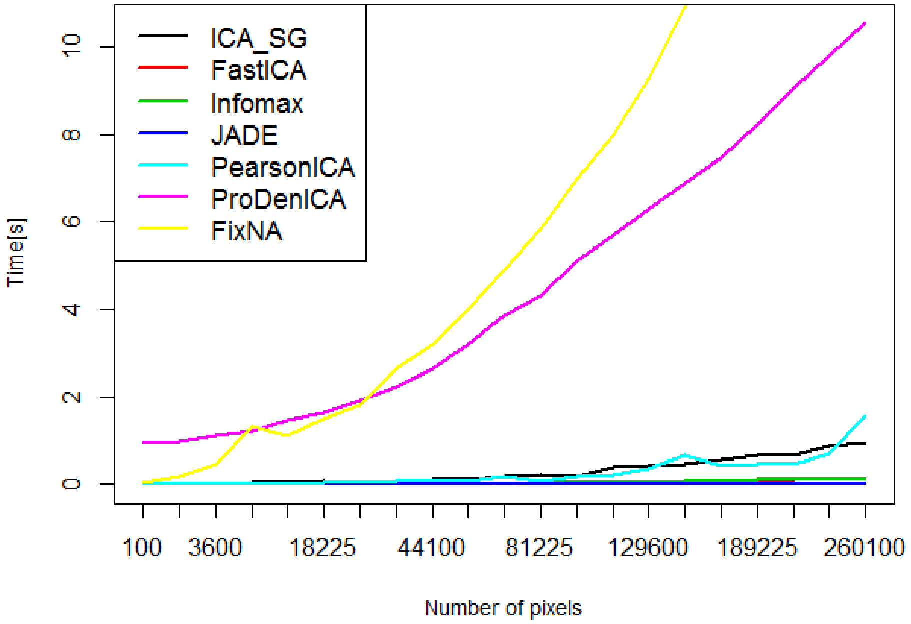


Figure 9: Comparison of computational efficiency between ICA$_{SG}$ and various ICA methods.

We evaluate our method in the context of images, sound, hyperspectral unmixing and EEG data. For comparison we use R package `ica` [70], `PearsonICA` [71], `ProDenICA` [72], `tsBSS` [73]. The most popular method used in practice is FastICA [48, 74] algorithm, which uses negentropy. In this context we can use three different functions to estimate neg-entropy: logcosh, exp and kurtosis. We also compare our method with algorithm using Information-Maximization (Infomax) approach [49]. Similarly to FastICA we consider three possible nonlinear functions: hyperbolic tangent, logistic and extended Infomax. We also consider algorithm which uses Joint Approximate Diagonalization of Eigenmatrices (JADE) proposed by Cardoso and Souloumiac's [75, 75, 74].

One of the most popular ICA methods dedicated for skew data is PearsonICA [44, 45], which minimizes mutual information using a Pearson [46] system-based parametric model. Another model we consider is ProDenICA [50, 51], 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 kernel Hilbert space, and make use of the kernel trick to search over this space efficiently. We also compare our method with FixNA [76], method for blind source separation problem.

6.1. Separation of images

One of the most popular application of ICA is the separation of images. In our experiments we use four images from the USC-SIPI Image Database of size 256×256 pixels (4.1.01, 4.1.06, 4.1.02, 4.1.03) and eight of size 512×512 pixels (4.2.04, 4.2.02, boat.512, elaine.512, 5.2.10, 5.2.08, 5.3.01, 4.2.03). We also use 8 images from the Berkeley Segmentation Dataset of size 482×321 with indexes (#119082, #42049, #43074, #38092, #157055, #220075, #295087, #167062). We make random pairs of above images and use them as a source signal, combined by the mixing matrix $A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. From practical point of view, we simply obtain two new images by adding and dividing sources pictures. Our goal is to reconstruct original images by using only the knowledge about mixed ones. The visualization of this process we present in Fig. 8. The results of this experiment are presented in Tab. 1 where we exhibit Tucker's congruence coefficients.

In the case of the Tucker's congruence coefficient measure almost in all situation we obtain better results. The ICA$_{SG}$ method essentially better recovers original signals. In Fig. 8(e) and 8(f) we can sow that ICA$_{SG}$ almost perfectly recovers source signal.

6.2. Cocktail-party problem

In this subsection we compare our method with classical ones in the case of cocktail-party problem. 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 estimate the two original speech signals.

In our experiments we use signal obtained by mixing synthetic sources[4] (similar as before we use mixing matrix $A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$). Comparison between methods we present in Tab. 2. In the case of cocktail-party problem our method recovers sources signal better then classical methods.

6.3. Hyperspectral Unmixing

Independent component analysis has been recently applied into hyperspectral unmixing as a result of its low computation time and its ability to perform without prior information. However, when applying ICA for hyperspectral unmixing, the independence assumption in the ICA model conflicts with the abundance sum-to-one constraint and the abundance nonnegative constraint in the linear mixture model, which affects the hyperspectral unmixing accuracy. Nevertheless, ICA was recently applied in this area [77, 78]. In this subsection we apply simple example which shows that our method can by used for spectral data.

Urban data [79, 80, 81] is one of the most widely used hyperspectral datasets used in the hyperspectral unmixing study. Each image has 307 × 307 pixels, each of which corresponds to a 2 × 2 m area. In this image, there are 210 wavelengths ranging from 400 nm to 2500 nm, resulting in a spectral resolution of 10 nm. After the channels 1–4, 76, 87, 101–111, 136–153 and 198–210 are removed (due to dense water vapor and atmospheric effects), there remain 162 channels (this is a common preprocess for hyperspectral unmixing analyses). There is ground truth [79, 80, 81], which contains 4 channels: #1 Asphalt, #2 Grass, #3 Tree and #4 Roof.

A highly mixed area is cut from the original data set in this experiment (similar example was showed in [77]), with the size of 200 × 150 pixels.

In our experiment we apply various ICA methods and report the Tucker's congruence coefficient measure between each layer and the closest reference channel, see Fig. 10. ICA$_{SG}$ and ProDenICA give layers which contain more information then the other approaches. Distance between four best channels to the reference ones we present in Tab. 3.

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

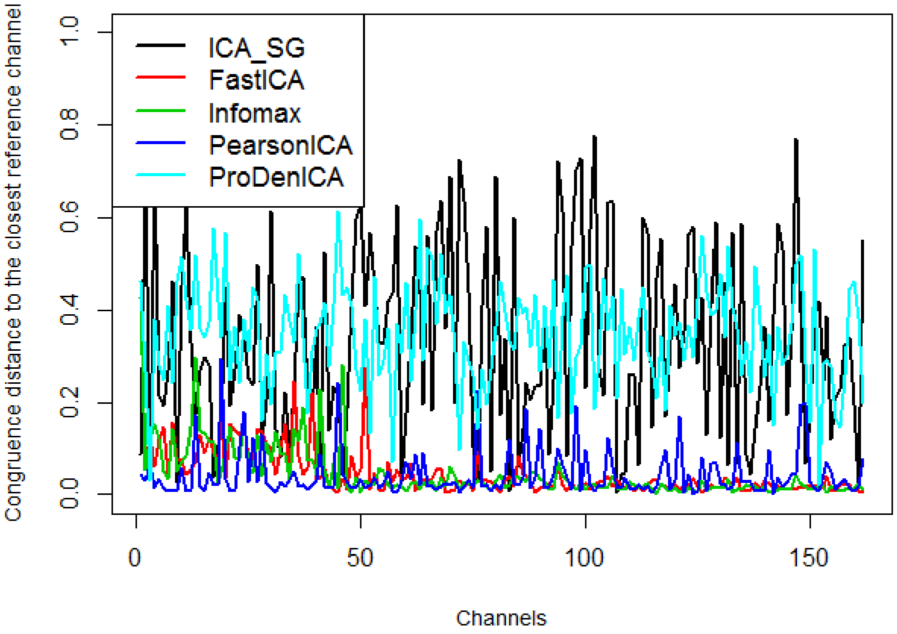


Figure 10: Congruence distance between layers obtain by different ICA algorithms and the closest reference channel.

6.4. 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 task like eye movements, blinks, muscle, heart and line noise e.t.c.. In this experiment we concentrate on eye movement and blink artifacts. Our goal here is to demonstrate that our method is capable of finding artifacts in 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 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 the projection strengths of the respective components onto the scalp sensors.

One EEG data set used in the analysis was collected from 40 scalp electrodes (see Fig. 11(a)). The second and the third are located very near to eye and can be understood as a base (we can use them for removing eye blinking artifacts). In Fig. 11(b) we present signals obtained by ICA$_{SG}$. 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. 11(c)). After removing selected signal and going back to the original situation we obtain signal (see Fig. 11(d)) without eye blinking artifacts (compare Fig. 11(a) with Fig. 11(d)).

7. Conclusion

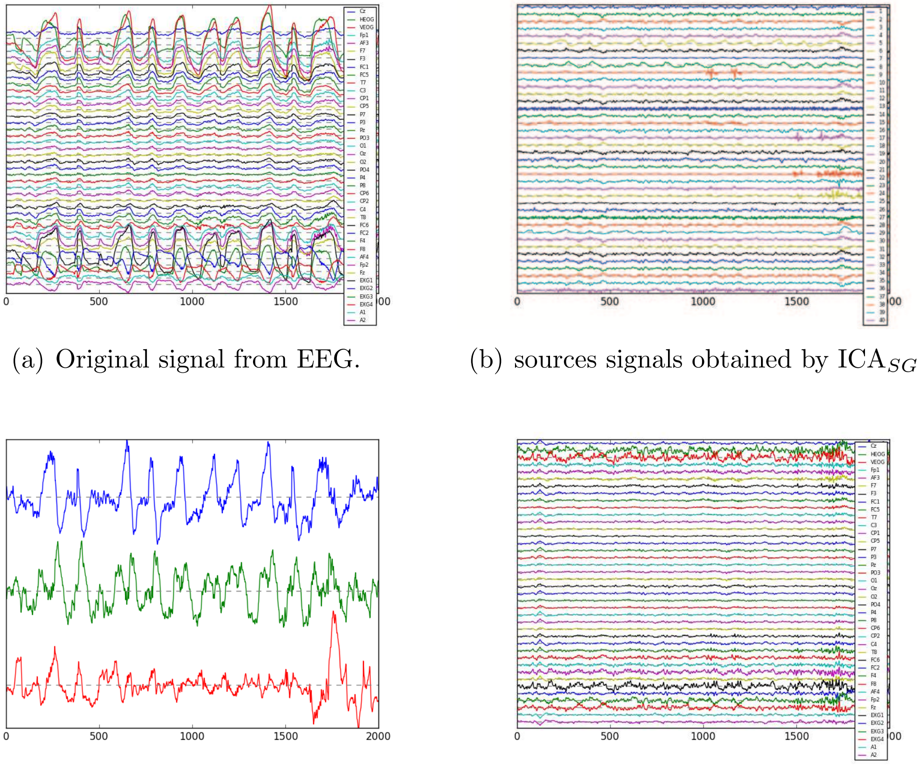
In our work we introduce and explore a new approach to ICA which is based on the asymmetry of the data. Roughly speaking in our approach, instead of approximating the data by product of densities with heavy tails, we approximate it by a product of asymmetric densities – the Split Gaussian distribution. Contrary to classical approaches which consider third or fourth central moment, our algorithm in practice is based on second moments. This is a consequence of the fact that Split Gaussian distributions arise from merging two opposite halves of normal distributions in their common mode. Therefore we use only second order moments to describe skewness in dataset, and therefore we obtain an effective ICA method which is resistant to outliers.

We verified our approach on images, sound and EEG data. In the case of source signal reconstructing our approach gives essentially better results (better recover original signals). The main reason is such that kurtosis is very sensitive to the outliers and that the asymmetry of the data is more popular than heavy tails in real data sets.

8. Appendix A

Proof of Theorem 5.1. Let $X = \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$. We write

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



(a) Original signal from EEG.

(b) sources signals obtained by ICA$_{SG}$.

(c) Three components 6, 9, 32.

(d) Original EEG signal with remove three components 6, 9, 32.

Figure 11: Results of ICA$_{SG}$in the case of EEG data.

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

Let us consider the likelihood function, i.e.

$$
\begin{aligned}
&L(X; \mathrm{m}, W, \sigma, \tau) = \prod_{i=1}^{n} GSN_d(\mathrm{x}_i; \mathrm{m}, W, \sigma, \tau) \\
&= \prod_{i=1}^{n} |\det(W)| \prod_{j=1}^{d} SN(\omega_j^T(\mathrm{x}_i - \mathrm{m}); 0, \sigma_j^2, \tau_j^2) \\
&= \Big(c_1 |\det(W)|\Big)^n \Big(\prod_{j=1}^{d} \sigma_j (1+\tau_j)\Big)^{-n} \prod_{i=1}^{n} \prod_{j=1}^{d} \exp\Big[-\tfrac{1}{2\sigma_j^2} z_{ij}^2 (\mathbb{1}_{\{z_{ij} \le 0\}} + \tau_j^{-2} \mathbb{1}_{\{z_{ij} > 0\}}) \Big],
\end{aligned}
$$

where $c_1 = \left(\sqrt{\frac{2}{\pi}}\right)^d$. Now we take the log-likelihood function, i.e.

$$
\begin{aligned}
&\ln(L(X; \mathrm{m}, W, \sigma, \tau)) \\
&= \ln\left(\left(c_1|\det(W)|\right)^n \Big(\prod_{j=1}^{d} \sigma_j(1+\tau_j)\Big)^{-n}\right) + \sum_{i=1}^{n}\sum_{j=1}^{d}\Big[-\tfrac{1}{2\sigma_j^2} z_{ij}^2 (\mathbb{1}_{\{z_{ij}\le 0\}} + \tau_j^{-2}\mathbb{1}_{\{z_{ij}>0\}})\Big] \\
&= \ln\left(\left(c_1|\det(W)|\right)^n \Big(\prod_{j=1}^{d} \sigma_j(1+\tau_j)\Big)^{-n}\right) - \tfrac{1}{2}\sum_{j=1}^{d}\Big(\sigma_j^{-2}\sum_{i\in I_j} z_{ij}^2 + \tfrac{\sigma_j^{-2}}{\tau_j^2}\sum_{i\in I_j^c} z_{ij}^2\Big) \\
&= \ln\left(\left(c_1|\det(W)|\right)^n \Big(\prod_{j=1}^{d} \sigma_j(1+\tau_j)\Big)^{-n}\right) - \sum_{j=1}^{d}\tfrac{1}{2\sigma_j^2}\Big(s_{1j} + \tfrac{1}{\tau_j^2}s_{2j}\Big).
\end{aligned}
$$

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

$$
\begin{aligned}
\tfrac{\partial \ln(L(X;\mathrm{m},W,\sigma,\tau))}{\partial \sigma_j} &= -\tfrac{n}{\sigma_j} + \sigma_j^{-3}(s_{1j} + \tau_j^{-2}s_{2j}) = 0, \\
\tfrac{\partial \ln(L(X;\mathrm{m},W,\sigma,\tau))}{\partial \tau_j} &= -\tfrac{n}{1+\tau_j} + \tfrac{s_{2j}}{\tau_j^3\sigma_j^2} = 0,
\end{aligned}
$$

for $j = 1, \ldots, d$. By simple calculations we obtain the expressions for the estimators

$$
\hat{\sigma}_j^2(\mathrm{m}, W) = \tfrac{1}{n} s_{1j}^{2/3} g_j(\mathrm{m}, W), \qquad \hat{\tau}_j(\mathrm{m}, W) = \left(\frac{s_{2j}}{s_{1j}}\right)^{1/3}.
$$

Substituting it into the log-likelihood function, we get

$$
\begin{aligned}
\hat{L}(\mathrm{m}, W) &= \left(\tfrac{2}{\pi}\right)^{\frac{dn}{2}} |\det(W)|^n \cdot \Big(\prod_{j=1}^{d} \tfrac{1}{\sqrt{n}} g_j(\mathrm{m}, W)^{\frac{3}{2}}\Big)^{-n} e^{-\frac{dn}{2}} \\
&= \left(\tfrac{2n}{\pi e}\right)^{\frac{dn}{2}} \Big(\tfrac{1}{|\det(W)|^{\frac{2}{3}}} \prod_{j=1}^{d} g_j(\mathrm{m}, W)\Big)^{-\frac{3n}{2}}.
\end{aligned}
$$

□

9. Appendix B

Proof of Theorem 5.2. Let us start with the partial derivative of $\ln(l)$ with respect to m. We have

$$
\tfrac{\partial \ln l(X;\mathrm{m},W)}{\partial \mathrm{m}_k} = \sum_{j=1}^{d} \tfrac{\partial \ln(g_j(\mathrm{m},W))}{\partial \mathrm{m}_k} = \sum_{j=1}^{d} \tfrac{1}{s_{1j}^{\frac{1}{3}} + s_{2j}^{\frac{1}{3}}} \tfrac{\partial (s_{1j}^{\frac{1}{3}} + s_{2j}^{\frac{1}{3}})}{\partial \mathrm{m}_k} \sum_{j=1}^{d} \tfrac{1}{s_{1j}^{\frac{1}{3}} + s_{2j}^{\frac{1}{3}}} \left(\tfrac{1}{3s_{1j}^{\frac{2}{3}}}\tfrac{\partial s_{1j}}{\partial \mathrm{m}_k} + \tfrac{1}{3s_{2j}^{\frac{2}{3}}}\tfrac{\partial s_{2j}}{\partial \mathrm{m}_k}\right).
$$

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

$$\frac{\partial s_{1j}}{\partial \mathrm{m}_k} = \sum_{i\in I_j} \frac{\partial[\omega_j^T(\mathrm{x}_i-\mathrm{m})]^2}{\partial \mathrm{m}_k} = \sum_{i\in I_j} 2\omega_j^T(\mathrm{x}_i-\mathrm{m})\frac{\partial\omega_j^T(\mathrm{x}_i-\mathrm{m})}{\partial \mathrm{m}_k} = \sum_{i\in I_j} -2\omega_j^T(\mathrm{x}_i-\mathrm{m})\omega_{jk}.$$

Analogously we get

$$\frac{\partial s_{2j}}{\partial \mathrm{m}_k} = \sum_{i\in I_j^c} -2\omega_j^T(\mathrm{x}_i-\mathrm{m})\omega_{jk}.$$

Hence

$$\frac{\partial \ln l}{\partial \mathrm{m}_k} = \sum_{j=1}^{d} \frac{-1}{s_{1j}^{\frac{1}{3}}+s_{2j}^{\frac{1}{3}}}\left(\frac{1}{3s_{1j}^{\frac{2}{3}}}\sum_{i\in I_j} 2\omega_j^T(\mathrm{x}_i-\mathrm{m})\omega_{jk} + \frac{1}{3s_{2j}^{\frac{2}{3}}}\sum_{i\in I_j^c} 2\omega_j^T(\mathrm{x}_i-\mathrm{m})\omega_{jk}\right).$$

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

$$\frac{\partial \ln l(X;\mathrm{m},W)}{\partial \omega_{pk}} = \frac{\partial \ln|\det(W)|^{-\frac{2}{3}}}{\partial \omega_{pk}} + \sum_{j=1}^{d}\frac{\partial \ln(g_j(\mathrm{m},W))}{\partial \omega_{pk}}.$$

To calculate the derivative of the determinant we use Jacobi's formula (see Lemma 5.1). Hence

$$\frac{\partial \ln(\det(W)^{-\frac{2}{3}})}{\partial \omega_{pk}} = \det(W)^{\frac{2}{3}}\Big(-\frac{2}{3}\Big)\det(W)^{-\frac{5}{3}}\frac{\partial \det(W)}{\partial \omega_{pk}} = -\frac{2}{3}\det(W)^{-1}\mathrm{adj}^T(W)_{pk}$$
$$= -\frac{2}{3}\frac{1}{\det(W)}\left[\det(W)(W^{-1})^T_{pk}\right] = -\frac{2}{3}(\omega^{-1})^T_{pk},$$

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

$$\frac{\partial \ln(g_j(\mathrm{m},W))}{\partial \omega_{pk}} = \frac{1}{s_{1j}^{\frac{1}{3}}+s_{2j}^{\frac{1}{3}}}\frac{\partial(s_{1j}^{\frac{1}{3}}+s_{2j}^{\frac{1}{3}})}{\partial \omega_{pk}} = \frac{1}{s_{1j}^{\frac{1}{3}}+s_{2j}^{\frac{1}{3}}}\left(\frac{1}{3s_{1j}^{\frac{2}{3}}}\frac{\partial s_{1j}}{\partial \omega_{pk}} + \frac{1}{3s_{2j}^{\frac{2}{3}}}\frac{\partial s_{2j}}{\partial \omega_{pk}}\right),$$

where

$$\frac{\partial s_{1j}}{\partial \omega_{pk}} = \sum_{i\in I_j}\frac{\partial[\omega_j^T(\mathrm{x}_i-\mathrm{m})]^2}{\partial \omega_{pk}} = \sum_{i\in I_j} 2\omega_j^T(\mathrm{x}_i-\mathrm{m})\frac{\partial \omega_j^T(\mathrm{x}_i-\mathrm{m})}{\partial \omega_{pk}} =$$
$$\begin{cases} 0, & \text{if } j \neq p \\ \sum\limits_{i\in I_p} 2\omega_p^T(\mathrm{x}_i-\mathrm{m})(\mathrm{x}_{ik}-\mathrm{m}_k), & \text{if } j = p \end{cases}$$

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

$$\frac{\partial s_{2j}}{\partial \omega_{pk}} = \begin{cases} 0, & \text{if } j \neq p, \\ \sum\limits_{i \in I_p^c} 2\omega_p^T(\mathrm{x}_i - \mathrm{m})(\mathrm{x}_{ik} - \mathrm{m}_k), & \text{if } j = p. \end{cases}$$

Hence we obtain

$$\tfrac{\partial \ln l}{\partial \omega_{pk}} = -\tfrac{2}{3}(\omega^{-1})^T_{pk} + \tfrac{1}{s_{1p}^{\frac{1}{3}} + s_{2p}^{\frac{1}{3}}} \Bigg(\tfrac{1}{3} s_{1p}^{-\frac{2}{3}} \sum_{i \in I_p} 2\omega_p^T(\mathrm{x}_i - \mathrm{m})(\mathrm{x}_{ik} - \mathrm{m}_k)$$
$$+ \tfrac{1}{3} s_{2p}^{-\frac{2}{3}} \sum_{i \in I_p^c} 2\omega_p^T(\mathrm{x}_i - \mathrm{m})(\mathrm{x}_{ik} - \mathrm{m}_k) \Bigg).$$

□

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.

References

[1] J. Herault, C. Jutten, Space or time adaptive signal processing by neural network models, in: Neural networks for computing, volume 151, AIP Publishing, pp. 206–211.

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

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

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

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

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

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

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

[9] K. Kiviluoto, E. Oja, Independent component analysis for parallel financial time series., in: ICONIP, volume 2, pp. 895–898.

[10] A. M. Haghighi, I. M. Haghighi, 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.

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

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

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

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

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

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

[17] D. Tao, L. Jin, Y. Yuan, Y. Xue, Ensemble manifold rank preserving for acceleration-based human activity recognition, IEEE transactions on neural networks and learning systems 27 (2016) 1392–1404.

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

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

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

[21] C. Xu, T. Liu, D. Tao, C. Xu, Local rademacher complexity for multi-label learning, IEEE Transactions on Image Processing 25 (2016) 1495–1507.

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

[23] P. Secchi, S. Vantini, P. Zanini, Hierarchical independent component analysis: a multi-resolution non-orthogonal data-driven basis, Computational Statistics & Data Analysis 95 (2016) 133–149.

[24] A. Hyvärinen, J. Karhunen, E. Oja, Independent component analysis, volume 46, John Wiley & Sons, 2004.

[25] T.-W. Lee, M. Girolami, T. J. Sejnowski, Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources, Neural computation 11 (1999) 417–441.

[26] J.-F. Cardoso, Source separation using higher order moments, in: Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on, IEEE, pp. 2109–2112.

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

[28] P. Comon, Independent component analysis, a new concept?, Signal processing 36 (1994) 287–314.

[29] B. Du, S. Wang, N. Wang, L. Zhang, D. Tao, L. Zhang, Hyperspectral signal unmixing based on constrained non-negative matrix factorization approach, Neurocomputing 204 (2016) 153–161.

[30] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural networks 13 (2000) 411–430.

[31] J. Gastwirth, G. Yulia, H. Wallace, L. Vyacheslav, M. Weiwen, N. Kimihiro, lawstat: Tools for Biostatistics, Public Policy, and Law, 2015. R package version 3.0.

[32] C. Jutten, J. Herault, Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture, Signal processing 24 (1991) 1–10.

[33] G. B. Giannakis, Y. Inouye, J. M. Mendel, Cumulant based identification of multichannel moving-average models, Automatic Control, IEEE Transactions on 34 (1989) 783–787.

[34] J.-L. Lacoume, P. Ruiz, Separation of independent sources from correlated inputs, Signal Processing, IEEE Transactions on 40 (1992) 3074–3078.

[35] J.-F. Cardoso, Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors, in: Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on, IEEE, pp. 3109–3112.

[36] J.-F. Cardoso, High-order contrasts for independent component analysis, Neural computation 11 (1999) 157–192.

[37] A. Sharma, K. K. Paliwal, Subspace independent component analysis using vector kurtosis, Pattern Recognition 39 (2006) 2227–2232.

[38] J. Yang, D. Zhang, J.-y. Yang, Is ica significantly better than pca for face recognition?, in: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, volume 1, IEEE, pp. 198–203.

[39] W. Deng, Y. Liu, J. Hu, J. Guo, The small sample size problem of ica: A comparative study and analysis, Pattern Recognition 45 (2012) 4438–4450.

[40] J. Stone, J. Porrill, N. Porter, I. Wilkinson, Spatiotemporal independent component analysis of event-related fmri data using skewed probability density functions, NeuroImage 15 (2002) 407–421.

[41] T. Kollo, Multivariate skewness and kurtosis measures with an application in ica, Journal of Multivariate Analysis 99 (2008) 2328–2338.

[42] Z. Liu, H. Qiao, Investigation on the skewness for independent component analysis, Science China Information Sciences 54 (2011) 849–860.

[43] J. Karvanen, V. Koivunen, Independent component analysis via optimum combining of kurtosis and skewness-based criteria, Journal of the Franklin Institute 341 (2004) 401–418.

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

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

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

[47] M. Gaeta, J.-L. Lacoume, et al., Source separation without a priori knowledge: the maximum likelihood solution, in: Proc. EUSIPCO, volume 90, Barcelona, Spain, pp. 621–624.

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

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

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

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

[52] F. Harroy, J.-L. Lacoume, Maximum likelihood estimators and cramer-rao bounds in source separation, Signal processing 55 (1996) 167–177.

[53] P. Comon, C. Jutten, Handbook of Blind Source Separation: Independent component analysis and applications, Academic press, 2010.

[54] R. J. Samworth, M. Yuan, et al., Independent component analysis via nonparametric maximum likelihood estimation, The Annals of Statistics 40 (2012) 2973–3002.

[55] V. Zarzoso, J. J. Murillo-Fuentes, R. Boloix-Tortosa, A. K. Nandi, Optimal pairwise fourth-order independent component analysis, Signal Processing, IEEE Transactions on 54 (2006) 3049–3063.

[56] J. J. Murillo-Fuentes, F. J. González-Serrano, A sinusoidal contrast function for the blind separation of statistically independent sources, Signal Processing, IEEE Transactions on 52 (2004) 3459–3463.

[57] J.-F. Cardoso, T. Adali, The maximum likelihood approach to complex ica., in: ICASSP (5), Citeseer, pp. 673–676.

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

[59] L. M. Maier, D. E. Anderson, P. L. De Jager, L. S. Wicker, D. A. Hafler, Allelic variant in ctla4 alters t cell phosphorylation patterns,

Proceedings of the National Academy of Sciences 104 (2007) 18607–18612.

[60] S. Pyne, X. Hu, K. Wang, E. Rossin, T.-I. Lin, L. M. Maier, C. Baecher-Allan, G. J. McLachlan, P. Tamayo, D. A. Hafler, et al., Automated high-dimensional flow cytometric data analysis, Proceedings of the National Academy of Sciences 106 (2009) 8519–8524.

[61] A. Clauset, C. R. Shalizi, M. E. Newman, Power-law distributions in empirical data, SIAM review 51 (2009) 661–703.

[62] A. Azzalini, A class of distributions which includes the normal ones, Scandinavian journal of statistics (1985) 171–178.

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

[64] M. Villani, R. Larsson, The multivariate split normal distribution and asymmetric principal components analysis, Communications in StatisticsTheory and Methods 35 (2006) 1123–1140.

[65] S. John, The three-parameter two-piece normal family of distributions and its fitting, Communications in Statistics-Theory and Methods 11 (1982) 879–885.

[66] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization algorithms on matrix manifolds, Princeton University Press, 2009.

[67] C. Khatri, K. Mardia, The von mises-fisher matrix distribution in orientation statistics, Journal of the Royal Statistical Society. Series B (Methodological) (1977) 95–106.

[68] P. Spurek, General split gaussian cross–entropy clustering, Expert Systems with Applications 68 (2017) 58–68.

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

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

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

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

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

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

[75] J.-F. Cardoso, A. Souloumiac, Blind beamforming for non-gaussian signals, in: Radar and Signal Processing, IEE Proceedings F, volume 140, IET, pp. 362–370.

[76] Z. Shi, Z. Jiang, F. Zhou, J. Yin, Blind source separation with nonlinear autocorrelation and non-gaussianity, Journal of computational and applied mathematics 229 (2009) 240–247.

[77] N. Wang, B. Du, L. Zhang, L. Zhang, An abundance characteristic-based independent component analysis for hyperspectral unmixing, IEEE Transactions on Geoscience and Remote Sensing 53 (2015) 416–428.

[78] C. F. Caiafa, E. Salerno, A. N. Proto, L. Fiumi, Blind spectral unmixing by local maximization of non-gaussianity, Signal Processing 88 (2008) 50–68.

[79] F. Zhu, Y. Wang, S. X. andBin Fan, C. Pan, Structured sparse method for hyperspectral unmixing, ISPRS Journal of Photogrammetry and Remote Sensing 88 (2014) 101–118.

[80] F. Zhu, Y. Wang, B. Fan, G. Meng, S. Xiang, C. Pan, Spectral unmixing via data-guided sparsity, CoRR abs/1403.3155 (2014).

[81] F. Zhu, Y. Wang, B. Fan, G. Meng, C. Pan, Effective spectral unmixing via robust representation and learning-based sparsity, CoRR abs/1409.0685 (2014).

	ICA$_{SG}$	FastICA			Infomax			JADE	PearsonICA	ProDenICA	FixNA
		logcosh	exp	kurtosis	tanh	tangent	logistic				
4.1.01	**-0.9818**	0.5481	-0.5457	-0.5485	0.548	-0.5484	-0.548	-0.5492	-0.5308	-0.0013	0.5503
4.1.02	**0.992**	0.6696	0.6644	0.6707	0.6695	0.6705	0.6695	0.6726	0.6696	-0.0981	-0.6761
4.1.06	**-0.9609**	-0.4297	-0.4297	-0.4296	-0.4297	-0.4297	-0.4296	-0.4296	-0.4297	0.4297	0.0148
4.1.03	**0.5664**	0.2062	0.2062	0.2057	0.2061	0.206	0.2058	0.2058	-0.2062	0.207	0.0127
4.2.04	**-0.5034**	0.0506	0.0528	-0.0499	0.0505	-0.0512	0.0508	0.0397	0.3123	-0.3164	0.1461
5.2.10	0.2893	-0.0719	-0.0749	0.0709	-0.0717	0.0727	-0.0722	-0.057	-0.4275	**0.4334**	-0.1979
4.2.02	**0.2305**	-0.0376	0.0203	-0.0017	0.0377	0.0265	0.0061	-0.0093	-0.1228	0.1282	0.1235
5.2.08	**0.5717**	0.1037	-0.0625	-0.0097	-0.1039	-0.0773	-0.0285	0.0086	-0.2913	-0.3091	-0.2931
boat.512	**0.3593**	0.0351	0.0314	-0.056	0.0343	-0.0449	0.0298	0.0356	-0.1046	-0.0461	0.3175
5.3.01	0.4316	0.0078	0.0138	-0.0262	0.0091	-0.008	0.0164	0.007	0.1061	0.0486	**-0.5303**
elaine.512	**0.5874**	0.32	0.32	-0.32	0.32	-0.32	0.32	0.32	-0.32	0.0287	0.2282
4.2.03	-0.0226	-0.3196	-0.3196	0.3201	-0.3196	0.3199	-0.3196	**-0.3202**	-0.3195	-0.048	-0.2554
119082	**0.9987**	0.5736	0.5736	0.5731	0.5737	0.5733	0.5735	0.5735	-0.032	0.5744	0.3695
157055	**0.389**	-0.3619	-0.3619	-0.3618	-0.3619	-0.3619	-0.3619	-0.3619	0.0046	0.3619	-0.2446
42049	**-0.7493**	0.3009	0.3028	-0.299	-0.3005	-0.3031	-0.3007	-0.2898	0.2596	0.0421	0.142
220075	0.4359	-0.5087	-0.5154	0.503	0.5074	**0.5168**	0.5081	0.4789	0.4838	-0.0645	-0.1839
43074	**-0.7371**	0.0344	0.0323	0.0429	0.0348	0.0404	0.0342	0.0324	0.0891	0.3925	0.2458
295087	-0.3997	-0.048	-0.0458	-0.0566	-0.0484	-0.0541	-0.0478	-0.0459	-0.1035	**0.4015**	-0.2406
38092	**-0.5949**	0.0555	0.0564	0.031	-0.0553	0.041	0.0557	0.0375	0.0535	0.4036	0.2614
167062	0.3255	-0.0025	-0.0041	0.0425	0.0021	0.0241	-0.0029	0.0306	0.0011	**0.7404**	-0.5495
4.1.01	0.9996	0.9956	0.9912	0.9964	0.9956	0.9963	0.9955	0.9977	0.7121	0.9966	0.9996
4.1.02	0.9995	0.9836	0.9759	0.9853	0.9836	0.985	0.9835	0.7706	0.7706	0.9965	0.9932
4.1.06	0.9876	1	1	0.9998	1	1	0.9999	0.9999	1	0.9999	0.9994
4.1.03	0.9934	0.9956	0.9959	0.9933	0.9951	0.9947	0.9941	0.9956	0.9956	0.9999	0.9981
4.2.04	0.9981	0.9871	0.986	0.9874	0.9871	0.9868	0.987	0.9921	0.9866	1	0.8869
5.2.10	0.9933	0.9861	0.9849	0.9864	0.9861	0.9858	0.986	0.9856	0.9856	1	0.8897
4.2.02	0.986	0.9561	0.9873	0.9999	0.9559	0.9784	0.956	0.9974	0.9557	0.9918	0.9609
5.2.08	0.9993	0.9421	0.9793	0.9995	0.9419	0.9682	0.942	0.9416	0.9416	0.9916	0.9476
boat.512	0.9588	0.9946	0.9957	0.9862	0.9949	0.9912	0.9961	0.9945	0.9506	0.9997	0.9378
5.3.01	0.9788	0.9999	0.9997	0.9988	0.9999	0.9999	0.9996	0.9805	0.9805	0.9997	0.9722
elaine.512	0.9871	1	1	1	1	1	1	0.9999	1	0.9995	0.7131
4.2.03	0.9954	0.9922	0.9919	0.9935	0.9922	0.9931	0.9922	0.9918	0.9918	0.9995	0.7929
119082	0.9982	0.9986	0.9986	0.9978	0.9987	0.9981	0.9984	0.9984	0.9986	1	0.7656
157055	0.9999	0.9999	0.9999	0.9997	1	0.9998	0.9999	1	1	1	0.7372
42049	0.9996	0.9886	0.9949	0.9825	0.9874	0.996	0.9882	0.9523	0.7291	0.9999	0.8845
220075	0.9999	0.9606	0.9732	0.9499	0.9583	0.9759	0.9598	0.7735	0.7735	0.9999	0.9378
43074	0.9962	0.8643	0.8231	0.9942	0.8711	0.9947	0.9963	0.9966	0.9668	0.9856	0.7814
295087	0.9892	0.88	0.841	0.9902	0.8864	0.9909	0.993	0.9581	0.9581	0.9852	0.8011
38092	1	0.9906	0.9902	0.9971	0.9906	0.9948	0.9904	0.9957	0.9917	1	0.7637
167062	0.9788	1	1	0.9983	1	0.9995	1	1	1	1	0.7419
4.1.01	0.9911	0.9956	0.9913	0.9965	0.9955	0.9963	0.9955	0.9977	0.7638	0.9978	0.9996
4.1.02	0.998	0.9836	0.976	0.9853	0.9835	0.985	0.9835	0.7096	0.7096	0.9945	0.9932
4.1.06	0.996	1	1	0.9998	1	1	0.9999	0.9999	1	0.9818	0.9994
4.1.03	0.9834	0.9956	0.9959	0.9933	0.9951	0.9947	0.9941	0.9956	0.9956	0.7809	0.9981
4.2.04	0.9995	0.9871	0.986	0.9875	0.9872	0.9868	0.987	0.9921	0.9872	0.9997	0.8869
5.2.10	0.9997	0.9861	0.985	0.9865	0.9862	0.9858	0.986	0.9862	0.9862	0.9999	0.8897
4.2.02	0.9965	0.9562	0.9874	0.9999	0.956	0.9785	0.9989	0.9974	0.9555	0.9958	0.9609
5.2.08	0.9923	0.9421	0.9794	0.9995	0.9419	0.9683	0.9958	0.9414	0.9414	0.9992	0.9476
boat.512	0.9946	0.9946	0.9957	0.9862	0.9949	0.9912	0.9961	0.9945	0.9503	0.9995	0.9378
5.3.01	1	0.9999	0.9997	0.9988	0.9999	0.9999	0.9995	0.9804	0.9804	0.9998	0.9722
elaine.512	0.9821	1	1	1	1	1	1	0.9999	1	0.9996	0.7131
4.2.03	0.9982	0.9921	0.9919	0.9936	0.9922	0.9931	0.9921	0.9918	0.9918	0.9995	0.7929
119082	0.9999	0.9986	0.9986	0.9978	0.9987	0.9981	0.9984	0.9984	0.9986	1	0.7656
157055	0.998	0.9999	1	0.9997	1	0.9998	0.9999	1	1	0.9999	0.7372
42049	0.9999	0.9887	0.9949	0.9825	0.9874	0.996	0.9881	0.9523	0.963	0.9999	0.8845
220075	0.9985	0.9607	0.9733	0.9499	0.9583	0.9759	0.9596	0.9196	0.9196	0.9999	0.9378
43074	0.9866	0.9962	0.9966	0.9941	0.9961	0.9947	0.9962	0.9966	0.975	0.9122	0.7814
295087	0.9976	0.9928	0.9935	0.99	0.9927	0.9909	0.9929	0.9674	0.9674	0.9622	0.8011
38092	0.9962	0.9906	0.9903	0.9971	0.9906	0.9949	0.9905	0.9957	0.9911	0.9999	0.7637
167062	1	1	1	0.9983	1	0.9995	1	1	1	1	0.7419

Table 1: The Tucker's congruence coefficient measure between original images and results of different ICA algorithms.

	ICA$_{SG}$	FastICA			Infomax			JADE	PearsonICA	ProDenICA	FixNA
		logcosh	exp	kurtosis	tanh	tangent	logistic				
source 1	**0.1597**	0.1097	0.1096	0.1101	0.1097	0.11	0.1097	0.1101	0.1097	0.1412	0.109
source 2	0.7739	0.7705	0.7713	0.7672	0.7705	0.7685	0.7705	0.7704	0.7704	**0.9998**	0.7751
source 2	**0.1388**	0.0899	0.0899	0.0908	0.0899	0.0899	0.0899	0.0908	0.0899	0.0984	0.0907
source 3	0.9435	0.9075	0.9076	0.898	0.9074	0.907	0.9074	0.9075	0.9075	**0.9989**	0.8988
source 3	**0.1985**	0.079	0.0791	0.079	0.079	0.079	0.079	0.079	0.0789	0.0843	0.0791
source 4	0.8453	0.8887	0.8882	0.8889	0.8887	**0.8892**	0.8887	0.8898	0.8898	0.8459	0.8882
source 4	**0.232**	0.0989	0.0989	0.099	0.0989	0.0989	0.0989	0.099	0.0989	0.1153	0.0989
source 5	0.7679	0.7798	0.7799	0.7793	0.7798	0.7798	0.7798	0.7801	0.7801	**0.9344**	0.7796
source 5	**0.1728**	0.0989	0.099	0.0988	0.0989	0.0989	0.0989	0.0989	0.0989	0.0963	0.0987
source 6	0.9424	0.9245	0.9243	0.9256	0.9246	0.925	0.9246	0.9245	0.9245	**0.9729**	0.9273
source 6	**0.15**	0.0404	0.0404	0.0402	0.0404	0.0404	0.0404	0.0402	0.0404	0.0567	0.0402
source 7	0.7417	0.7129	0.7134	0.707	0.7132	0.7125	0.7129	0.7124	0.7124	**0.9998**	0.7099
source 7	**0.1036**	0.0839	0.084	0.0839	0.0839	0.0839	0.0839	0.0839	0.084	0.093	0.0836
source 8	0.908	0.9016	0.9015	0.9019	0.9019	0.9019	0.9017	0.9014	0.9014	**0.9999**	0.9056
source 8	0.1166	0.1153	0.1156	0.1145	0.1152	0.1148	0.1153	0.1155	0.1149	**0.1427**	0.1147
source 9	0.8212	0.8136	0.8116	0.8195	0.8141	0.8174	0.8138	0.8165	0.8165	**0.9996**	0.8176
source 1	0.9999865	0.9999911	0.9999954	0.9999563	0.9999911	0.9999732	0.9999911	0.999956	0.9999883	0.7744559	0.9999956
source 2	0.9999721	0.9999823	0.9999886	0.9999386	0.9999823	0.999959	0.9999823	0.9999383	0.9999784	0.6339634	0.9999992
source 2	1	0.9999804	0.9999827	0.9996019	0.9999795	0.9999733	0.99998	0.9995758	0.9999803	0.9999318	0.9996485
source 3	1	0.9999856	0.9999877	0.9996271	0.9999849	0.9999795	0.9999853	0.9996018	0.9999856	0.9984157	0.9996721
source 3	0.99343	0.9999881	0.9999921	0.9999854	0.9999879	0.9999814	0.999988	0.9999889	0.9999763	0.8468988	0.9999924
source 4	0.999505	0.9999995	0.9999982	0.9999999	0.9999996	1	0.9999995	0.9999993	0.9999995	0.9861616	0.9999981
source 4	0.9981493	0.999991	0.9999917	0.9999874	0.9999909	0.9999905	0.999991	0.9999855	0.9999948	0.6486176	0.9999894
source 5	0.9991865	0.9999991	0.9999989	0.9999998	0.9999991	0.9999993	0.9999991	1	0.9999972	0.7956493	0.9999995
source 5	0.9999037	0.9999748	0.9999695	0.9999905	0.9999759	0.9999825	0.9999753	0.9999848	0.9999749	0.9523035	1
source 6	0.996108	0.9999982	0.9999966	0.9999997	0.9999985	0.9999997	0.9999983	0.9999999	0.9999982	0.4580579	0.999986
source 6	0.9997931	0.9999779	0.9999727	0.9999985	0.9999747	0.9999816	0.9999779	0.9999931	0.9999826	0.7219539	0.9999972
source 7	0.9997944	0.9999608	0.9999541	0.9999999	0.9999566	0.9999658	0.9999608	0.9999989	0.9999672	0.7210854	0.9999896
source 7	0.9954272	0.9999801	0.9999785	0.9999847	0.9999848	0.9999843	0.999982	0.9999802	0.999968	0.4435124	0.9999955
source 8	0.9997822	1	0.9999999	0.9999998	0.9999998	0.9999999	1	1	0.9999982	0.9054196	0.9999582
source 8	0.9991897	0.999998	0.9999852	0.9999667	0.9999993	0.9999896	0.9999986	0.9999913	0.9999984	0.6002706	0.9999881
source 9	0.9997275	0.9999619	0.999986	0.9998215	0.999954	0.999883	0.9999589	0.9999784	0.9999215	0.8268882	0.9998782

Table 2: The Tucker's congruence coefficient measure between original sound and results of different ICA algorithms in the case of cocktail-party problem.

	ICA$_{SG}$	FastICA			Infomax			PearsonICA	ProDenICA
		logcosh	exp	kurtosis	tanh	tangent	logistic		
#1 Asphalt	**0.6774**	0.2859	0.2864	-0.2595	-0.2972	-0.2954	-0.2972	0.20978	0.4928
#2 Grass	**-0.7784**	-0.2746	-0.2605	-0.2798	-0.2814	-0.2816	-0.2814	-0.2412	-0.4323
#3 Tree	**0.7267**	0.2338	0.2717	-0.2547	0.2441	0.2354	0.2442	0.2482	-0.5961
#4 Roof	**0.6666**	-0.4256	0.4279	0.4167	-0.4244	0.4301	-0.4244	0.4193	-0.6128

Table 3: The Tucker's congruence coefficient measure between reference layers and results of different ICA algorithms in the case of the urban data set.