A Tutorial on Blind Source Separation using Independent Component Analysis and Related Methods

Alexandru Paul Condurache,
Institute for Signal Processing, University of Lübeck,
D-23538 Lübeck, Germany
condurache@isip.uni-luebeck.de

***Abstract*—Blind Source Separation (BSS) is needed to recover several source signals from several mixture-signals. The mixture-signals are linear combinations of the sources signals. Such a setup is encountered for example when it is desired to recover the speech of N speakers, speaking simultaneously from N microphone signals placed at various positions in the same room with the speakers. Conversely the Independent Component Analysis (ICA) is a term covering a methods that aim to represent a set of observations from several random variables in terms of a linear combination of observations from several other random variables that are independent from one another. The solutions to the BSS problem usually imply some weak assumptions on the source signals, like for example independence. Thus, the ICA represents a possible solution to the BSS problem.**

I. Introduction

In the signal processing practice we are often confronted with the task of understanding the data at our disposal, or better said the task of representing the available data such that what is informative there can be easily seen. In order to define what is informative we make the assumption that the data we see is a combination of some main information carriers. Therefore this information carriers can not be represented as a combination of other information carriers and we say they are independent. On the search for these independent components we make use of the stochastic formalism. This gives us the possibility to look at the history of this "independent component" analysis, where first independence was considered equal to decorrelation, which of course is true only for Gauss-distributed variables.

We will discuss next methods for finding the independent components from some data. To better grasp the intuition behind ICA we will introduce it as the solution to a practical problem: the cocktail-party problem.

Consider the setup of a cocktail party organized in summer in a park. There are several people tacking simultaneously, birds singing in the trees and low-level music. Despite this cacophony of speech and noise, a human has no difficulty to listen to just one speaker. The purpose of this tutorial is to introduce methods to teach a machine to do the same thing.

If this technical report helped you in your research and want to use it, please cite as well "Statistical Pattern Recognition for Biometric Person Identification and Event Detection: Hysteresis and Sparse Classifiers, Dynamic Bayes Networks and the Gaussianity assumption" (see https://www.isip.uni-luebeck.de/mitarbeiter/alexandru-condurache.html), which originated it.

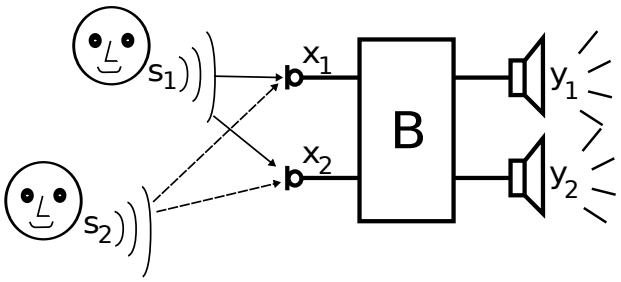

Fig. 1. An illustration of the "cocktail party" problem.

For the drawing in Fig. 1, the signals recorded by the two microphones are:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$$

where $s_1(t)$ and $s_2(t)$ are the two sources/speakers. With only the microphone signals at our disposal, we aim now to linearly recover (up to some negligible ambiguities) the original sources so that we can play them back over the loudspeakers. Our estimates of the sources are:

$$y_1(t) = b_{11}x_1(t) + b_{12}x_2(t)$$
$$y_2(t) = b_{21}x_1(t) + b_{22}x_2(t)$$

These estimates include the following ambiguities:

- The *scale* ambiguity: which means that we cannot determine the energy of the source signals.
- The *permutation* ambiguity: which means that we cannot determine the order of the source signals.

The purpose of BSS is to simply recover the (two) source signals. Currently this can be done under various assumptions. We divide these assumptions into core assumptions without which the sources can not be recovered at all and additional assumptions, which when holding lead to a simpler solution.

There are various alternative core assumptions:

- The signals should be *independent & nonGaussian*.
- The signals should have a time structure such that they are *nonstationary* or *nonwhite*.
- The source signals should be *positive*.

Under the first joint core assumptions, the ICA offers a

solution to the BSS problem by recovering the signals such that they are independent and their higher-order (higher than two) crosscorrelations cancel. In detail they are described as follows:

- the independence assumption means that the joint density of the speech signals can be factorized as:

$$p(s_1, s_2) = p(s_1)p(s_2). \tag{1}$$

- the nongaussianity assumption implies that the distribution of the source signals should not be Gaussian. At the same time it is also assumed that each source generates independent samples – i.e., the samples from each source are independent and identically distributed (i.i.d.).

Under either of the other assumptions, the sources can still be recovered even if they are Gaussian and irrespective of them being independent.

Under the second core assumption, only lower-order (two or less) moments are used to find the signals, but they are considered also over time, thus:

- The nonstationarity assumption implies that the sources are stochastic signals, whose statistical properties vary with time.
- The nonwhitness assumption implies that the sources are stochastic signals, such that for example current signal samples are related to future signal samples, and this relationship can be described statistically by second-order moments such as the autocorrelation.

Under the third core assumption, not only the signals should be positive, but also the mixing matrix.

This tutorial concentrates on methods that assume independence and nongaussianity. For our discussion here, besides the implicit linearity assumption on the way the independent sources are combined to generate the observed signals, we make additional assumptions:

- We assume that the number of microphones should be equal to the number of sources.
- We assume that the speech signal reaches the microphones directly, with negligible delay and there are no reflections. The assumption about reflections holds when the party takes place in open and it does not hold when the party takes place inside, like for example in a ballroom. In the latter case, the sound of the same voice reaches the microphones several times with delays due to reflections at the walls. Here we ignore potential reflections as well as the delay on the direct way from the speaker to the microphone and discuss instantaneous methods as opposed to convolutive methods.
- We assume that the microphones are ideal and there is no band playing music, no birds singing, just people speaking. In other words, we discuss only the noise-free case.

Furthermore we will often make without loss of generality the assumption that the sources have unit variance and zero mean.

We will start by introducing the ICA in a maximum-likelihood approach and continue showing how this is related to other standard ICA methods requiring maximization of nongaussianity and mutual information. In the end we will shortly discuss additional ICA methods like cumulant-tensor methods and nonlinear decorrelation. We will also briefly review second-order BSS methods and hint at their relationship to ICA. We will also discuss the choice of ICA method for various application setups and conclude also with a short discussion on practical problems that appear when we relax some of our assumptions. We discuss thus the influence of noise as well as solutions for convolutive mixtures, for the case when the number of microphones is less than the number of speakers and for the case when the sources have rotational invariant distributions.

II. The Maximum-Likelihood Approach to ICA

In vector-matrix notation, the microphone signals are given by

$$\mathbf{x} = \mathbf{A}\mathbf{s},$$

with $\mathbf{A}$ a nonsingular matrix, while the estimated sources are computed as:

$$\mathbf{y} = \mathbf{B}\mathbf{x}. \tag{2}$$

Thus, we can state our purpose as to estimate the nonsingular square matrix $\mathbf{B}$ while making use only of $\mathbf{x}$, under the assumption that the components of $\mathbf{y}$ are independent and not Gaussian.

a) ICA does not work for Gaussian sources: The Gaussian density is fully specified by moments up to the second order. Thus for such sources decorrelation is equivalent to independence. Decorrelated sources can be computed with the help of the Principal Component Analysis (PCA). The PCA transform, which computes the decorrelated sources (i.e., such that in the transformed space the covariance matrix is diagonal) is orthogonal and thus equivalent to a rotation of the axis.

Conversely, scaled independent components are still independent. Thus we can also apply scaling factors such that the independent sources have unitary variance without loss of generality. In this case the covariance matrix of the sources is the identity matrix and the transform that achieves this is called whitening. The issue is that after computing the whitened independent Gaussian sources, any further rotation of the axis leads to a new set of independent sources with the same covariance matrix.

Thus the ICA problem is ill posed for Gaussian sources, as for a set of observed mixed signals we can find an infinite number of independent sources.

b) The likelihood objective function: It is well known that for an invertible and differentiable transform $\mathbf{a} = T(\mathbf{b})$, the density of the random variable defined over the transformed space is related to the density of the input random variable as:

$$p_{\boldsymbol{a}}(\mathbf{a}) = \frac{1}{\left|\left|T'(\boldsymbol{b})\right|\right|} p_{\boldsymbol{b}}(\mathbf{b}),$$

with $\boldsymbol{b} = T^{-1}(\boldsymbol{a})$ and $|T'(\boldsymbol{b})|$ the determinant of the Jacobian matrix of the transform. If the transform is linear such that $\mathbf{a} = \mathbf{M}\mathbf{b}$, this simplifies to

$$p_{\boldsymbol{a}}(\mathbf{a}) = \frac{1}{\left|\left|\mathbf{M}\right|\right|} p_{\boldsymbol{b}}(\mathbf{b}), \tag{3}$$

with $|\mathbf{M}|$ the determinant of the matrix $\mathbf{M}$.

Therefore, the density of the vector of observed microphone signals can be written as[1]

$$\begin{aligned} p_{\boldsymbol{x}}(\mathbf{x}) &= \big|\big|\mathbf{B}\big|\big| p_{\boldsymbol{y}}(\mathbf{y}) \\ &= \big|\big|\mathbf{B}\big|\big| \prod_i p_i(s_i) \\ &= \big|\big|\mathbf{B}\big|\big| \prod_i p_i(\mathbf{b}_i^T \mathbf{x}) \end{aligned} \tag{4}$$

with $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)^T$ for n independent sources. Assuming that we have at our disposal a sample of $\mathcal{T}$ i.i.d. realizations of the random variable $\mathbf{x}$ denoted as $S = \{\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(\mathcal{T})\}$, the likelihood of this sample under the density in Equation (4) is:

$$L(\mathbf{B}) = \prod_{t=1}^{\mathcal{T}} \big|\big|\mathbf{B}\big|\big| \prod_{i=1}^{n} p_i(\mathbf{b}_i^T \mathbf{x}(t)).$$

The log-likelihood can be computed then as:

$$\log L(\mathbf{B}) = \mathcal{T} \log \big|\big|\mathbf{B}\big|\big| + \sum_{t=1}^{\mathcal{T}} \sum_{i=1}^{n} \log p_i(\mathbf{b}_i^T \mathbf{x}(t)).$$

By multiplying the second term in the above sum with $\frac{\mathcal{T}}{\mathcal{T}}$, we get the following formula for the objective function that should be maximized over $\mathbf{B}$:

$$\begin{aligned} O(\mathbf{B}) &= \frac{1}{\mathcal{T}} \log L(\mathbf{B}) \\ &= \log \big|\big|\mathbf{B}\big|\big| + E\left\{ \sum_{i=1}^{n} \log p_i(\mathbf{b}_i^T \mathbf{x}(t)) \right\} \end{aligned} \tag{5}$$

where by the expectation operator we actually denote its' average-based estimate.

A. The gradient-ascent solution

What remains to be done now is just to maximize the likelihood objective function in Equation (5) over the parameters of the ICA transform $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)^T$. However, it should be noticed that we do not know the density $p_{\boldsymbol{y}}(\mathbf{y})$. The straight-forward solution of this optimization problem is given by the gradient-ascent approach. The optimal demixing matrix $\mathbf{B}$ can then be computed iteratively from

$$\Delta\mathbf{B}(k) = \mathbf{B}(k+1) - \mathbf{B}(k) = \eta(k) \frac{\partial O(\mathbf{B})}{\partial \mathbf{B}}$$

as

$$\mathbf{B}(k+1) = \eta(k) \frac{\partial O(\mathbf{B})}{\partial \mathbf{B}} + \mathbf{B}(k)$$

with k the iteration index and η the positive learning rate.

The gradient-ascent approach implies computing the matrix gradient of the two terms of the sum on right-hand side in Equation (5).

1) Vector and matrix gradients: We are interested here mainly in gradients of scalar-valued functions g. Then the first order vector gradient[2] of such a function after a vector $\mathbf{v}$ is defined as the vector of partial derivatives, after each vector component[3]:

$$\frac{\partial g}{\partial \mathbf{v}} = \begin{bmatrix} \frac{\partial g}{\partial v_1} \\ \vdots \\ \frac{\partial g}{\partial v_n} \end{bmatrix}$$

Thus, the vector gradient of a scalar product[4] of two vectors is

$$\frac{\partial \mathbf{a}^T \mathbf{v}}{\partial \mathbf{v}} = \mathbf{a} = \frac{\partial \mathbf{v}^T \mathbf{a}}{\partial \mathbf{v}}, \tag{6}$$

considering that a scalar is his own transpose.

The first-order matrix gradient of g after a $m \times n$ matrix $\mathbf{V}$, is a matrix of partial derivatives:

$$\frac{\partial g}{\partial \mathbf{V}} = \begin{bmatrix} \frac{\partial g}{\partial v_{11}} & \cdots & \frac{\partial g}{\partial v_{1n}} \\ \vdots & & \vdots \\ \frac{\partial g}{\partial v_{m1}} & \cdots & \frac{\partial g}{\partial v_{mn}} \end{bmatrix}$$

A matrix $\mathbf{V}$ can be also written as $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_m)^T$, using only its rows. Then, the matrix derivative can be written with the help of the vector derivative as:

$$\frac{\partial g}{\partial \mathbf{V}} = \frac{\partial g}{\partial \mathbf{v}_1, \ldots, \partial \mathbf{v}_m} = \begin{bmatrix} \left(\frac{\partial g}{\partial \mathbf{v}_1}\right)^T \\ \vdots \\ \left(\frac{\partial g}{\partial \mathbf{v}_m}\right)^T \end{bmatrix} \tag{7}$$

a) Matrix gradient of the determinant: The determinant $|\cdot|$ of a matrix $\mathbf{V}$ is a scalar function of matrix elements. Its' matrix derivative is computed as (see [2] pp. 61):

$$\frac{\partial |\mathbf{V}|}{\partial \mathbf{V}} = \left(\mathbf{V}^T\right)^{-1} |\mathbf{V}| \tag{8}$$

b) Matrix gradient of a sum of functions of matrix lines: The derivative of a sum is a sum of derivatives after each individual sum component. A vector derivative of a sum is a vector of sum derivatives after each vector component. When the sum is built over functions of vector components, then the vector derivative of this sum is a vector of derivatives of functions of one sum component after this component. Thus for the vector $\mathbf{v} = [v_1, \ldots, v_m]^T$ we can write

$$\frac{\partial \sum_i f_i(v_i)}{\partial \mathbf{v}} = \begin{bmatrix} \frac{\partial f_1(v_1)}{\partial v_1} \\ \vdots \\ \frac{\partial f_m(v_m)}{\partial v_m} \end{bmatrix}$$

The matrix derivative of a sum of functions of matrix lines can then be computed using Equation (7) as:

$$\frac{\partial \sum_i f_i(\mathbf{v}_i)}{\partial \mathbf{V}} = \begin{bmatrix} \left(\frac{\partial f_1(\mathbf{v}_1)}{\partial \mathbf{v}_1}\right)^T \\ \vdots \\ \left(\frac{\partial f_m(\mathbf{v}_m)}{\partial \mathbf{v}_m}\right)^T \end{bmatrix} \tag{9}$$

[1] In our setup we ideally have that $\mathbf{B} = \mathbf{A}^{-1}$ and thus $\mathbf{y} \equiv \mathbf{s}$.

[2] The second-order gradient is given by the Hessian matrix: $H = \frac{\partial g}{\partial \mathbf{v}} \frac{\partial g^T}{\partial \mathbf{v}}$.

[3] The generalization of this vector gradient to a vector-valued function is the Jacobi matrix.

[4] The vector gradient of the quadratic product $g(\mathbf{v}) = \mathbf{v}^T \mathbf{A} \mathbf{v}$ is: $\frac{\partial \mathbf{v}^T \mathbf{A} \mathbf{v}}{\partial \mathbf{v}} = \mathbf{A}\mathbf{v} + \mathbf{A}^T \mathbf{v}$.

2) The derivative of the objective function: The first term of the objective function in Equation (5) is the logarithm of the absolute value of the determinant[5]. However, as the logarithm is defined only over positive values, the absolute value can be ignored. By the chain rule of differentiation we then have that

$$\begin{aligned}\frac{\partial \log \left||\mathbf{B}|\right|}{\partial \mathbf{B}} &= \frac{1}{|\mathbf{B}|}\frac{\partial |\mathbf{B}|}{\partial \mathbf{B}} \\ &= \left(\mathbf{B}^T\right)^{-1}\end{aligned}$$

using Equation (8).

The second term is the expectation of a sum of functions of the lines of the matrix $\mathbf{B}$. By the sum rule of differentiation, the expectation of the gradient is the gradient of the expectation. We have therefore:

$$\frac{\partial E\left\{\sum_{i=1}^{n}\log p_i(\mathbf{b}_i^T\mathbf{x}(t))\right\}}{\partial \mathbf{B}} = E\left\{\frac{\partial \sum_{i=1}^{n}\log p_i(\mathbf{b}_i^T\mathbf{x}(t))}{\partial \mathbf{B}}\right\}.$$

We need now to compute the matrix derivative of a sum of functions of matrix lines. According to Equation (9) we have then:

$$\frac{\partial \sum_{i=1}^{n}\log p_i(\mathbf{b}_i^T\mathbf{x}(t))}{\partial \mathbf{B}} = \begin{bmatrix} \left(\frac{\partial \log p_1(\mathbf{b}_1^T\mathbf{x})}{\partial \mathbf{b}_1}\right)^T \\ \vdots \\ \left(\frac{\partial \log p_n(\mathbf{b}_n^T\mathbf{x})}{\partial \mathbf{b}_n}\right)^T \end{bmatrix}$$

Using again the chain rule of differentiation we have for one entry in the above vector:

$$\begin{aligned}\frac{\partial \log p_i(\mathbf{b}_i^T\mathbf{x})}{\partial \mathbf{b}_i} &= \frac{1}{p_i(\mathbf{b}_i^T\mathbf{x})}\frac{\partial p_i(\mathbf{b}_i^T\mathbf{x})}{\partial \mathbf{b}_i^T\mathbf{x}}\frac{\partial \mathbf{b}_i^T\mathbf{x}}{\partial \mathbf{b}_i} \\ &= \frac{p_i^{'}(\mathbf{b}_i^T\mathbf{x})}{p_i(\mathbf{b}_i^T\mathbf{x})}\mathbf{x}\end{aligned}$$

using Equation (6) and with $p_i^{'}(\mathbf{b}_i^T\mathbf{x}) = \frac{\partial p_i(\mathbf{b}_i^T\mathbf{x})}{\partial \mathbf{b}_i^T\mathbf{x}}$. Denoting

$$g_i = \frac{p_i^{'}}{p_i} \tag{10}$$

and taking into consideration that $g_i(\mathbf{b}_i^T\mathbf{x}))$ is a scalar function of a scalar argument, we obtain

$$\begin{aligned}\frac{\partial \sum_{i=1}^{n}\log p_i(\mathbf{b}_i^T\mathbf{x}(t))}{\partial \mathbf{B}} &= \begin{bmatrix} g_1(\mathbf{b}_1^T\mathbf{x})\mathbf{x}^T \\ \vdots \\ g_n(\mathbf{b}_n^T\mathbf{x})\mathbf{x}^T \end{bmatrix} \\ &= \mathbf{g}(\mathbf{B}\mathbf{x})\mathbf{x}^T\end{aligned}$$

Therefore we may finally write the derivative of the objective function as:

$$\frac{\partial O(\mathbf{B})}{\partial \mathbf{B}} = \left(\mathbf{B}^T\right)^{-1} + E\left\{\mathbf{g}(\mathbf{B}\mathbf{x})\mathbf{x}^T\right\}.$$

[5]The determinant is defined only for square matrices, in our case the matrix is also nonsingular as it has an inverse.

Thus we should use the following iteration for finding $\mathbf{B}$:

$$\Delta\mathbf{B}(k) = \eta(k)\left[\left(\mathbf{B}^T\right)^{-1} + E\left\{\mathbf{g}(\mathbf{B}\mathbf{x})\mathbf{x}^T\right\}\right] \tag{11}$$

In practice, the stochastic gradient is used, such that the training sample S is used in several training epochs k until the fulfillment of the stopping condition $\Delta\mathbf{B}^k < \lambda$, where λ a small positive constant. The stochastic gradient allows us to eliminate the expectation operator in Equation (11), obtaining thus the iteration

$$\Delta\mathbf{B}^k(t) = \eta^k(t)\left[\left(\mathbf{B}^T\right)^{-1} + \mathbf{g}(\mathbf{B}\mathbf{x}(\mathbf{t}))\mathbf{x}(\mathbf{t})^T\right],$$

with $t = 1, \ldots, T$.

B. The natural gradient

It can be shown [1], [2][6] that the parameter space for our optimization consists of all nonsingular $n \times n$ matrices $\mathbf{B}$. This space is a manifold in the space of all $n \times n$ matrices. The gradient rule (11) follows the direction of the steepest descent in the space of all matrices, which is not necessarily the direction of steepest descent in the space of nonsingular matrices. Therefore, the gradient update rule should be modified to take into consideration the metric specific to this manifold. As this manifold has a Riemannian structure, meaning that the corresponding metric tensor is positive definite, a distance function (i.e., metric) can indeed be defined[7]. It turns out that the direction of steepest descent with respect to the metric of the parameter space $\mathbf{B}$ is computed with the help of the natural gradient defined as:

$$\frac{\partial^{nat}}{\partial \mathbf{B}} = \frac{\partial}{\partial \mathbf{B}}\mathbf{B}^T\mathbf{B}$$

instead of the usual gradient (see [1]).

Thus, the iteration in Equation (11) becomes:

$$\begin{aligned}\Delta\mathbf{B}(k) &= \eta(k)\left[\left(\mathbf{B}^T\right)^{-1} + E\left\{\mathbf{g}(\mathbf{B}\mathbf{x})\mathbf{x}^T\right\}\right]\mathbf{B}^T\mathbf{B} \\ &= \eta(k)\left[\mathbf{I} + E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\}\right]\mathbf{B}\end{aligned} \tag{12}$$

with $\mathbf{I}$ the identity matrix and using Equation (2).

C. Practical choice of nonlinearities

The iteration in Equation (12), or its' equivalent stochastic version, depends on the nonlinear set of functions $\mathbf{g}(\mathbf{y}) = (g_1(y_1), \ldots, g_n(y_n))$. Ideally each of this function is computed using the density of the corresponding independent component as described in Equation (10). However these densities are not known and in practice they are replaced by approximations (see [2] pp. 204 ff.). Theoretically the number of approximative component-density models to choose from is infinite. The problem becomes simpler if we can assume that the independent components are all part of a family of distributions that either has few (but enough) members or whose individual members can be specified by a small number of parameters. This assumption is usually made in practice.

[6]See [2] in particular for a nice and simple explanation

[7]The metric is used to compute the geodesic of any two points on the manifold. The geodesic is the shortest path within the manifold between the two respective points.

Several families of distributions are available, depending on the number of used parameters and on the symmetry[8] of the models they generate. Of course these families must be rich-enough for our modeling purposes.

Next we discuss some of the most often used families suited for mono-modal densities both symmetric and skewed. The multi-modal case is not discussed here, it could be handled by estimating the densities as Gaussian mixture models but at a high computational cost.

1) Solutions for symmetric distributions: It turns out that for ICA, when working with symmetric (i.e., non-skewed or zero-skewed) densities, we need at least two members per family: one member that is well suited as a model for leptokurtic (i.e., positive kurtosis, also known as super-Gaussian) distributions and the other one for platikurtic (i.e., negative kurtosis, also known as sub-Gaussian) distributions. Furthermore speech signals usually have symmetric distributions so this case is particularly important in the context of BSS.

a) Constructing families of two member distributions: The question to be asked in this case is what are the core properties of families of distributions suited for models in ICA? The answer is that these have to have members corresponding to both platikurtic and leptokurtic distributions. Thus in the most simple case, the respective family must have at least two members. One possible solution to construct such a family is to simply pick known parametric distributions that are lepto- and respectively platikurtic and pair them. An alternative is to construct such distributions from scratch. These have to be valid densities (i.e, positive functions that integrate to one) and their nonpolynomial moments have to have opposing signs. The nonpolynomial moment (see [2] pp. 187) is defined with the help of the nonlinearities g_i from Equation (10) as

$$m_{np} = E\left\{y_i g_i(y_i) - g_i'(y_i)\right\}$$

and it has to be positive for the ML estimate to be consistent (i.e., to converge to the true value). For $g_i(y) = -y^3$ the nonpolynomial moment is the same as the kurtosis and indeed it has the same sign as the kurtosis. However this nonlinearity does not correspond to a density. This can be easily shown by computing the solution of the corresponding differential equation (derived from Equation (10)) that involves it. The solution is $p_i(y) = e^{y^4/4}$ and it can not be a density as it is not integrable.

A symmetric density often used as supergaussian member of a family of two distributions is the hyperbolic cosine distribution defined as:

$$p(y) = \frac{1}{\pi \cosh(y/\sigma^2)}$$

Its corresponding ICA nonlinearity is:

$$g_i(y_i) = -\tanh(y_i/\sigma_{y_i}^2).$$

The corresponding subgaussian component of the family, i.e., the density for which the nonpolynomial moment has the same mathematical expression but with reversed sign is:

$$p(y) = \frac{1}{e^{y^2}\cosh(y)}$$

Its corresponding ICA nonlinearity is:

$$g_i(y_i) = \tanh(y_i) - y_i$$

Thus practically our task is simplified to a choice between two possibilities, based in the end on knowledge about the sign of the kurtosis of a component distribution.

b) The generalized Gaussian distribution: Yet another often used symmetric density family is that generated by the generalized Gaussian distribution. The generalized Gaussian distribution is computed as:

$$q(y) = \frac{r}{2\sigma\Gamma(1/r)} \exp\left(-\frac{1}{r}\left|\frac{y}{\sigma}\right|^r\right)$$

with $r > 0$ and $\Gamma(r) = \int_0^\infty y^{r-1}\exp(-y)dy$ the gamma function. For leptokurtic distributions r is small (e.g., $r = 1$ corresponds to the Laplacian distribution) and for platikurtic distributions, r is larger (e.g., for the Gaussian distribution, $r = 2$). Assuming without loss of generality that $\sigma = 1$, the corresponding nonlinear function is

$$\begin{aligned} g_i(y_i) &= \frac{q_i(y_i)'}{q_i(y_i)} \\ &= \frac{q_i(y_i)\cdot -\frac{1}{|\sigma_i|^r}|y_i|^{r-1}\mathrm{sign}(y_i)}{q_i(y_i)} \\ &= -|y_i|^{r_i-1}\mathrm{sign}(y_i), \end{aligned}$$

which, considering that $\mathrm{sign}(y) = \frac{y}{|y|}$, translates in practice into

$$g_i(y_i) = -\frac{y_i}{(|y_i|^{2-r_i}+\epsilon)},$$

with $\epsilon = 10^{-4}$, to avoid the function singularity at $y_i = 0$.

Typically, $r = 1$ for leptokurtic distributions and $r = 4$ for platikurtic distributions. The parameter r can be expressed as a function of the kurtosis (see [1] pp. 249 ff.), which gives us a way to set it automatically, provided we know the kurtosis.

c) Investigating the kurtosis of the component distributions: The choice of the optimal nonlinearity for symmetric component distributions depends on the sign of the kurtosis k defined using the fourth and second moments as $k = m_4 - 3m_2^2$. As the component distributions are not known and thus we can not compute the moments directly, we have to resort to various approximations.

Thus, if information about the sign of the kurtosis is considered enough, then this can be obtained from the "non-polynomial moment" defined above.

Conversely, the moments can be iteratively estimated online. The j'th order moment m_j of a random variable y can be computed starting with $m_j(0) = 0$ as

$$m_j(k) = (1-\eta_0)m_j(k-1) + \eta_0|y(k)|^j \qquad (13)$$

with $\eta_0 = 0.01$ and k an iteration index running also over the available sample $S_y = \{y(1), \ldots, y(N)\}$ of the random variable.

[8]Perfect symmetry means zero skew.

2) Solutions for non-symmetric distributions: Families of symmetric distributions are ill suited to model independent components with skewed (or asymmetric) distributions. For such cases, the Pearson model can be used (see [1] pp. 253). The Pearson model is described by the differential equation:

$$q^{'}(x) = \frac{(x-a)q(x)}{b_0 + b_1 x + b_2 x^2}$$

with parameters a, b_0, b_1 and b_2. The corresponding nonlinear function is:

$$g_i(y_i) = \frac{y_i - a}{b_0 + b_1 y_i + b_2 y_i^2}$$

The Pearson model accommodates both symmetric and non-symmetric distributions. However, for symmetric distributions it needs more parameters than the two models discussed above. Therefore, its true strength lies in its ability to model nonsymmetric distributions. The parameters can be estimated directly with the help of moments up to the fourth order that can be estimated online as previously discussed (see Equation (13)).

D. Conclusion

The ML-based approach to ICA using the natural gradient forms the backbone of this contribution. In the case this is used to solve the cocktail-party problem, it is often assumed that speech has a symmetric mono-modal super-gaussian distribution which is approximated either with the generalized Gaussian distribution with $r = 1$ or with the Laplacian distribution. Nevertheless, the approach is more general than this. For ICs with symmetric mono modal distributions of various kurtosis – which seems to be the case most often encountered in practice – the generalized Gaussian distribution is used as described above. We show in Figure 2 a flow diagram describing the more general algorithm. As this is a gradient ascent algorithm, we initialize $\eta = 1$. $\mathbf{B}$ is initialized randomly.

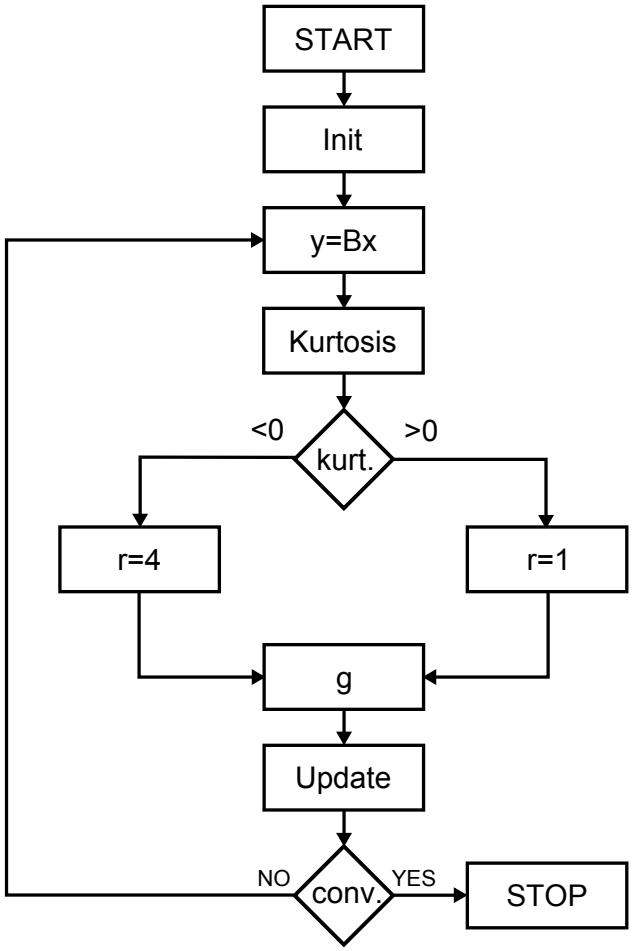

Fig. 2. Flow diagram of the ML-based ICA method.

III. From Maximum-Likelihood over Mutual Information to Nongaussianity

In the previous chapter we have introduced the ICA in a maximum-likelihood approach. As it turns out, the ICA can be introduced from several other different perspectives. However, many of these other perspectives are equivalent to the maximum likelihood approach in the sense that they yield the same or very similar objective functions as the one in Equation (5).

A. The Kullback-Leibler divergence and the mutual information

By the definition of independence, a set of random variables are considered independent if their joint density can be factorized (see also Equation (1)) as

$$p(\mathbf{y}) = \prod_{i=1}^{n} p_i(y_i).$$

We discuss next two ways to measure independence that can be used to perform ICA: the Kullback-Leibler divergence and the mutual information. We show then that they are equivalent to one another and to the ML formulation from before.

a) The Kullback-Leibler divergence: The Kullback-Leibler divergence can be used to measure how similar two distributions are, which makes it in our case a natural measure[9] of independence, when measuring how similar the two sides of the above equation are. The Kullback-Leibler divergence computed with respect to the vector $\mathbf{y}$ is then

$$KLd(\mathbf{y}) = \int_{(n)} p(\mathbf{y}) \log \frac{p(\mathbf{y})}{\prod_{i=1}^{n} p_i(y_i)} d\mathbf{y}$$

where $\int_{(n)} = \int \cdots \int_{n}$ is a multiple integral of order n.

The Kullback-Leibler divergence is always positive, being zero only if the two distributions are identical.

b) The mutual information: For two random variables, the mutual information is a measure of how much information one random variable carries over the other one. This can be easily extended to a set of random variables. The mutual information is then a measure of how much information does a random variable or a subset of random variables carry over the other random variables in the set.

[9]The Kullback-Leibler divergence is not a measure in the mathematical sense because it does not fulfill all axioms of a measure: it is not symmetric.

For a realization of a discrete random variable, information is defined as the negative of the logarithm in base two of the probability of that realization. The mean information over all possible realizations is called entropy and is used to describe the information content of a random variable. For continuous random variables, the equivalent description is given by the differential entropy defined as:

$$H(y) = -\int p(y)\log p(y)dy \tag{14}$$

The mutual information in a set $\mathbf{y} = \{y_1, \ldots, y_n\}$ of n continuous random variables is defined with the help of the differential entropy as:

$$I(y_1, \ldots, y_n) = \sum_{i=1}^{n} H_i(y_i) - H(\mathbf{y}) \tag{15}$$

c) Equivalence between the Kullback-Leibler divergence and the mutual information: Making use of Equation (14), we can write the mutual information as:

$$\begin{aligned} I(\mathbf{y}) &= -\sum_{i=1}^{n}\int p_i(y_i)\log p_i(y_i)dy_i - H(\mathbf{y}) \\ &= -\sum_{i=1}^{n}\int \log p_i(y_i)\left(\int_{(n-1)} p(\mathbf{y})d\hat{\mathbf{y}}\right)dy_i - H(\mathbf{y}) \\ &= -\int_{(n)}\sum_{i=1}^{n} p(\mathbf{y})\log p_i(y_i)d\mathbf{y} - H(\mathbf{y}) \\ &= \int_{(n)} p(\mathbf{y})\log\frac{1}{\prod_{i=1}^{n} p_i(y_i)}d\mathbf{y} + \int_{(n)} p(\mathbf{y})\log p(\mathbf{y})d\mathbf{y} \\ &= \int_{(n)} p(\mathbf{y})\log\frac{p(\mathbf{y})}{\prod_{i=1}^{n} p_i(y_i)}d\mathbf{y} \\ &= KLd(\mathbf{y}) \end{aligned}$$

with $d\hat{\mathbf{y}} = dy_1, \ldots, dy_{i-1}, dy_{i+1}, \ldots, dy_n$. In the expansion above we have also used the fact that by Fubini's theorem $\int \log p_i(y_i)\left(\int_{(n-1)} p(\mathbf{y})d\hat{\mathbf{y}}\right)dy_i = \int_{(n)} \log p_i(y_i)p(\mathbf{y})d\mathbf{y}$ in the third leg of the expansion, the sum and integral can be interchanged because we have there the expectation of a sum and by the linearity of the expectation operator, the expectation of a sum is a sum of expectations.

Alternatively, using the same tricks as above we have that:

$$\begin{aligned} KLd(\mathbf{y}) &= \int_{(n)} p(\mathbf{y})\left(\log p(\mathbf{y}) - \log\prod_i p_i(y_i)\right)d\mathbf{y} \\ &= \int_{(n)} p(\mathbf{y})\log p(\mathbf{y})d\mathbf{y} - \int_{(n)} p(\mathbf{y})\log\prod_i p_i(y_i)d\mathbf{y} \\ &= -H(\mathbf{y}) + \sum_i H_i(y_i) \\ &= I(\mathbf{y}) \end{aligned}$$

Thus the mutual information is the same thing as the Kullback-Leibler divergence between the joint density of a vector and the product of individual densities. The mutual information is thus always positive being zero only if the respective variables are independent.

d) ICA objective functions: The mutual information (or equivalently the Kullback-Leibler divergence) can be used as objective function for ICA. Considering Equation (14) it is clear that:

$$H_i(y_i) = -E\left\{\log p_i(y_i)\right\}.$$

Therefore, with $\mathbf{y} = \mathbf{B}\mathbf{x}$ and thus $y_i = \mathbf{b}_i^T\mathbf{x}$, the mutual information-based objective function for ICA is

$$O(\mathbf{B}) = I(\mathbf{B}\mathbf{x}) = -H(\mathbf{B}\mathbf{x}) - \sum_{i=1}^{n} E\left\{\log p_i(\mathbf{b}_i^T\mathbf{x})\right\}.$$

To find the demixing matrix $\mathbf{B}$ we have to minimize the mutual information.

1) Relation to maximum-likelihood: Using equation (3) it can be easily shown the differential entropy of a transformed random variable $\mathbf{a} = \mathbf{M}\mathbf{b}$ is (see [2] pp. 109)

$$H(\mathbf{a}) = H(\mathbf{b}) + \log\big||M|\big|, \tag{16}$$

we have that the mutual information-based objective function

$$O(\mathbf{B}) = -H(\mathbf{x}) - \log\big||B|\big| - \sum_{i=1}^{n} E\left\{\log p_i(\mathbf{b}_i^T\mathbf{x})\right\}, \tag{17}$$

Comparing Equation (5) with Equation (17) and taking into consideration that a sum of expectations is the expectation of a sum, we may see that the the mutual-information-based objective function is the same as the negative of the maximum-likelihood objective function, save for a constant term $-H(\mathbf{x})$, which is independent of $\mathbf{B}$. Thus, finding the maximum of the likelihood function is equivalent to finding the minimum of the mutual information (or equivalently minimizing the corresponding Kullback-Leibler divergence). Paying attention to the minus sign, the iterative procedure from Section II can be then applied directly.

2) Link to non-Gaussianity: Negentropy (see [2] pp. 112) is a normalized version of the entropy defined as

$$J(\mathbf{y}) = H^{gauss}(\mathbf{y}) - H(\mathbf{y}), \tag{18}$$

with $\mathbf{y}_{gauss}$ a Gaussian random vector of the same covariance/correlation as $\mathbf{y}$. The negentropy is nonnegative, being zero only for Gaussian random vectors. This happens because of all random variables of equal variance, a Gaussian random variable has the largest entropy. For independent unit-variance random variables $y_i, \forall i \in \{1, 2, \ldots, n\}$, using equations (15) and (16) the mutual information can be written using the negentropy of each independent component as (see [2] pp. 223)

$$\begin{aligned} I(\mathbf{y}) &= \sum_i \left(H^{gauss}(y_i) - J_i(y_i)\right) - H(\mathbf{x}) - \log\big||B|\big| \\ &= const. - \sum_i J(y_i), \end{aligned}$$

where the constant term does not depend on the independent components, as the entropy of the independent components under the Gaussian assumption is constant. Thus to minimize the mutual information, we have to maximize the negentropy, which is equivalent to looking for y_i such that each one is as different from Gaussian as possible.

As it may be seen, conducting ICA over nongaussianity also hints to a change of setup, where the independent components are computed one by one or sequentially as opposed to the block methods from above. We concentrate next on such sequential methods.

B. Sequential, nongaussianity-based methods

As discussed above, the ICA can be conducted by enforcing nongaussianity. The same result can also be derived from the central limit theorem that says that the distribution of a sum of independent identically distributed random variables of finite mean and variance tends to be Gaussian (see [2] pp. 166). Thus, single independent nonGaussian components are less Gaussian than their linear combinations, and it makes sense to look for independent components along directions of maximal nongaussianity.

Nongaussianity can be measured in several ways. Besides the negentropy method hinted above, one could use higher-order moments and cumulants (which are zero for Gaussian variables) like the kurtosis for example.

As already pointed out, nongaussianity methods allow you to conduct ICA sequentially. This means that you compute the independent components one after the other, or equivalently you compute the lines (a.k.a. rows) of the demixing matrix (which are the vectors over which you project the data to make it nonguassian) one after the other. Such methods allow us also to find a variable number (less than the number of available mixtures) of independent components.

To better grasp the intuition behind such procedures it is good to consider the ICA from a slightly different perspective. This is the perspective of ICA as a further step after whitening (see [2] pp. 160). This perspective hints as well to an iterative method for conducting PCA that makes further use of nongaussianity methods.

1) ICA and whitening: For independent random variables all mixed moments starting with the correlation are zero. Thus on the path towards independence it makes intuitively sense to start by decorrelating the observed data. Without loss of generality we can also assume/impose unit variance of the independent components. Such white (i.e., uncorrelated and with unit variance) data brings also computational and conceptional advantages in the ICA, mostly related to the fact that the ICA demixing matrix of whitened data is orthogonal. Thus, the first step in many nongaussianity ICA methods is to whiten the input data (see [2] pp. 158).

Assuming whitened data, we show next that the corresponding ICA demixing matrix is orthogonal (it keeps an orthonormal basis). Denoting the whitened data as

$$\mathbf{z} = \mathbf{V}\mathbf{x}$$

the relation to the original sources is described by:

$$\mathbf{z} = \mathbf{V}\mathbf{A}\mathbf{s}.$$

We have thus a "new" mixing matrix $\tilde{\mathbf{A}} = \mathbf{V}\mathbf{A}$. This matrix is orthogonal (its' inverse is its' transpose), as it can be easily seen from[10]

$$E\left\{\mathbf{z}\mathbf{z}^T\right\} = \tilde{\mathbf{A}} E\left\{\mathbf{s}\mathbf{s}^T\right\} \tilde{\mathbf{A}}^T = \underline{\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T = \mathbf{I}},$$

where we have assumed without loss of generality that the independent sources are white.

Thus, ICA can be seen in this case as a rotation of the whitened data and the corresponding ICA algorithm consists of whitening the data and then looking for a rotation such as the nongaussianity of each component is maximized. We discuss next practical ways to measure nongaussianity.

2) Negentropy methods: ICA as an additional step towards maximal nongaussianity after whitening needs nongaussianity measures. As already pointed out above, negentropy is such a measure. Then, the negentropy-based ICA objective function is

$$O(\mathbf{B}) = \sum_i J(\mathbf{w}_i^T \mathbf{z})$$

where $\mathbf{z}$ has unit variance and is computed by whitening $\mathbf{x}$. It should be maximized over $\mathbf{W}$, which is the new orthogonal unmixing matrix of white data as already discussed. This is equivalent to maximizing a sum of positive variables, which can be done by maximizing each variable individually. However, in this case we have to pay attention to eliminating the influence of one variable after we have maximized it and to make sure that $\mathbf{W}$ is orthogonal. This procedure is called deflation and is discussed in more detail below.

Negentropy as defined in Equation (18) is difficult to compute in practice, as the density of $\mathbf{y}$ is not known. For our ICA purpose, instead of computing the negentropy directly, we approximate it with a formula that is easy to evaluate practically. A simple approximation is given by:

$$J(y) = \frac{1}{12} E\left\{y^3\right\}^2 + \frac{1}{48} k(y)^2$$

However, this approximation uses high-order moments and is not very robust. Thus, the following approximation of the negentropy for scalar random variables is used in practice (see [2] pp. 183 ff.):

$$J(y) = \left[E\{G(y)\} - E\{G(\nu)\}\right]^2$$

with ν a standard Gaussian variable ($N(0,1)$) and $G(\cdot)$ a nonquadratic function like for example $G(y) = -e^{-y/2}$. Actually, this nonlinearity is related to our assumptions on the way the densities of the independent components can be modeled. Starting here a gradient-ascent algorithm can be devised.

3) Kurtosis methods: The kurtosis is easier to compute than the negentropy, but using fourth order cumulants/moments is very sensible to outliers in the available sample S. Conversely, making use of the kurtosis implies the assumption that the independent components can be effectively approximated by symmetric monomodal distributions as these can be described very well by the kurtosis alone. As already discussed above, the kurtosis of a random variable x can be computed as:

$$k(x) = E\left\{x^4\right\} - 3E^2\left\{x^2\right\}.$$

[10] With respect to our notation above, we have that $\tilde{\mathbf{A}} \equiv \mathbf{W}^T$.

It can be also defined in relation to the fourth order cumulant $k_4(x)$ as:

$$k^c(x) = \frac{k_4(x)}{k_2^2(x)} = \frac{E\left\{x^4\right\}}{E^2\left\{x^2\right\}} - 3$$

The kurtosis is used here as a measure of nongaussianity, as for Gaussian variables the kurtosis (as well as all cumulants of an order higher than two) is zero. Therefore nongaussian variables have a kurtosis with a large absolute value. The minimal absolute kurtosis value of zero is achieved only for Gaussian variables.

To find one independent component in this setup we have to search for a projection vector over which to project $\mathbf{z}$ such that the absolute value of the kurtosis of the projection result $y = \mathbf{w}^T\mathbf{z}$ is as large as possible.

As discussed above, an iterative algorithm that finds such a $\mathbf{w}$ will make use of the gradient of the absolute value of the kurtosis with respect to the projection vector $\mathbf{w}$. The gradient of the kurtosis with respect to $\mathbf{w}$ is:

$$\begin{aligned}
\frac{\partial k(y)}{\partial \mathbf{w}} &= \frac{\partial\left(E\left\{y^4\right\} - 3E^2\left\{y^2\right\}\right)}{\partial \mathbf{w}} \\
&= \frac{\partial\left(E\left\{(\mathbf{w}^T\mathbf{z})^4\right\} - 3E^2\left\{(\mathbf{w}^T\mathbf{z})^2\right\}\right)}{\partial \mathbf{w}} \\
&= 4\left[E\left\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\right\} - 3E\left\{(\mathbf{w}^T\mathbf{z})^2\right\}E\left\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\right\}\right] \\
&= 4\left[E\left\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\right\} - 3\|\mathbf{w}\|^2\mathbf{w}\right]
\end{aligned}$$

taking into consideration that with $\mathbf{w}^T\mathbf{z}$ a scalar value, we have: $E\left\{(\mathbf{w}^T\mathbf{z})^2\right\} = E\left\{(\mathbf{w}^T\mathbf{z})(\mathbf{w}^T\mathbf{z})^T\right\} = E\left\{(\mathbf{w}^T\mathbf{z})(\mathbf{z}^T\mathbf{w})\right\} = \mathbf{w}^T E\left\{\mathbf{z}\mathbf{z}^T\right\}\mathbf{w} = \|\mathbf{w}\|^2$ and $E\left\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\right\} = E\left\{\mathbf{z}(\mathbf{z}^T\mathbf{w})\right\} = E\left\{\mathbf{z}\mathbf{z}^T\right\}\mathbf{w} = \mathbf{w}$.

The gradient of the absolute value of the kurtosis is:

$$\frac{\partial|k(y)|}{\partial \mathbf{w}} = 4\text{sign}(k(\mathbf{w}^T\mathbf{z}))\left[E\left\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\right\} - 3\|\mathbf{w}\|^2\mathbf{w}\right]$$

With the help of the above gradient we can devise a gradient ascent algorithm to maximize the absolute value of the kurtosis and find one maximum nongaussianity direction and equivalently one independent component. The gradient ascent algorithm is built around

$$\begin{aligned}
\Delta\mathbf{w} &= \frac{\partial|k(y)|}{\partial \mathbf{w}} \\
&= 4\text{sign}(k(\mathbf{w}^T\mathbf{z}))\left[E\left\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\right\} - 3\|\mathbf{w}\|^2\mathbf{w}\right] \\
&\propto \text{sign}(k(\mathbf{w}^T\mathbf{z}))\left[E\left\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\right\}\right]
\end{aligned}$$

after eliminating the terms that influence only the norm of $\mathbf{w}$ and not its direction, because due to whitening the norm is one. This is then enforced in an additional step where

$$\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|}.$$

When looking for more than one independent component this algorithm needs to be applied several times, once for each sought component. The components are estimated sequentially (i.e., one by one). Again we have to conduct deflation after estimating each single component.

4) Deflation procedures.: In the sequential setup, after we find an independent component we have to apply a deflation procedure to somehow eliminate its influence before beginning the search for another independent component, because obviously otherwise will find the same component time and again. There are several ways to apply this deflation procedure depending if the data on which they are applied is white or not.

In [2] (pp. 194), where we work with whitened data, this deflation procedure takes the form of a Gram-Schmidt orthogonalization on the lines of the demixing matrix, which ensures that the demixing matrix is orthogonal (as discussed above, the mixing matrix of whitened data is orthogonal). The procedure is applied after every iteration of the gradient descent algorithm used to find the respective single independent component.

The deflation procedure in [1] (pp. 191-192) works differently by effectively removing the found independent component from the available data, and it also does not require whitened data. However, when applied to whitened data the projection vector that allows us to eliminate the respective independent component is the same as the corresponding line in the demixing matrix.

IV. Other Linear ICA Methods

A. Cumulant-tensor methods

The covariance matrix is actually a second order tensor. Considering that for mean-free random variables the second order moment is equal to the second order cumulant, the covariance matrix is a second-order cumulant tensor. In a similar manner a fourth-order cumulant tensor can be defined (of course you would need at least four random variables). The fourth-order cumulant includes both the skew and the kurtosis.

Decorrelation can be achieved by diagonalizing the second-order cumulant tensor (i.e., the covariance matrix). In a similar manner independence (at least up to fourth-order moments) can be achieved by diagonalizing the fourth-order cumulant tensor. The eigen-decomposition of a fourth-order tensor is made of second-order tensors (i.e., eigen-matrices). It can be shown that the desired diagonalization of the fourth-order cumulant tensor can be achieved by diagonalizing its eigenmatrices. If this diagonalization is conducted iteratively (i.e, the solution of the corresponding characteristic equation is found iteratively) then the iteration is the same as the one of the FastICA (i.e., the sequential, nongaussianity-based methods with a fixed-point iteration instead of gradient ascent).

B. Nonlinear decorrelation and nonlinear PCA

The idea behind nonlinear decorrelation is to find the ICA demixing matrix such that correlations of a certain (very large) class of nonlinear functions of the observations are zero. This is related to independence because if you express these functions as Taylor series and impose the decorrelation condition over their Taylor expansions, you find out that it (usually) implies that all cross moments need to cancel. Several iterative methods to estimate the ICA demixing matrix are built on this idea (i.e., the Jutten-Hernault and Chicocki-Unbehauen algorithms). It can be shown that nonlinear decorrelation is equivalent to the ML estimation when computing the ML iteration with the help of the natural gradient.

Nonlinear PCA implies finding the sources such as to minimize the approximation error:

$$\mathcal{E} = E\left\{\mathbf{x} - \sum_{i=1}^{n} g_i(\mathbf{b}_i^T\mathbf{x})\mathbf{b}_i\right\}$$

with $g_i(v)$ some nonlinear function, which is derived from the PCA objective function where $g_i(v) = v$. It can be shown that nonlinear PCA is equivalent to the ML estimation of the sources.

C. Non-negative matrix factorization

Assuming that the independent sources are non-negative – meaning that they can't generate negative observations – and they are additively combined – meaning that the mixing matrix has only positive entries – then the observed sources will also be non-negative. Thus, if we gather all observations from the observed sources into a data matrix, to find the original sources and the mixing matrix, we have to decompose the data matrix into two non-negative matrices. This process is called non-negative matrix factorization. If the non-negativity assumptions hold and the number of observed sources is larger or equal to the number of independent sources, there are non-negative matrix factorizations algorithms that return the independent components.

V. Relaxing the original assumptions for finding a solution to the BSS problem

Next we discuss what happens if we go about relaxing the assumptions under which we have computed the independent sources. We start with the gaussianity assumption and then continue with methods to find the sources when there are less observations than sources, then find the sources in reflective environments and finally we discuss the noisy ICA case.

A. Relaxing the Gaussianity assumption: second-order BSS and relation to ICA

The ICA framework does not take into consideration eventual time dependencies between the observations from the independent sources, it assumes that the observations are independent and identically distributed (i.i.d.). However, in many cases – like for example in the case of BSS – the observations are not i.i.d., but they exhibit meaningful time relationships. In such cases, it is enough to decorrelate the two observed signals at several (ideally all) time instances in order to recover the original independent sources [6][11]. Thus it is possible to perform ICA while making use of moments only up to and including the second order, which in turn implies that in this case we can recover the independent sources even if they are Gaussian.

[11]In this paper they construct an equivalence space of signals and mixing systems that lead to the same signal at the microphones. Afterwards they demonstrate what properties should signals have such that they are not part of this equivalence class. The signals should exhibit a meaningful time structure and be decorrelated alternatively they should have different kurtosis. This shows why ICA methods using kurtosis are good for BSS, it does not however show that complete independence is achieved. Thus it may be that other independence measures that do not overrely on the kurtosis are better for independence.

1) BSS using time-lagged covariances: The algorithm used in this case implies defining a time-lagged covariance matrix that includes all correlations between the sources and within each source at two different time instances and then diagonalizing both the instantaneous covariance matrix and the time-lagged one.

Should some lagged covariances be equal, such that the eigen-decomposition leads to equal eigenvalues, the algorithm does not work, as the corresponding ICs are not uniquely defined[12]. The solution includes using several time-lags and hoping that you can find at least one where all lagged covariances are different.

It is possible that even this will not work, for example in the case when the sources are wide-sense stationary signals and their autocorrelation/autocovariance functions are equal (the signals have the same power spectral density). Clearly the eigenvalues of the lagged covariance matrix of the observations are the autocovariances of the sources at the respective time lag/index. Should these be equal at any time index, any lagged covariance matrix of the observations will have equal (but potentially different for different lags) eigenvalues. In this case you must resort to the standard ICA to find the independent components, which as already discussed only works for non-Gaussian sources.

2) BSS for nonstationary signals: The methods of the previous section can be applied to both stationary and non-stationary signals. However, if the signals are nonstationary, we have a chance at finding the sources, even if the time-lagged covariances are equal all the time (i.e., irrespective of the lag and of the start point in time), like for example as in the case of a purely random non-stationary signal. To do so we need to find the transform that decorrelates the observed signals at all time indices simultaneously. The corresponding objective function is built around a measure of the diagonality of a positive semidefinite matrix (as all covariance matrices are symmetric and positive semidefinite). For a matrix $\mathbf{M}$ with elements m_{ij} this measure is:

$$\mathcal{M}(\mathbf{M}) = \sum m_{ii} - \log\left|\left|\mathbf{M}\right|\right|$$

and is zero only for diagonal matrices, being otherwise positive. In this case we assume that the covariances vary slowly and linearly in time.

There is also an alternative that uses cross-cumulants.

B. BSS for more sources than mixtures

Here, we use the fact that the sources are sparse in time, i.e., we would rarely have the case that all sources are active at the same time.

C. Relaxing the no-reflections assumption: BSS for convolutive mixtures

In this case we differentiate between time-methods and frequency methods. Time methods describe the convolution

[12]The eigenvectors corresponding to equal eigenvalues span an eigenspace. Any vector in this eigenspace – i.e., any linear combination of the eigenvectors corresponding to the equal eigenvalues – is again an eigenvector with the same eigenvalue.

explicitly and put it into matrix-vector notation (similar to the blind deconvolution case), then solve the standard ICA. The problem in this case is that the computational complexity explodes even by a limited number of sources.

The alternative is to apply the short time Fourier transform (STFT) to the signal (i.e., apply the Fourier transform block wise – meaning in non-overlapping windows – to the signal), then for each frequency bin in a (discrete) STFT time-block we have the instantaneous ICA setup. Each block gives us one observation from each frequency bin and thus we can apply any of the standard ICA methods on this data. Thus the convolutive ICA is a set of instantaneous ICAs. The number of components of this set equals the number of frequency bins to be found into one STFT time-block.

The two uncertainties of the ICA: the scaling and the permutation represent major problems in this setup. The scaling uncertainty can be solved efficiently by rescaling the estimated sources such that their mix by the inverse of the unmixing matrix yields observations with the same energy as the original observations. In other words we do not find the original sources, but the dampened sources as they reach the microphones. We say we solve the scaling problem by the minimal distortion principle ([5]). The permutation uncertainty is solved such that the frequency structure of each source can not vary abruptly. One approach is to ensure that the envelops of each independent signal are maximally correlated. Alternatively we can solve the permutation issue by comparing the form of the approximated probability densities in each frequency bin. This should be slowly varying with the frequency in each independent source and different between sources at most frequencies.

D. Relaxing the noise-free assumption: noisy ICA

If the signals have some time structure, then apply adaptive-filter based inference cancellation to the observed signal, followed by any BSS method to obtain the sources and then apply again the adaptive-filter based procedure to the recovered independent sources.

Otherwise use MAP estimation with a prior defined such as to add a constraint to the objective function defined as the sum of squared differences between the mixed recovered sources and the corresponding observations (see [2] pp. 299).

E. Relaxing the linearity assumption: nonlinear ICA

Nonlinear ICA describes the case when the independent components are non-lineally combined to construct the observations. The nonlinear ICA is usually ill-posed, as there are several (potentially an infinite number of) solutions. The issue is that, for example, two independent components remain independent even after each one is individually nonlinearly transformed. Even worse, there are nonlinear mixtures of independent components that are themselves independent.

The solution in this case involves using a variant of the Self Organizing Map (SOM). The SOM should be constructed such that the data is uniformly distributed over the SOM grid. In this case, clearly the marginals (over the grid) are independent. For BSS problems, the SOM approach is not well suited, as it will find 'some' independent components and not 'the' independent components.

ICA for BSS purposes works only for post-nonlinear mixtures, i.e., when the observations are generated as nonlinear functions of linearly mixed components. The inverse nonlinearities are estimated with a neural network and then the independent components are estimated in a ML-approach.

VI. Discussion

A. ICA and sparse coding

Conducting ICA under the assumption that the sources are leptokurtic is equivalent to looking for sparse sources (assuming that the sources are mean free – that can always be achieved), hence the connection to sparse coding [3].

B. ICA and blind deconvolution

Blind deconvolution can be also expressed as an ICA problem. In this case, the original signal is related to the independent sources and the convolved signal to the observations. For the ICA model of blind deconvolution, we have as many independent sources as the length of the deconvolution filter, be this l. We express filtering (using circular convolution) as a matrix operation applied to overlapping chunks of length l of the original signal. To obtain the deconvolved signal we apply one unit of any sequential ICA algorithm. Due to our setup the deconvolved signal will exhibit a shift of between zero and l time instances with respect to the original.

C. ICA and classification

1) Feature extraction by ICA: ICA can be seen also as a feature-extraction transform. The question raised here it is when it is important to conduct the ICA and when it is enough to apply only a whitening transform. ICA can be seen as a rotation of whitened data. Therefore it makes sense to conduct ICA over whitening only if the additional rotation brings something in terms of classification accuracy.

If we assume we are interested only in feature extraction and not feature selection (i.e., we keep all components after the transformation), then ICA makes sense only if the subsequent classifier is not rotationally invariant. If we are interested in feature selection after ICA, then again, it may be that the ICA makes sense whatever the used classifier, as in this case we also have a projection. As described next, in some cases the procedure is equivalent to invariant feature extraction.

2) ICA for independent subspaces: ICA can be used to find also independent/invariant subspaces. To the limit each component can be regarded as an 1D independent subspace. The question is how to find ND independent subspaces and what is their meaning. It turns out that this can be done in an ML approach, assuming that the ND densities in the independent subspaces are rotational invariant, such that they depend on the sum of squared 'independent' components of the respective subspace (see [2] pp. 381).

3) Significance of various ICA methods: It has been shown empirically that when conducting feature extraction and feature selection by ICA, the results differ depending on the used method. This means that for example, ICA by ML finds other ICs then ICA by explicit rotation of whitened data. It can not be said that one method is consistently better over the other, just that they seem to return different ICs. I believe the reason for this is in the different approximations each method uses. Another reason may be the inherent limitations of the respective objective functions, as for example it is not guaranteed that a density is fully described by moments up to the fourth order.

D. ICA for rotational symmetric distributions

In this case, the joint distribution of the observations is rotationally symmetric (but not Gaussian) such that any further rotation leads to the same projections (as in the Gaussian case).

The solution in this case is to apply a nonlinear radial transform over the radial marginal densities of the rotational symmetric distributions (obtained after whitening the observations) such as to transform them to Gauss. Then they would be Gaussian and decorrelated and thus independent. Clearly this will not recover 'the' independent components but 'some' independent components (see [4]).

VII. Summary, Conclusions and Outlook

In the signal processing practice we are often confronted with the task of representating the data such that what it is important becomes easier to discover. What is important in the data depends on the problem to be solved and on the model assumptions we make. Here we made the assumption that the data is available as a combination of main information carriers defined as signal components that can not be represented as combination of other signal components being thus independent. To clarify the intuition behind this independent component analysis, it was introduced here as solution to a practical problem: a blind source separation type of problem. The importance of ICA is underlined also by the high number of scientific contributions on the topic. These introduced many ICA methods, however these methods can be classified in few general approaches. Here we have discussed two of them: by the first approach the ICs are sought after with the help of the standard stochastic definition of independence and by the second, they are sought after such that their densities are maximally different from the Gauss density. We have showed how methods of the same approach group together and how the two approaches are linked to each other. Another such main approach is the one based on nonnegative matrix factorization, which was just mentioned here.

Each of these main approaches makes certain assumptions on the independent components. These approaches are needed to compensate our lack of precise knowledge on the independent components and usually translate to some nonlinearities that appear in the solutions. For example, methods from the approach that uses the standard definition of independence make assumptions on the way the densities of the components can be represented parametrically. Methods that use the difference to the Gauss distribution hide these assumptions in the measure of nongaussianity. For example, the kurtosis is a good measure of non-Gaussianity for symmetric mono-modal densities and using the negentropy under this assumption on the densities of the ICs leads to using a certain type of nonlinearity when estimating it.

Alternatively, the ICA methods can be divided into block (also batch) methods that find all components at once and sequential (also unit) methods that find the components one after the other. Which one should be used depends on the targeted application. Further research is conducted into faster and more robust methods, which work very well even when we renounce our assumptions.

References

[1] A. Cichocki and S. Amari. *Adaptive blind signal and image processing*. John Wiley & Sons, 2002.

[2] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley and Sons, 2001.

[3] K. Labusch, E. Barth, and T. Martinetz. Sparse coding neural gas: learning of overcomplete data representations. *Neurocomputing*, 72(7-9):1547–1555, 2009.

[4] S. Lyu and E. P. Simoncelli. Nonlinear extraction of independent components of natural images using radial gaussianization. *Neur. Comput.*, 21:1485–1519, June 2009.

[5] R. Mazur and A. Mertins. Solving the permutation problem in convolutive blind source separation. In *Proceedings of the Independent Component Analysis and Signal Separation (ICASS)*, volume 4666 of *LNCS*, pages 512–519. Springer, 2007.

[6] L. Tong, R.-W. Liu, V.C. Soon, and Y.-F. Huang. Indeterminacy and identifiability of blind identification. *IEEE Transactions on Circuits and Systems*, 38:499–509, 1991.